Seminar Nasional Aplikasi Teknologi Informasi 2007 (SNATI 2007) *ISSN: 1907-5022*
Yogyakarta, 16 Juni 2007

HANDWRITTEN ALPHABET RECOGNITION USING *HAMMING NETWORK*

Arnold Aribowo, Samuel Lukas, Handy
Universitas Pelita Harapan, Indonesia
UPH Tower, Lippo Karawaci, Tangerang 15811, Indonesia
e-mail: {arnold, slukas}@uph.edu

ABSTRACT

Handwritten Alphabet Recognition is one of the artificial intelligence applications which provides an important fundamental for various advanced applications, including information retrieval and human-computer interaction applications. This paper elaborates a research which is performed to build a system which is able to recognize handwritten latin alphabets in the form of images. The system is developed using the Hamming Network method.

From the experiments of 100 prototypes of data, the system is able to identify handwritten latin alphabets with 76.97% average accuracy. Mistaken recognitions are mainly caused by the similarity of the alphabet patterns, such as the pattern of I, T and Z, or the pattern of C and G. The value of ε and the matrix b which are given in the system may affect the recognition results.

Keywords: *Handwritten Alphabet Recognition, Hamming Network*

1. INTRODUCTION

Artificial intelligence can be defined as the art of creating machines that perform functions that require intelligence when performed by people [2]. There are six disciplines which people focuses on to enable computer has intelligence [6]. They are natural language processing, knowledge representation, automated reasoning, machine learning, computer vision, and robotics. One of the important primary processes which is discussed to enable a machine works in a way which require intelligence when performed by people is the process of recognizing objects. This process is related with machine learning and computer vision discipline. This process provides a fundamental for many advance applications, such as information retrieval and human-computer interaction applications.

Handwritten alphabet recognition is the system which is able to recognize handwriting characters that are given (one by one). Initially, all the characters will be learned by the system. Through the learning processes which are previously performed, when an input character is given, it is identified by the system.

This paper elaborates the development of a handwritten alphabet recognition system using the Hamming Network method. In section 2, related works are provided. Section 3 would summarize theories on hamming network. Section 4 elaborates design and implementation of the system. Section 5 evaluates performance of the system. Section 6 concludes the paper by mentioning several important points, and gives ideas for future works.

2. RELATED WORK

There are several related researches which was performed to recognize objects. The hamming network was used to identify characters in the form of images, however the character identified was not in the handwritten form [8]. There are three kinds character, which are identified. They are Times New Roman, Ariel Rounded Bold and Courier New.

Research on handwritten recognition was performed by applying fuzzy logic and backpropagation neural network [3]. The system which is built consists of preprocessing subsystem, feature extraction subsystem and classification subsystem.

The similar researches on handwriting recognition were done by applying neocognitron method [4] and another research which implements backpropagation neural network to recognize numerical arabic characters [7].

3. THEORETICAL BACKGROUND

The Hamming network method was developed by a mathematician, Richard W. Hamming. He has many contributions not only in the mathematical field, but also for computer science and telecommunication [5]. He was also the founder and has been the president of Association for Computing Machinery.

Hamming network method is developed to solve pattern recognition problems which use binary format, such as a matrix with only two possible values, 0 and 1. In the Hamming network, there is a matrix which stores the patterns of all objects, called the prototype data matrix. The patterns will not be learned by the system, but rather to be stored as a matrix data. The matrix will be used to define the output of the network. The objective of the Hamming network is to decide which prototype matrix is closest to the input matrix. It calculates the similarities between the prototype matrix of all objects and the input.

The Hamming network method is divided into two parts, the feedforward layer and the recurrent layer.

3.1 Feedforward Layer

Feedforward layer is a layer which calculates the correlation between each patterns of the prototype matrix and the input matrix (figure 1). The calculation results will be processed to generate the output neurons for this layer.

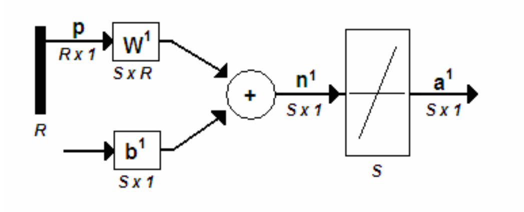

Figure 1. The Feedforward layer

As shown in the figure 1, the layer has the input matrix from p, which has the dimension as R x 1. This input matrix goes to the weight matrix (W^1) with the dimension of S x R. The net of this layer (n^1) will be the sum of the W^1p and the bias input b.

