Fifth IFSA World Congress (1993), 1125-1128

The Learning of the Neural Network Using Hadamard Transform

Hiromu KATAYAMA Shinichi TSURUTA Tomohiro NAKAO Hisamochi HARADA and Ryosuke KONISHI

Department of Electrical Engineering and Electronics
Faculty of Engineering, Tottori University, Koyama, Tottori 680, Japan

ABSTRACT

We propose the new method about the neural-based pattern recognition by using Hadamard transform for the improvement of learning speed, stability and flexibility of network. We can obtain the spatial feature of pattern by Hadamard transformed pattern.

We carried out an experiment to estimate the effect of Hadamard transform. We tried the learning of numeric patterns, and tried the pattern recognition with noisy pattern.

As a result, the learning times of the network for the 'Hadamard' case is smaller than that of usual case. And the recognition rate of the network for the 'Hadamard' case is higher than that of usual case, too.

NOTE: neural network, Hadamard transform

1 INTRODUCTION

Pattern recognition on image processing has been applied to ZIP code reader, parts inspection, and various field.

Generally, neural network has been used to pattern recognition, because it is the artificial model based on human brain with parallel computing and self-learning.

However, if we apply neural network to pattern recognition, it is important to set suitable learning parameters and learning patterns at the learning of network. Because those patterns and parameters have an influence on learning speed and stability of learning.

We propose the new method about learning of neural network by using Hadamard transform for the improvement of speed and stability of learning. Patterns are transformed into characteristic patterns by multiplication of Hadamard matrix.

In this paper, we carried out an experiment to estimate the effect of Hadamard transform at the learning of some numeric character patterns, compared with the usual learning.

And we tried the pattern recognition with noisy pattern to check the flexibility of the adjusted neural network.

2 THEORY

2.1 Hadamard Transform

The Hadamard matrix is a symmetric square matrix which elements are plus 1 and minus 1. Its rows (and columns) are mutually orthogonal. The simplest Hadamard matrix is that of order 2 as following.

$$H_2 = \begin{bmatrix} + & + \\ + & - \end{bmatrix} \tag{1}$$

If we want the Hadamard matrix of order $N = 2^n$, where n is an integer, we can obtain it by applying the Kronecker product recursively

$$H_n = \underbrace{H_2 \otimes H_2 \otimes \cdots \otimes H_2}_{n/2 \text{ times}} = \begin{bmatrix} H_{n/2} & H_{n/2} \\ H_{n/2} & -H_{n/2} \end{bmatrix} \tag{2}$$

,where the operation $\otimes$ is a Kronecker product.

By using this operation, we obtained the Hadamard matrix of order 8 and 16 for the experiment.

The Hadamard matrix can be used to obtain the Hadamard transform[1]. It is one of the orthogonal transform like Fourier transform, and successfully used in image processing and spectroscopy.

There are some transformation types on Hadamard transform. We show two types of them in eq.(3)(4)

$$Y^F_{m\times n} = H_m X_{m\times n} \tag{3}$$

$$Y^R_{m\times n} = X_{m\times n} H_n \tag{4}$$

,where $X_{m\times n}$ is the original matrix composed of $m \times n$ elements, $Y^F_{m\times n}$ and $Y^R_{m\times n}$ are the transformed matrix of $X_{m\times n}$ with Hadamard matrix H_m and H_n respectively.

2.2 Neural Network

Neural network is constructed with some elements called 'neuron' as shown Fig.1. Each neuron elements performs the sum of n inputs x_i ($i = 1, \cdots, n$) weighted by weights w_i and passes the results through a non-linear function f which also gives the threshold h. The output y is expressed as following.

$$y = f\left(\sum_{i=1}^{n} w_i x_i - h\right) \quad (5)$$

In combination with some neuron elements and finding the suitable weights of each neuron by learning, we can obtain the network adaptable the desired outputs for each input patterns.

Generally, we use the Back Propagation methods at the learning process on the layered neural network, which minimize the total error of each output by dividing it among weights and thresholds of each neuron in front.

At the method, we use the learning parameter α as following

$$w^{(t+1)} = w^{(t)} + \alpha \frac{\Delta E^{(t)}}{\Delta w^{(t)}} \quad (6)$$

,where $w^{(t+1)}$ and $w^{(t)}$ are the new value of weight obtained by the learning and old value of weight before, and $E^{(t)}$ is the total error of each output at the learning.

