NEURO-CHAIN :A HANDWRITTEN CHARACTER RECOGNITION SYSTEM

M.Fatih Amasyalı, Nilgün Erdem, Hakan Haberdar, Filiz Koyuncu,
Yıldız Technical University, Department of Computer Engineering, Istanbul, Turkey,
[mfatih,nilgun,hakan,filiz] @ce.yildiz.edu.tr

Abstract

In previous studies, characters were given to character recognition systems in a pixel position dependent structure. In this study, direction information, that is obtained from pixel positions(chain code), were used instead of pixel position information. After obtaining chain code, there is two basic operation that are used to make data ready for system. One of that is straightening and the other one is summarizing of chain code. Summarized chain code is used in training of 2 different neural network structure. The performance of different neural network structures and template matching are compared.

Keywords

Chain Code, Artificial Neural Networks, Handwritten Character Recognition, Circular Average

1. INTRODUCTION

Neural network is one of the main systems that are used at several hand-written character recognition studies. In this study, direction information(known as chain code), that is obtained from pixel positions, were used. After obtaining chain code, there is two basic operations that are used to make data ready for system. One of that is straightening and the other one is summarizing of chain code. Summarized chain code is used in training of Back Propagation and Learning Vector Quantization Nets(Figure 1.1). In figure 1.2 shows how the performance of system is measured.

In the second section, it is explained why we used the chain code. In the following section, we summarize the algorithm of the chain code. In the fourth section, chain codes are prepared for classification procedures. Classification methods used in this study, are explained in the last section.

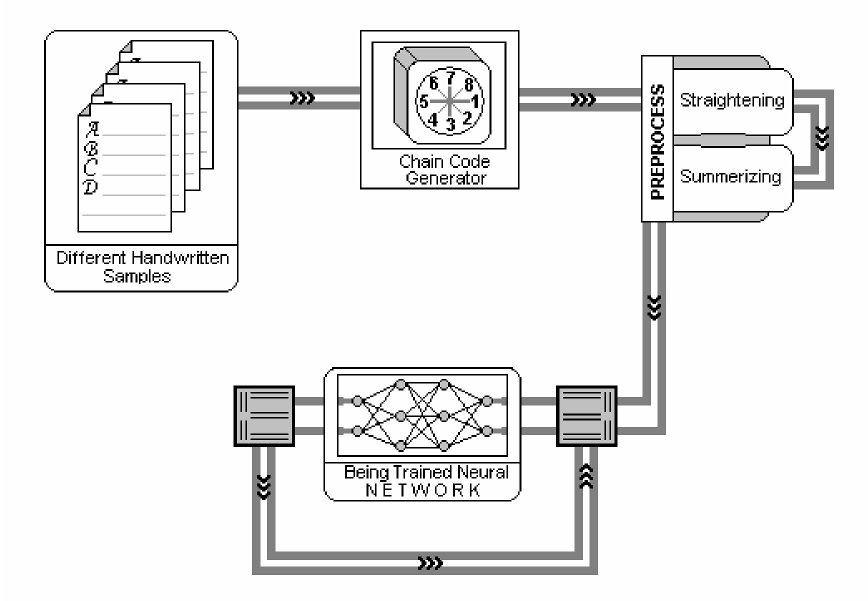


Figure 1.1 Training

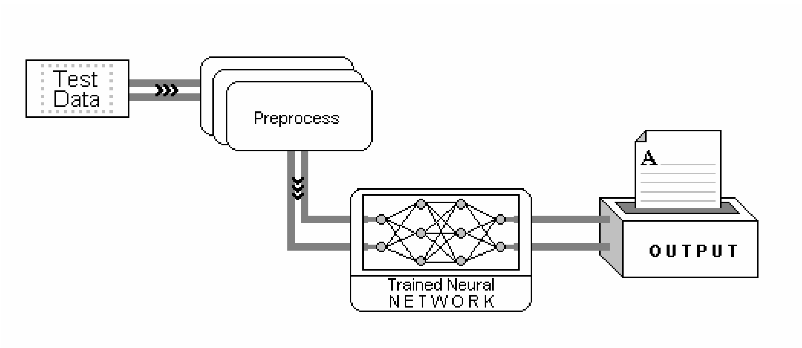


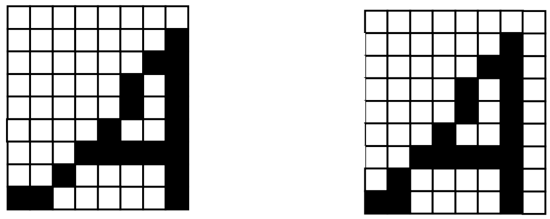
Figure 1.2 Test of Systems

2. WHY CHAIN CODE USED

According to our experimental studies, the direction information obtained from the pixel maps of some characters, show noticeable similarities. That's why we used chain code algorithm.