The weight matrix of W^1 will be the matrix of the prototype data which include the patterns of all objects. The element of the bias b will be given as the number of R. The transfer function which is used in this layer is the linear transfer function (purelin). This function will not change the value so the output of this feedforward layer (a^1) will be given as:

$$a^1 = \mathrm{purelin}(W^1p + b^1) \Leftrightarrow a^1 = W^1p + b^1 \quad (2.1)$$

The output neurons of this layer will be used as the initial input for the next layer, the recurrent layer.

3.2 Recurrent Layer

The recurrent layer is also known as a competitive layer. In this layer, there is a neuron for each prototype pattern. The neurons in this layer are initialized with the output neurons of the feedforward layer, which indicate the correlation between the prototype patterns and the input matrix. The neurons will compete each other to determine a winner. When the processes are finished, there will be only one neuron with nonzero output. This neuron indicates the prototype pattern that is closest to the input.

The processes in the recurrent layer will be divided into iterations. After one iteration is finished, a function will check whether there is only one nonzero output. If so, the process in this layer will be stopped, and the process will continue to generate the output.

Figure 2 below shows the structure of the recurrent layer.

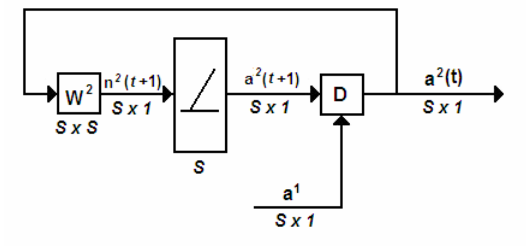

Figure 2. The Recurrent layer

As depicted in the figure 2, D is the function to check whether there is only one nonzero output. W^2 is the weight matrix for this layer with the dimension of S x S. The iteration number will be given as t, and it will be added by one until the iteration stopped. The activation function which is used is the positive linear transfer function (poslin). This function is linear for positive values and zero for negative values. The equation for the transfer function can be defined in the following way:

$$a = f(n) = \begin{cases} 0 & \text{jika } n < 0 \\ n & \text{jika } n \geq 0 \end{cases} \quad (2.2)$$

The weight matrix W^2 will be given as the equation below:

$$w_{ij}^2 = \begin{cases} 1 & \text{if } i = j \\ -\varepsilon & \text{if } i \neq j \end{cases} \quad (2.3)$$

In the above equation, ε is some number which is less than 1/(S-1). Therefore, the output neuron of this recurrent layer can be given as:

$$\mathbf{a}^2(t+1) = \mathbf{poslin}(\mathbf{W}^2\mathbf{a}^2(t)) \quad (2.4)$$

where $a^2(0) = a^1$.

4. DESIGN AND IMPLEMENTATION OF THE SYSTEM

The system which is developed is a system that gets an input of character, process it through the network, and generates the result. The alphabets which are used in the development are limited to handwritten capital Latin alphabet from A to Z. The system has some prototype data that consists of the pattern of all alphabets, from A to Z. This prototype data is used as the weight matrix for the process in the feedforward layer of the Hamming network. The system is built using the Microsoft Visual Basic 6.0 and the images is processed using the Microsoft Paint.

The alphabet data is obtained from 15 persons. Each person writes the 26 alphabets on a piece of paper and then it is scanned into a computer. Afterwards, every alphabet is saved as one image file. The type of the image file is bitmap (.bmp) and the size of the file is 120 x 100 pixels (height x width).

The prototype data is then created by taking 26 alphabets from 10 persons. The data is processed and then saved as a matrix. This matrix is saved as a text file (.txt). In the recognition process, user loads a prototype data to be used.

The system will have a function that simulates the Hamming network. The function will act as the network and process the input data to generate the output. The output neuron of the network indicates the result of the recognition process.

4.1 Image Processing

Every time an image of alphabet is used in the system, it will be initially processed, because the system can only recognize binary input. As mentioned before, the size of the image file is 120 x 100 pixels. This image will be converted into a matrix by reading every pixel of the image. A pixel which has black color will be represented by the number of 1 and a pixel which has white color will be represented by the number of 0.

After the image is converted into a matrix, it will be continued with the process of cropping and scaling. The cropping process will cut the matrix so that a matrix is focused on the object of the alphabet. This process will remove the unused part of the matrix. The example of the cropping process can be shown in the following table :

Table 1. Cropping process

Initial matrix	*After cropping process*
000000000000000	000010000
000000010000000	001111100
000001111100000	111111111
000111111111000	001111100
000001111100000	000010000
000000010000000	
000000000000000	

The above table shows how the cropping process of the matrix is performed. The matrix is now focused on the object that is represented by the number of 1.

