A WEIGHTED COMPETITIVE LEARNING METHOD EXTRACTING SKELETON PATTERN
FROM JAPANESE KANJI CHARACTERS

Kenji NAKAYAMA          Takuo KATO

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: A weighted competitive learning (WCL) method was proposed by authors for extracting skeleton patterns from digit and alphabet characters. The extracted pattern is essential in character recognition. It can satisfy the following important requirements. (a) Insensitive to irregular edge lines. (b) Non-structure patterns are not extracted. (c) Insensitive to non-uniform line width. (d) Line information should be held even though the line width widely changes in a character.

In this paper, the previous WCL method is improved for application to more complicated characters, such as Japanese Kanji characters. Furthermore, a PDP model, implements the WCL method, is provided.

I INTRODUCTION

Neural networks (NN) have been successfully applied to signal processing, pattern recognition and others based on waveform analysis [1]. Handprinted and handwritten character recognition are also important application fields of NNs. Many methods have been proposed [2]-[5]. However, performance of recognition are easily affected by distortion, non-uniform line width and blurred lines. Therefore, it is very important to extract essential structure of the character pattern to be recognized.

Conventional skeletonizing methods are based on shaving bold lines. Therefore, when line width is non-uniform, and blurred lines are included, it is difficult to obtain accurate skeleton patterns. Extracting unnecessary structures and missing important parts easily occur [5],[6].

A weighted competitive learning (WCL) method was proposed by authors for extracting essential structure of character patterns. It is insensitive to the above distortions. The idea behind this method is quite different from existing methods.

In this paper, the WCL method is improved in order to be applied to more complicated patterns, such as Japanese Kanji characters. Furthermore, implementation of this method based on a PDP model is discussed. Computer simulation using many distorted characters is demonstrated.

II BASIC STRATEGY

In handprinted or handwritten character recognition, requirements for skeletonization can be summarized as follows:
(a) Insensitive to irregular edge lines.
(b) Non-structure patterns are not extracted.
(c) Insensitive to non-uniform line width.
(d) Line information should be held.

The WCL method has been developed so as to satisfy the above require-
ments [7],[8]. Regarding (a) and (b), unnecessary pattern information is removed by representing some region of the pattern by a representative point, denoted 'RP'. This is a sort of 'vector quantization'. In order to optimize the RPs, the competitive learning was employed [9].

For the requirement (c), the range, covered by a RP, is adaptively changed according to the line width. Therefore, this method is called 'weighted competitive learning (WCL)'. The winner is evaluated using this region, that is a weighting factor.

The condition (d) is satisfied by connecting the RPs along the pattern and also through the border of the regions covered by them.

III WEIGHTED COMPETITIVE LEARNING

3.1 Generation and Movement of Representative Points

The WCL method [7],[8] is described for reader's reference here.

A pixel of a pattern is abbreviated as 'point' in this paper. A pattern and a pixel are denoted P and p_i, respectively. At the initial state, there is no RP. In the learning process, RPs are generated and moved toward the optimum position. The process is described step by step in the following.

1. One point p_i of the pattern is randomly selected.
2. Look for the RP, whose region includes this point. When a RP is found, this RP is shifted toward the above point by

   \[ \delta_{i_k} = \mu (p_i - r_k) \]  

   \( \mu \) is a learning coefficient. Where p_i and r_k are assumed to be 2-dimensional vectors, expressing the coordinates of the ith pixel of P and RP_k, respectively.

If there is no such a RP, then a new RP is generated on the same position as the point p_i.

3. Repeat the above steps for all points in the pattern.

4. When the RPs locate close to each other, as shown in Eq.(2), after shifting, they are combined, resulting in a new RP.

   \[ |r_k - r_{k'}| < C(r_k + r_{k'}) \]  
   \[ r_k = (r_k + r_{k'})/2 \]  

   |x| is the Euclidian distance of a vector x.

The process including (1) through (4) is counted as one iteration. The above steps are repeated until RPs converge to the stable state. How to evaluate convergence will be described later. After the RPs reach the optimum positions, their movement become very small.

In this method, some parameters, used in adjusting the regions, are determined before the learning. However, the number of RPs and their location are automatically optimized through the WCL.

3.2 Adjusting Region Covered by Representative Point

In order to extract the precise structural pattern from the distorted pattern with non-uniform line width, it is very important how to control the region covered by a single RP. If the region is narrow, then a zigzag line will be extracted. On the other hand, when the region is wide, fine structure cannot be extracted. This relation is shown in Fig.1.

In the WCL method, the region is adjusted according to the line width of each part of the pattern. Length of the lines crossing the RP are measured as shown in Fig.2.