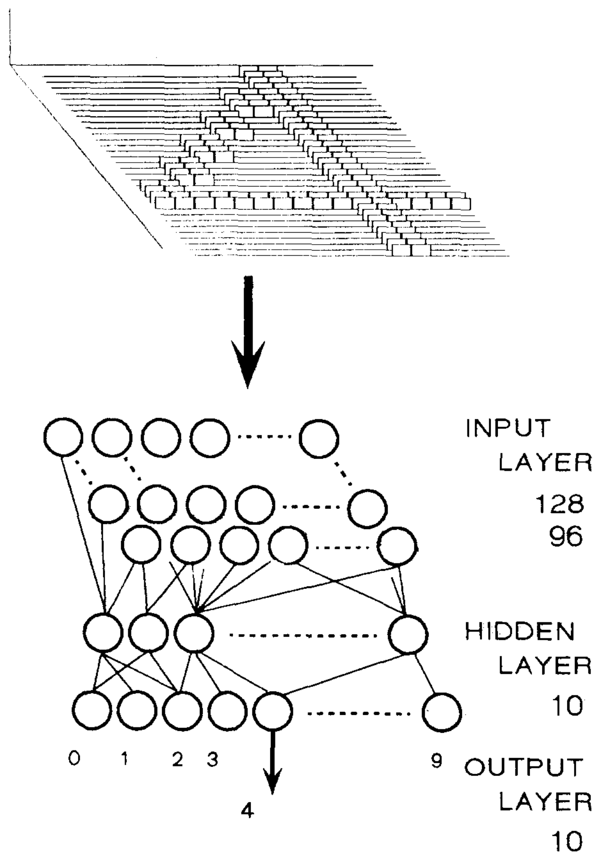

Fig.1 The three-layered neural network

3 RESULTS AND DISCUSSIONS

3.1 Learning of the Neural Network

We used the three-layered neural network as shown in Fig.1, which has 128 or 96 neurons for the input layer, 10 neurons for the hidden layer, and 10 neurons for the output layer. The sigmoid function is used as a non-linear function of each neuron as following.

$$f(x) = \frac{1}{1 + \exp(-x)} \quad (7)$$

Input patterns are the numeric characters from 0 to 9 as shown in Fig.2. Each patterns has 128 (16×8) regions. This is one of the reason that we chose the number of neurons for the input and output layer at the network.

We used the two cases of patterns as following.

Case-(A)
: Pattern (black region = 1, white region = 0) is simply given to the input layer. We call this case 'binary' case.

Case-(B)
: Two types of Hadamard transform (Eq.(3)(4)) is done at each pattern. 4 lines at the upper parts of matrix Y^F(32 elements) and 4 columns at the left parts of matrix Y^R(64 elements) are picked up as the transformed pattern, and are given to the input layer with division by 16. We call this case 'Hadamard' case.

If the input pattern is '4' for example, the data of the 'binary' is shown in Fig.3(a), and the data of 'Hadamard' case is shown in Fig.3(b).

For the 'Hadamard' case, transformed pattern Y^F obtained by Eq.(3) shows the concentration to the upper parts of matrix, and Y^R obtained by Eq.(4) shows the concentration to the left parts of matrix. So we decided to use the 'concentrated' parts by the Hadamard transform, and that is the another reason that we chose the number of neurons for the input layer.

We tried the learning of those patterns by changing the learning rate of weight α at the range of 0.05 to 0.90, and the learning rate of threshold β is fixed at 0.10.

Fig.4 shows the relation between learning rate of weight *alpha* and learning times. The horizontal axis is the value of the learning rate of weight *alpha* and the vertical axis is the learning times.

At $\alpha \geq 0.20$, the effect of Hadamard transform was indicated by the reduction of learning times.

At the 'binary' case, the learning with each value of α almost could not finish at 500 times, and varied more widely than that of 'Hadamard' case.

In contract, the learning times with each value of α almost finished at 200 times or below at the 'Hadamard' case.

At the great value of α, the change of each weight of neuron gets greater, and it depends on the input to each weight. So the learning with the 'binary' case, which has only two value of input 0 or 1, has a risk of divergence of network. However, the learning with the 'Hadamard' case has an ability of convergence of network, for it has some small value of input which are moderately distributed by the Hadamard transform.