Example: In fig. 2.(a) and (b), the character A is shown. Although the both figures show the character A, the template matching of the pixel maps will not give the desired result. But the direction information is very similar to each other. If we use chain code algorithm on fig. 2.(a) and (b), we will obtain the following result:

Chain A1: 3 3 3 3 3 3 3 7 6 5 5 5 4 4 5 1 8 8 8 8 7 8 8
Chain A2: 3 3 3 3 3 3 3 7 6 5 5 5 4 3 5 1 8 8 8 8 7 8 8



(a) (b)

Figure 2: Pixel map of character A

3. CHAIN CODE ALGORITHM

The basis were introduced in 1961 by Freeman [1] who described method permitting the encoding of arbitrary geometric configurations. In this approach, an arbitrary curve is represented by a sequence of small vectors of unit length and a limited set of possible directions, thus termed the *unit--vector method*. On the digital (rectangular or six--connected) grid, encoding is based on the fact that successive contour points are adjacent to each other. Depending on the 8-connected grid is employed, the chain code is defined as the digits from 1 to 8, assigned to the 8 neighbouring grid points in a counter--clockwise sense. An illustration is given in Fig.3.1. A direct straight--line segment connecting two adjacent grid points is called a link, and a chain is defined as an ordered sequence of links with possible interspersed signal codes [2].

First, the origin point of the character should be located. As to the algorithm in this paper, the origin point is considered as the most left pixel of the uppermost. Starting from origin point, the position of the following pixel to the former one represents the direction. In figure 3.1, the directions and the numbers representing the directions are shown. In figure 3.2 shows the indice relationship between a pixel(i,j) and its 8-neighbours.

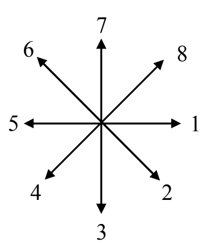


Figure 3.1: Definition of the chain code in the 8-connected grid.

i-1,j-1	i-1,j	i-1,j+1
i ,j-1	i , j	i,j+1
i+1,j-1	i+1,j	i+1,j+1

Figure 3.2: indice relationship between a pixel(i,j) and its 8-neighbours

4. MODIFIED CHAIN CODE

Two sequential procedures are used to prepare the chain codes for classification procedures.

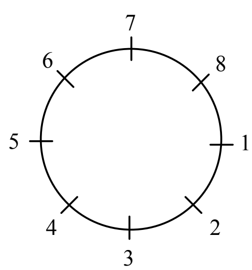
4.1 Straightening Methods

Two different methods are considered as straightening procedure and the effects of these two methods on the performance of the system are inquired.

4.1.1 Neighbourhood simulation: For every element of the chain code array; if the nearest neighbours values of the element of the array are equal to each other, this element's value is set to its neighbour's value.

before : 1 2 1 1 3 1 2 2 4 2 2 1 3 1 2 8 8 1 8
after: 1 1 1 1 1 1 2 2 2 2 2 1 1 1 2 8 8 8 8

4.1.2 Circular Average: In this method, circular average of every element and its closest neighbour elements is calculated. In figure 4.1.2 each number on the circle is greater than the number that is on its left side. For example ″One″ is greater than ″eight″. According to that the circular average is calculated.



before : 1 2 1 1 3 1 2 2 4 2 2 1 3 1 2 8 8 1 8
after: 1 1 1 1 1 1 2 3 3 2 2 2 2 2 1 8 8 8 8

Figure 4.1.2: number cycle

4.2 Summarization of chain codes

All data must be given in the same dimension to Neural Network. Because of the difference between chain code lengths of our patterns, all of them are reduced into a constant length.

5. CLASSIFYING TECHNICS

Three classification methods are used. The first one is Backpropagation, the second one is Learning Vector Quantization and the last one is Template Matching that consist of two approach (Maximum Correlation and Minimum Error).

5.1 Backpropagation algorithm

The backpropagation algorithm was derived from a Least Mean Squares Approach and a description of this may be found in Rumelhart and McClelland [3]. What follows is a description of how to implement the algorithm which is essentially derived from Lippmann [4]. If the bias terms are imagined as an input which is always set to 1 and fed through a weight, The output equation of basic element of the network is given in equation 5.1.1.

$$Output_j = S(\sum_{i=0}^{N} w_{ij} x_i) \quad (5.1.1)$$

where S is a sigmoid non-linearity activation function(Equation 5.1.2).

$$S(a) = \frac{1}{1 + Ae^{-a}} \quad (5.1.2)$$

where A determines the level of non-linearity.
where

- [x_i] is input i
- [w_{ij}] is the weight connecting input *i* to neuron *j*.

x_i is input *i* and w_{ij} Now for the algorithm[5].