After the cropping process, the matrix will be scaled through the scaling process. The purpose of this process is to get a smaller size of the matrix and to standardize the size of matrix. This standarization should be done because the cropping process will produce a different size of matrix for every alphabet. Size of the matrix after the scaling process will be the same for all alphabets, that is, 12 x 10.

The matrix that is used for the input or the formation of prototype data is the matrix that has been processed through the cropping and scaling processes.

4.2 Details in the Hamming Network

As mentioned before, the input image will be processed first, therefore the matrix will be 12 x 10. On the other hand, a prototype data is consist of the pattern of all 26 alphabets. Therefore, in the prototype data, there will be 26 matrices of 12 x 10 which are stored in a stack, from the matrix for the alphabet A to the matrix for the alphabet Z.

In the feedforward layer, $\mathbf{p}$ is the input matrix. It will be the matrix of the input image which size is 12 x 10. Therefore, the input $\mathbf{p}$ will be a matrix of 120 x 1. The $\mathbf{R}$ number is 120, which is the number of input neuron, while $\mathbf{S}$ is the number of the output neuron for this network, which is 26. The weight matrix $\mathbf{W}^1$ will be generated using the prototype data. It will take the prototype data matrix of the 26 alphabets, so the weight matrix will be a matrix of 26 x 120. The bias $\mathbf{b}$ will be a matrix of 26 x 1.

In the recurrent layer, the weight matrix $\mathbf{W}^2$ will be a matrix of 26 x 26. It is similar to an identity matrix. There are 26 output neurons which represent the number of alphabets. Therefore, the output of this layer will be a matrix of 26 x 1.

5. PERFORMANCE EVALUATION

Some experiments have been made to know the performance of the Hamming network method in developing a handwriting alphabet recognition system. The tests which is performed consist of :

a) Tests of 100 prototype data
b) Test with different ε values
c) Test with different bias b values

The result of the test of 100 prototype data can be shown in the table 2 and 3.

Table 2. Result of 100 prototype data test (1-50)

Prototype	Average Percentage	Prototype	Average Percentage
1	77.18	26	73.85
2	77.44	27	77.95
3	75.90	28	76.67
4	77.44	29	75.90
5	76.92	30	73.33
6	75.90	31	78.21
7	78.46	32	74.87
8	77.44	33	77.44
9	73.08	34	80.77
10	80.51	35	77.95
11	76.67	36	77.95
12	75.90	37	77.95
13	76.67	38	77.44
14	76.92	39	76.15
15	76.92	40	76.67
16	78.46	41	74.87
17	76.15	42	80.26
18	78.46	43	74.36
19	80.51	44	75.13
20	78.97	45	77.69
21	79.49	46	80.26
22	80.00	47	75.13
23	78.97	48	80.00
24	78.21	49	75.90
25	73.85	50	76.67

Table 3. Result of 100 prototype data test (51-100)

Prototype	Average Percentage	Prototype	Average Percentage
51	74.10	76	79.49
52	72.31	77	76.67
53	79.74	78	75.13
54	75.90	79	78.72
55	**71.28**	80	82.05
56	74.62	81	79.23
57	75.64	82	78.21
58	75.64	83	71.54
59	76.92	84	78.46
60	76.15	85	79.74
61	79.49	86	73.59
62	79.23	87	80.00
63	73.85	88	74.36
64	74.10	89	78.46
65	74.10	90	74.62
66	79.49	91	76.41
67	77.69	92	76.92
68	75.13	93	76.41
69	77.69	94	77.95
70	73.33	95	77.44
71	75.64	96	77.44
72	**82.56**	97	73.59
73	78.97	98	80.00
74	76.15	99	75.64
75	76.67	100	78.97

The percentage of the table above is the average number of the success recognition result for every alphabet. The inputs used are all the alphabet data from 15 persons that have been taken before. From the above table, the highest percentage is 82.56% and the lowest is 71.28%.

The result of the test of 100 prototype data can be given according to the alphabets as follows:

Table 4. Result according to the alphabets

Alphabet	Percentage	Alphabet	Percentage
A	97.80	N	92.20
B	98.27	O	67.73
C	49.87	P	87.67
D	87.80	Q	81.33
E	86.00	R	89.73
F	40.40	S	53.20
G	70.60	T	77.13
H	74.93	U	73.73
I	32.33	V	89.00
J	77.20	W	92.93
K	88.20	X	90.73
L	50.53	Y	70.93
M	83.93	Z	97.07

From the above table, it is shown that the highest percentage of the true recognition results is taken by the alphabet B, and the lowest percentage is taken by the alphabet I.