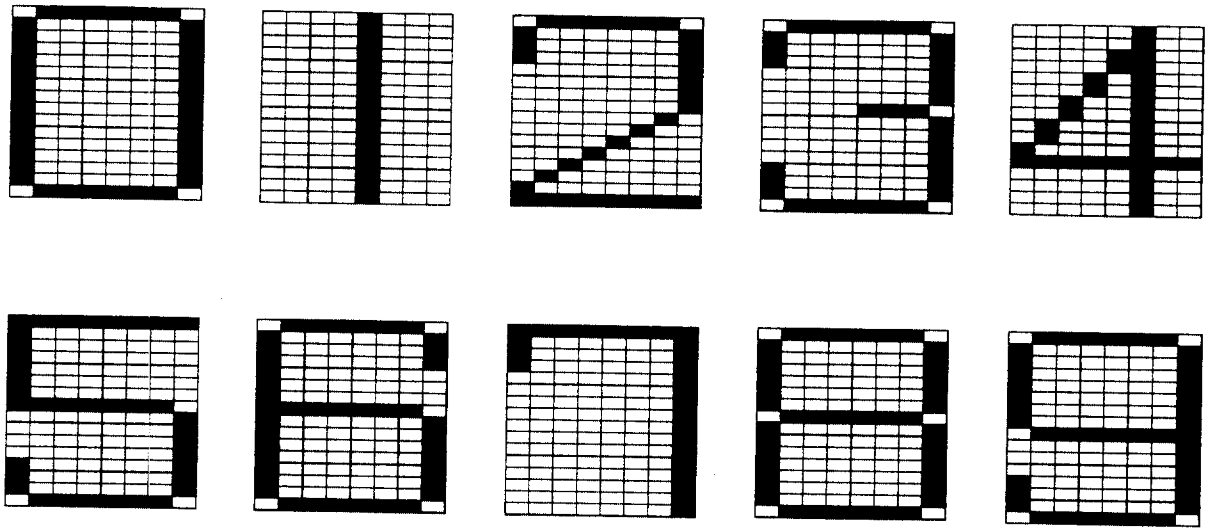

Fig.2 10 kinds of numeric patterns used for the learning and pattern recognition

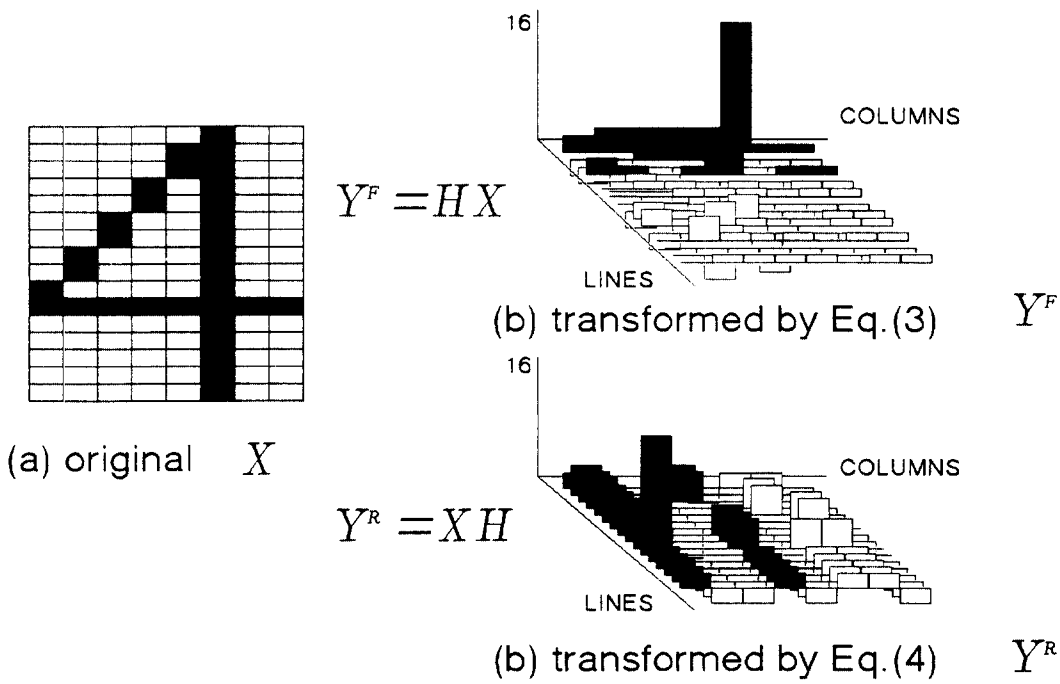

Fig.3 The two cases of pattern '4' given to the input layer

(a) for the 'binary' case

(b) for The 'Hadamard' case

In addition, this may be one of the reason to indicate the reduction of learning times for 'Hadamard' case that the number of input neuron is smaller than that for the 'binary' case.

