用 CPLD 建構灰楷漢明神經網路

The Implementation of Gray Level Hamming Neural Network Using Complex Programmable Logic Devices

林煒鋒　　　　汪啓茂
Wei-Feng Lin　　　　Chii-Maw Uang
Department of Electronic Engineering, Chinese Air Force Academy,
P.O. Box 90277-4, Kangshan, Kaohsiung County, Taiwan, R.O.C.

摘要

本文提出使用 CPLD 製作灰階漢明類神經網路之方法。此灰階漢明類神經網路之原型晶片包含 16 x16 個神經元，並可依需要而予以擴充。學習法則及分類函數以微碼寫入於唯讀記憶中。此晶片之學習及記憶能力可達 256 個灰階影像。

關鍵字:灰階漢明類神經網路, 自聯想, 異質聯想。

Abstract

In this paper, we will propose an easy way to implement the gray level Hamming neural network by using the Complex Programmable Logic Devices (CPLD). The prototype gray level Hamming neural net chip is composed of 16X16 (256 neurons) , and can be expandable. The learning rules and classifying functions are programmed in a micro ROM.

Keywords: Gray level Hamming net, Neural network, Complex Program-mable Logic Devices (CPLD).

1. Introduction

A Hamming net neural network is essentially a bipolar network[1]. We have extended the bipolar architecture to the gray level architecture[2,3], and proved to be a good image classifier[4-6]. However, to implement a gray level Hamming net by hardware is rather difficult[7,8]. In this paper, we will discuss the way of implementing the gray level Hamming neural network by using the Complex Programmable Logic Devices (CPLD)[9,10]. Although, the method by using gate array technology is not a suitable way to fabricate a neural chip, it is still a simple and easy way for testing the real hardware functions. Therefore, at this stage, a better solution for us to choose to verify the gray level Hamming network functions is the technology of using CPLD.

Based on the modified learning rule discussed in [3,5], we will propose a digital circuit that includes the input interface, the learning circuits, the classifier circuits and also the output interface to classify image or associate memorized patterns. The input and output interfaces should meet the real world requirements, so, the circuit must provide the function to avoid the problem of shifted gray level. The basic learning circuits is composed of 16X16 (256 neurons) , and can be expandable. The learning rules are programmed in a micro ROM. The classifier is actually a Max net circuit, which is also implemented by micro code. This neural chip can learn or memorize up to 256 images (patterns). A mapping circuit is added at the output end for having the ability of either autoassociation or heteroassociation. From research point of view, the work can be seen as a prestudy of the implementation of neural chip by using VLSI technology.

2. The Learning Rules

Suppose that the system is a G gray levels Hamming net with N neurons, which stores M exemplars. The *i*-th neuron, x_i^j, is associated with the *j*th stored pattern, X^j, and it is with R-gray level, $(0 \le R \le G-1)$. The interconnection weight, w_{ij}, in the gray level Hamming layer can be defined as[3]:

$$w_{ij} = x_i^j, \qquad (0 \le i \le N-1,\ 0 \le j \le M-1) \qquad (1)$$

where N is the number of input neurons and M is the number of stored patterns. x_i^j is the *i*-th element of the *j*-th exemplar and w_{ij} is the link weight from input node i to the Hamming layer output (Max net input) node j.

The weight matrix is the square of the number of neurons. In our case, N is 256, then the size of the weight matrix is 64k. Too many weight links is the most difficult problem of implementing neural chip by electronic circuit. We have proposed a simple way[5] to halve the size of the input neurons. Assume that A is a set of the input pixels of the original gray level image, then the neighbor pixels in A are partitioned into two sets, named A_o and A_e, which are the sets of the odd and even pixels of the input pixels, respectively, i.e.,

$$A^k = \{x_i^k | 0 \le i \le N-1,\ 0 \le k \le M-1\} \qquad (2)$$

$$A_o^k = \{x_j^k \big| j = 2i+1,\ 0 \le i \le \frac{N}{2}\} \qquad (3)$$

$$A_e^k = \{x_j^k \big| j = 2i,\ 0 \le i \le \frac{N}{2}\} \qquad (4)$$

$$A^k = A_o^k \,\mathrm{Y}\, A_e^k \qquad (5)$$

Before the image is sent to the network for training or classifying, we do a preprocessing, that is to calculate the gray level difference between the elements in A_o^k and A_e^k, i.e.,

$$B^k = \{b_j^k \big| b_j^k = abs(a_{oj}^k - a_{ej}^k),\ 0 \le j \le \frac{N}{2}\} \quad (6)$$

where abs( ) is an absolute function of B^k, and $a_{oj}^k \in A_o^k, a_{ej}^k \in A_e^k$, and B^k is now become the training pattern which will be used to replace the original pattern A^k. We see that the pixel number of B^k becomes half of A^k. The network size (the size of weight matrix) and cost are also reducedto a quarter. While the performance remains the same. Due to a subtraction operation in Eq. (6), the shifted gray level (DC bias) is automatically removed.