For the experiment using different values of ε, the input alphabets will be taken from one person, and the prototype used is prototype number 72. The result can be given as follows :

Table 5. Test of different ε values

Alphabet	Prototype 72 – Average Percentage of True Result		
	ε=0.03 & b=26	ε=0.02 & b=26	ε=0.01 & b=26
A	100	100	100
B	100	100	100
C	46.67	46.67	46.67
D	100	100	100
E	86.67	86.67	86.67
F	80	73.33	80
G	66.67	66.67	66.67
H	80	73.33	80
I	60	40	53.33
J	86.67	86.67	80
K	93.33	93.33	93.33
L	53.33	60	40
M	93.33	93.33	93.33
N	93.33	93.33	93.33
O	66.67	66.67	60
P	93.33	93.33	93.33
Q	93.33	93.33	93.33
R	100	93.33	100
S	53.33	53.33	53.33
T	66.67	66.67	66.67
U	80	80	80
V	93.33	93.33	93.33
W	93.33	93.33	93.33
X	93.33	93.33	93.33
Y	73.33	73.33	93.33
Z	100	100	100
Average	**82.56**	**81.28**	**82.05**

From the above table, the percentage may change if the ε value is different. However, the result is not definite. Sometimes bigger value of ε produce better percentage, but sometimes it is not.

For the test of different values of bias b, the result is shown on the following table:

Table 6. Test of different bias b values

Alphabet	Prototype 72 – Average Percentage of True Result		
	ε=0.03 & b=0	E=0.03 & b=26	E=0.03 & b=50
A	100	100	100
B	100	100	100
C	46.67	46.67	40
D	100	100	100
E	86.67	86.67	86.67
F	66.67	80	73.33
G	66.67	66.67	66.67
H	73.33	80	80
I	60	60	66.67
J	86.67	86.67	86.67
K	93.33	93.33	93.33
L	53.33	53.33	46.67
M	93.33	93.33	93.33
N	93.33	93.33	93.33
O	60	66.67	60
P	100	93.33	93.33
Q	100	93.33	93.33
R	100	100	93.33
S	53.33	53.33	53.33
T	66.67	66.67	66.67
U	80	80	80
V	93.33	93.33	93.33
W	93.33	93.33	93.33
X	93.33	93.33	93.33
Y	73.33	73.33	80
Z	100	100	100
Average	**82.05**	**82.56**	**81.79**

The result of the above test shows that the percentage may also change if the b value is different. However, the change is also not definite because smaller value of bias b used does not always produce a better result.

6. CONCLUSION

This paper discuss about the development of a handwriting recognition system using Hamming network method. The processes of the Hamming network involve only matrix operations. From the experiment results, the system is good enough in recognizing the alphabets. However, there are some alphabets which have big percentages of failure recognition result. They are C, F, I, L, and S. From the analysis, those false recognition results are caused by the similarities of the pattern of some alphabets. For example, the alphabet I has a similar pattern with the alphabet T and Z. The experiments using different values of ε and b show that the recognition results may change depend on the values used.

REFERENCES

[1] Hagan, Martin T., *Neural Network Design*, Boston, USA: PWS Publishing Company, 1996.
[2] Kurzweil, *The Age of Intelligent Machines*, Cambridge, Massachusetts: MIT Press, 1990.
[3] Kusumoputro, B., E. Philipus, M. Rahmat Widyanto, Pengenalan Huruf Tulisan Tangan Menggunakan Logika Fuzzy dan Jaringan Syaraf Tiruan, Surabaya, Indonesia: *Proceeding, Seminar of Intelligent Technology and Its Applications (SITIA '2000)*, 2000.
[4] Patricia Marta, Hanapi Gunawan, Hendra Tjahyadi, Pengenalan Angka Tulisan Tangan dengan Jaringan Syaraf Tiruan Menggunakan Metoda Neocognitron, Surabaya, Indonesia: *Proceeding, Seminar of Intelligent Technology and Its Applications (SITIA '2000)*, 2000.
[5] Richard Hamming at: http://en.wikipedia.org/wiki/Richard_hamming. (accessed: 27 Juli 2006).
[6] Russell, Stuart J. and Peter Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*, New Jersey, USA: Pearson Education, Inc., 2003.
[7] Sepreni, *Pengenalan Tulisan Tangan pada Karakter Numerik Arab menggunakan Jaringan Syaraf Tiruan (Artificial Neural Network) Model Backpropagation*, Tangerang, Indonesia: Universitas Pelita Harapan, 2004.
[8] Thiang, Pengenalan Karakter dengan Menggunakan Hamming Network, Yogyakarta, Indonesia: *Proceedings Seminar Nasional Aplikasi Teknologi Informasi 2005*, 2005.