First the error for the output layer nodes is computed by using equation 5.1.3.

$$E_j = (t_j - a_j) a_j (1 - a_j) \quad (5.1.3)$$

where

- [E_j]error for node j of the output layer
- [t_j] target activation for node j of the output layer
- [a_j]actual activation for node i of the output layer

Then, successively, the error values for all hidden layer nodes are computed by using equation 5.1.4.

$$E_i = a_j (1 - a_j) \sum_j E_j w_{ij} \quad (5.1.4)$$

where

- [E_i]error for node i in a hidden layer,
- [E_j]error for node *j* in the layer above,
- [w_{ij}]weight for the connection between node i in the hidden layer and node *j* in the layer above,
- [a_i]activation of node *i* in the hidden layer.

At the end of the error backward propagation phase, nodes of the network (except the input layer nodes) will have error values. The error value of a node is used to compute new weights for the connections which lead to the node. Very generally, the weight change is done by using equation 5.1.5.

$$w_{ij} = w_{ij} + \Delta w_{ij} \tag{5.1.5}$$

where

- [w_{ij}]weight of the connection between node *i* in the previous layer and node *j* in the output layer or in a hidden layer,
- [Δw_{ij}]weight change for the connection between node *i* in the previous layer and node *j* in the output layer or in a hidden layer.

5.2 Learning Vector Quantization Algorithm

Learning vector quantization is a supervised algorithm. It basis on competitive learning. The algorithm is described by the following code[6]:

```
Begin
  Initialize η, n, μ_1, ..., μ_c
  Repeat
        Accept a new pattern x
        Find nearest cluster:
        k ← arg min_j ||x − μ_j||
        If k is the correct label
        μ_k ← μ_k + η(x − μ_k)
        Else μ_k ← μ_k − η(x − μ_k)
  Until no significant change in μ in n epoch
End
```

where

- [η] is learning parameter,
- [n] is maximum epoch number,
- [c] the number of representative vectors
- [$\mu_1,...,\mu_c$] are representative vectors(centroids),
- [x] a data vector from training set.

5.3 Template Matching

Template matching is a natural approach to pattern classification. The training data can be used as templates. To classify one of the test example(T_i), simply compare it to the other templates(A). And it is classified as the class of nearest neighbour. Its is shown in Equation 5.3.1.

$$k = \arg\min_k \{ \mathrm{d}(\mathrm{T}, \mathrm{A}_\mathrm{t}) \;\; \mathrm{t} = 1,2,...294\} \tag{5.3.1a}$$

$$A_{k_c} \text{ is the class of } \mathrm{A_k}. \qquad T_c \leftarrow A_{k_c} \tag{5.3.1b}$$

We used two distance function. These functions are given in Equation 5.3.2 and 5.3.3.

$$d(T, A_k) = \sum_{i=1}^{30} f(T_i, A_{k\,i}) \tag{5.3.2}$$

$$f_1(x,y) = \begin{cases} 0 & x = y \\ 1 & else \end{cases} \tag{5.3.3a}$$

$$f_2(x,y) = \begin{cases} 0 & x = y \\ 0.5 & |(\mathrm{x},\mathrm{y}) \bmod 8| = 1 \\ 1 & else \end{cases} \tag{5.3.3b}$$

6. EXPERIMENTAL RESULTS

Training and test data are obtained from different written patterns that are taken from different people. 294 training data, 87 test data and total 381 hand-written character patterns are taken from 4 person. Chain code of each character is obtained. Training data are used to train three different classification methods (Back propagation, Learning Vector Quantization(LVQ), Template matching with two different distance functions). The performance of system is measured for each classification methods. The LVQ structure, has best performance, has 294 centroids. The MLP structure, has best performance, has 196 neuron in first layer, 200 neuron in second layer and 23 neuron in output layer. The performance measures are given in Table 1. The highest performance is measured on Template Matching using second distance function with %70.

Table1. Comparison of Classification Methods

LVQ(with 294 centroids)	BP(MLP 196-200-23)	TEMP (f_1)	TEMP (f_2)
%47	%34	%64	%69

References

[1] H. Freeman. ``On the encoding of arbitrary geometric configurations''. *IRE Trans. on Electronic Computers*, Vol. EC-10, pp. 260--268, 1961.
[2] H. Freeman. ``Computer processing of line-drawing images''. *ACM Computing Surveys*, Vol. 6, No. 1, pp. 57--97, March 1974.
[3] Rumelhart, D.E. & McClelland,J.L.,"Parallel Distributed Processing:Explorations in the Microstructure of Cognition, MIT Press, 1986.
[4] Lippmann,R.P.,"An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, April 1987.
[5] http://cbl.leeds.ac.uk/nikos/pail/intml/subsection3.11.4.html
[6] http://prlab.ee.memphis.edu/frigui/ELEC7901/LVQ/lvq.html
[7] http://www.engr.sjsu.edu/~knapp/HCIRODPR/PR_simp/template.htm