For easy implement by digital hardware, the interconnection weights of the second layer (Max net) can be obtained by calculating the matching score. The matching score at the *j*-th node can be calculated as

$$y_j = \sum_{i=0}^{N-1} [\mathrm{G} - \mathrm{abs}(w_{ij} - x_i^j)],\ (0 \le j \le M-1) \quad (7)$$

where abs( ) is an absolute function, and G is a constant. We can further define that G can be equal to the value of the gray level.

These matching scores are then fed to the Maxnet to select a node which has the maximum value. We will show a simple circuit in section 3, which interatively performs the winner-take-all function. Thus, in our case, the maximum node is found by iterative comparing each node in the Max net. Each iterative screens one node, so, the time complexity to finish winner-take-all is O(n), which is a linear relation with the stored patterns.

3. Hardware Implementation

The hardware implementation of gray level Hamming net using CPLD[9,10] can be shown in Fig. 1, in which the solid line denotes the data flow, and the dot line represents the control signal. The I/O block is used to transfer either data or command between host and this chip. The data are then sent to input buffer, which is used to store the training set or the test data temporarily. The training set data or test data can be captured from the CCD or inputted from host database. We note that for the same image, the data captured from the CCD will be varied case by case which is caused due to the DC bias or shifted gray level caused by the unnormal distribution of the light. Thus, in order to have a better image quality, we add a gray level normalization block right after the input buffer. As we mentioned in section 2, Eq(2)-(6). We add a pixel half logic to quarter the number of the weight links. The link decision block according with the learning block are used to construct the weight matrix of the Hamming layer (W_1) and output mapping layer (W_2) which is used for association. Eq.(1) is the basic operation rule used by the link decision logic to generate W_1 and W_2.

The command buffer is used to store the command conducted by the host computer. The commands include i/o operation; such as transfer input data and output classified results, and neural function; such as learning, classifying. The command is then sent to the control unit to generate suitable control signals for all blocks.

This chip is designed including 16x16=256 neurons. If more neurons are needed, then we can send a command to conduct expand logic to do chip expansion. The expansion units are 2, 4, 8,16,... and son on.

The classifying circuit composed of two parts; calculate matching score and Max net. Fig. 2 depicts the hardware logic to perform Eq.(7). The summation in Eq.(7) is performed under the control of micro codes, which are already bounded in the classifying circuit. There are three adders; the upper part is two adders with n-bit wide which are used to find the Hamming distance between node j and the input test pattern. The lower part adder is with 2n-bit wide in which a delay logic is introduced to calculate the summation shown in Eq(7). In our case, we have gray level normalization logic, which sets the gray level to 256, i.e., G=256, so, n is equal to 8 bits, and 2n is equal to 16 bits .

The Max net is depicted in Fig. 3, in which the maximum node is found under the control of micro codes, which are already put in the classifying block. The matching score of the node j is temporary stored in a register file with 256 registers each with 2n-bits wide. In our case 2n is equal to 16. The sequence generator is a 8-bit up counter. The micro codes drive control unit to generate control signal to the sequence generator to count from 0 to 255 (j=0 to 255). For each count, one matching score of node j is loaded to register A. Register B is started from a clear signal, then after each count, B is loaded with the larger one between A and B from the result of comparator. If A>B, then the index is kept in the index register, which is the finally classified output.