3.2 Pattern Recognition with Noisy Patterns

We tried the pattern recognition with noisy pattern to check the flexibility of adjusted neural network.

We used the network which learning finished at 151 times for the 'binary' case, and we used the network which learning finished at 78 times for the 'Hadamard' case.

We put the black region(=1) or white region(=0) as a noise on each numeric patterns shown in Fig.2 at random. These noisy patterns are given to the input layer for the 'binary' case. And these patterns are transformed with Hadamard matrix and given to the input layer for the 'Hadamard' case.

Fig.5 shows the relation between the number of noise on the pattern and recognition rate. We tried 100 times recognition at each noise and pattern.

At each number of noise, the recognition rate of the network for the 'Hadamard' case is higher than that of network for the 'binary' case.

Fig.6 shows the recognition rate of each pattern. For the 'binary' case, the average of these recognition rate is 82 %. And for the 'Hadamard' case, the average of them is 86 %.

This is the reason to improve the flexibility of the network against noise for 'Hadamard' case that Hadamard transform extract the spatial feature of each pattern and reduce the influence of noise at pattern recognition.

4 SUMMARY

We carried out the experiment on the learning process and pattern recognition with noisy pattern. Through the experiment, we obtained some results as following.

a) At $\alpha \geq 0.20$, we could obtain the convergence of network at 200 times with the 'Hadamard' case, though we could not obtain it with the 'binary' case.

b) At many value of α, the reduction of the learning times was indicated with the 'Hadamard' case compared with the 'binary' case.

c) The distribution of the learning times with the 'Hadamard' case was smaller than that of the 'binary' case.

d) The recognition rate of the network for the 'Hadamard' case is higher than that of the network for the 'binary' case.

From these results, the method of the learning by using Hadamard transform is useful at the speed and stability of learning with many value of learning rate α.

And Hadamard transform is useful for pattern recognition at feature extraction and noise reduction.

REFERENCES

[1] W.K.Pratt, J.Kane and H.C.Andrews, "Hadamard Transform Image Coding", Proceedings of IEEE,Vol.57,No.1,pp58-68(1969).

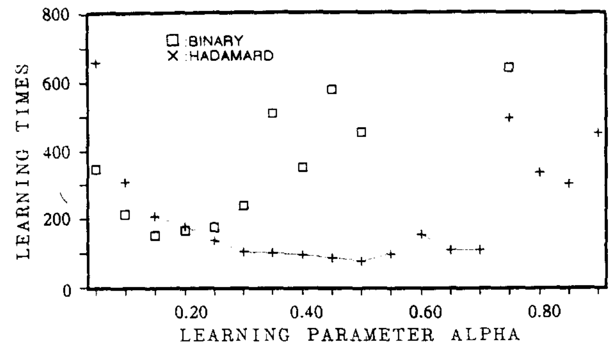

Fig.4 Relation between parameter α and learning times

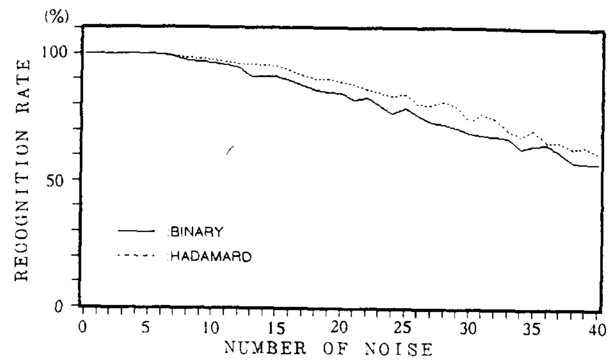

Fig.5 Relation between number of noise and recognition rate

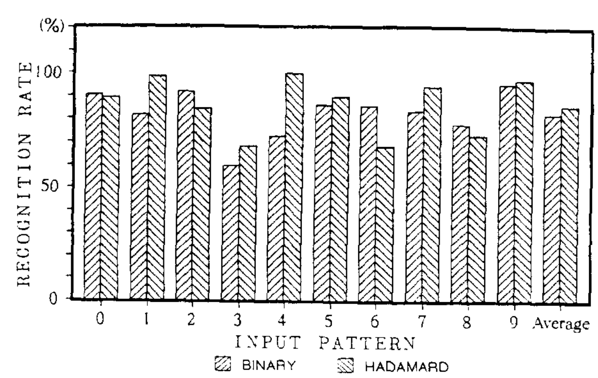

Fig.6 Recognition rate of each pattern