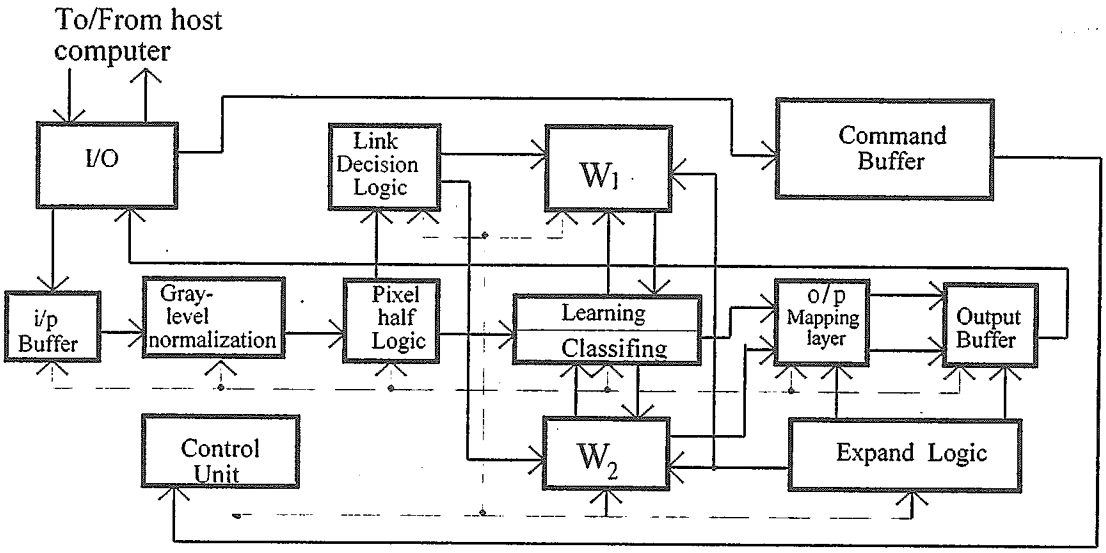

Fig. 1. The logic block diagram of the gray level Hamming net.

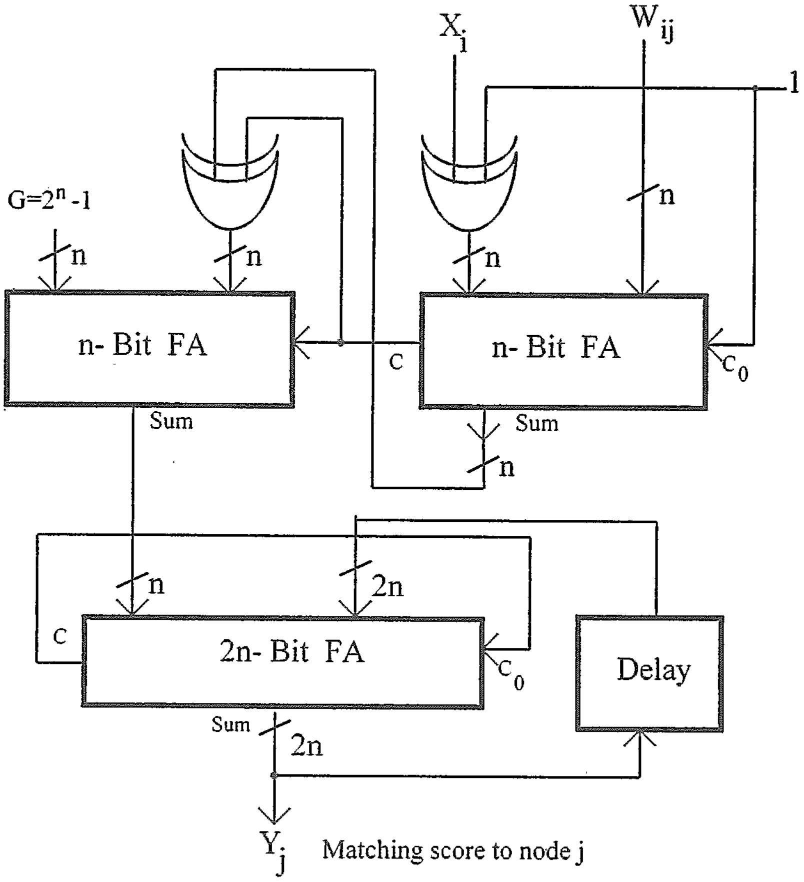

Fig. 2. The hardware configuration to calculate matching score.

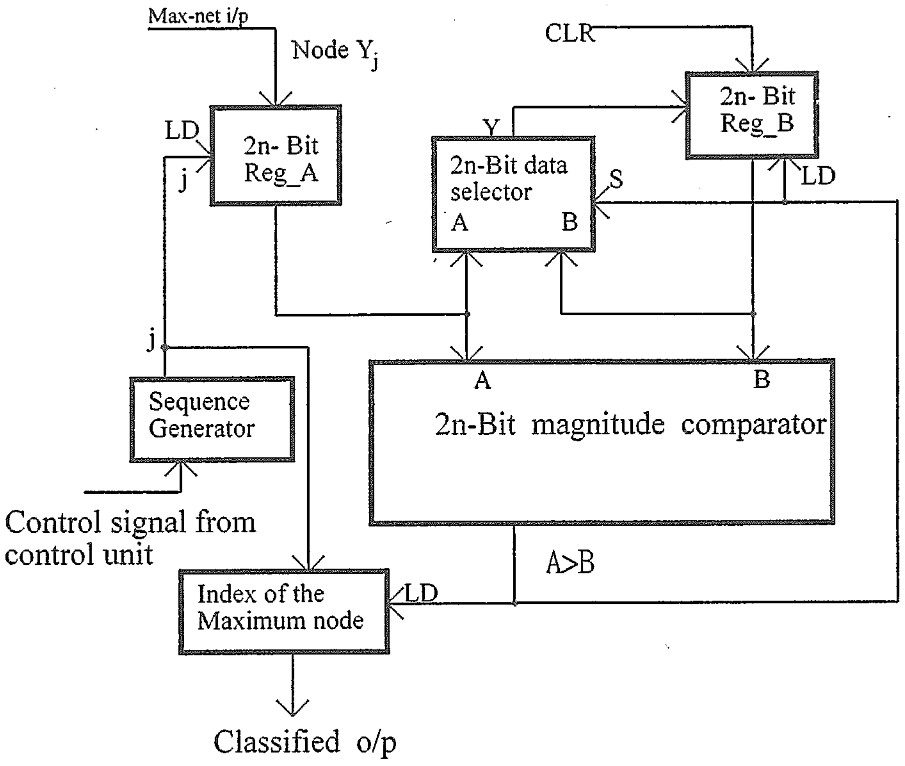

Fig. 3. The Max net.

A mapping layer is put right after the classified output. If the command from host is only to make classification, then the output is the index from the maximum index register, the classifying task is ended. If either autoassociation or heteroassoiciation function is required, then the output mapping logic will perform this task.

4. Summary

We have presented the the hardware implementation of the gray level Hamming net neural network by using CPLD. A digital circuit which includes the input interface, the learning circuit, the classifier circuit and also the output interface to report the classified result or associate the memorized patterns. Even through the prototype of this CPLD chips is only composed of 16x16 (256 neurons), it can be expanded to meet the real world application requirements.

Acknowledgment

This research was supported by the National Science Council of the Republic of China under contract NSC-86-2213-E-013-001.

References

[1] R. P. Lipmann, "An introduction to computing with neural nets," IEEE Trans. Acoust speech signal processing, ASSP-45, pp.4-12, Apr. 1987.

[2] F. T. S. Yu, Chii-Maw Uang, and S. Yin, "Gray-level discrete associative memory," Appl. Opt. 32, 1322-1329, 1993.

[3] Chii Maw Uang, Wei-Feng Lin, and Ming-huei Lu, "A gray level Hamming net for gray level image classification," ISANN'95, E3-13-19, 1995.

[4] Chii Maw Uang, Wei-Feng Lin, and Guowen Lu, "A unipolar Hamming net neural network," 1994 Intern. Comp. Conf. Symp, ICS'94, 1251-1255, 1994.

[5] Wei-Feng Lin, and Chii-Maw Uang, "The Performance Analysis of a Gray Level Hamming

net," The third Military Academy Symposium on Fundamental Science, 409-412. 1996.

[6] F. T. S. Yu, Guowen Lu, and Chii Maw Uang, "On the analysis of a multilayer Hopfield-type associative memory," 1994 Intl. Symp. on Artificial Neural Network, ISANN'94, 314-320, 1994.

[7] Chii-Maw Uang and F. T. S. Yu, "Implementation of polychromatic Hamming Net for color image classification," Appl. Opt. 33, 5443-5447, 1994.

[8] F. T. S. Yu, S. Yin, and Chii-Maw Uang, "A content-addressable polychromatic neural net using a (Ce:Fe)-dopped $LiNbO_3$ photorefractive crystal," Optics Communications **107**, pp.300, Apr. 1993.

[9] Pak K. Chan, S. Mourad, *Digital Design Using Field Programmable Gate Array*, Prentice-Hall, 1994.

[10] Xilinx Inc., *The ISP Application Guide and CPLD Data Book*, Xinlinx Inc., 1996.