They include the horizontal and vertical lines, and slopes with ±45
3.3 Convergence of Learning

Efficiency of the RPs is evaluated the following error function.}

\[ E_k = \sum \frac{|r_k - p_k|}{R_k} \]  
\[ E = \sum E_k \]  

$p_k$ is a point included in the RP$_k$ region. This error function can be reduced as the RPs approach to the optimum positions. Finally, the optimum allocation of all RPs will produce the minimum error.

Since the minimum value of $E$ is slightly different for the distorted patterns. Therefore, instead of $E$, the relative change of $E$ is used.

\[ \Delta E = \left| \frac{E(n) - E(n-1)}{E(n)} \right| < \varepsilon \]  

$E(n)$ is the error at $n$ iteration. After $\Delta E$ satisfies the above condition, the learning process is regarded as converges. $\varepsilon$ takes a small value. Its actual value is also determined by experience. Figure 4 shows an example of a learning curve.

3.4 Linkage of Representative Points

The optimized RPs are linked along the pattern and through the border of the regions. Figure 5 shows an example of linkage of the RPs.

As shown in Fig.4, a triangle is some time issued at the junction point. This is incorrect structure, and is replaced by a junction point as shown in Fig.6.
Suppose a selected point $p_i$ is included in the region of $RP_k$. If a straight line connecting both points $p_i$ and $r_k$ crosses non-pattern region, then this point $p_i$ is not included in the $RP_k$ region. An example is shown in Fig.6. $\bigcirc$ is a point locates on the pattern, $\bigcirc\bigcirc$ is a selected point $p_i$, and $\bigcirc$ is $RP_k$. In this example, $\bigcirc\bigcirc$ is not represented by $\bigcirc$.

$$\begin{array}{c}
\bigcirc\bigcirc \\
\bigcirc \bigcirc \\
\bigcirc \\
\bigcirc \bigcirc \bigcirc \\
\bigcirc \bigcirc \bigcirc \bigcirc \\
\end{array}$$

Fig.7 Example of point $\bigcirc\bigcirc$, which is not included in region of $RP\bigcirc$.

### 4.3 Lower Bound of Region

In a fine pattern, the radius $R_k$ of the region some time takes 1. This narrow region is not always useful. Therefore, the lower bound for the region is set.

$$R_k \geq R_{min}$$  \hspace{1cm} (8)

$R_{min}$ is used for all RPs, regardless of the line width. $R_{min}=2\sim3$[dots] can be widely used.

### 4.4 Overlap Regions

When the selected point $p_i$ is included in several RP regions, the winner is determined by a ratio of the distance between $p_i$ and $r_k$ and $R_k$.

$$d_{1k} = |p_i - r_k|$$  \hspace{1cm} (9)

$$\eta_{1k} = d_{1k}/R_k$$  \hspace{1cm} (10)

The RP, which has the minimum $\eta_{1k}$, is selected as the winner.

### V IMPLEMENTATION OF WCL METHOD

#### 5.1 Network Structure and Operation

The WCL method can be implemented using a PDP model. It consists of the input layer and the competitive layer.
as shown in Fig. 8.

![Diagram of Competitive Layer](image)

**Input Layer**

Fig. 8 A PDP model for WCL method.

The input layer expresses the character patterns to be processed. Each unit outputs a 2-dimensional vector \( p_i = (p_{i1}, p_{i2}) \). This vector is normalized to unity by adding the following auxiliary element \( a_i \) \[10\].

\[
D > \max_i |p_i| \quad \text{(11)}
\]

\[
a_i = (D^2 - |p_i|^2)^{1/2} \quad \text{(12)}
\]

\[
p'_i = \frac{(a_i, p_{i1}, p_{i2})}{D} \quad \text{(13)}
\]

D is constant for all \( p_i \). The connection weight \( r'_k \) also includes \( r_k = (r_{k1}, r_{k2}) \) and an auxiliary element. The input of the competitive layer is an inner product of \( p'_i \) and \( r'_k \).

\[
\text{net}_k = p'_i \cdot r'_k \quad \text{(14)}
\]

The maximum \( \text{net}_k \) indicates the shortest distance between \( p_i \) and \( r_k \). Therefore, if the \( k' \)-th unit in the competitive layer is the maximum input,

\[
\text{net}_k = \max_k \{\text{net}_k\} \quad \text{(15)}
\]

then it can be the winner.

**5.2 Adjusting Connection Weights**

If \( R_k \) becomes the winner, its connection weights are updated.

\[
r_k(n) = r_k(n-1) + \mu (p_i - r_k(n-1)) \quad \text{(16)}
\]

\( r_k(n) \) is the component of \( R_k \) at the \( n \)-th update cycle.

**VI SIMULATION RESULTS**

**6.1 Alphabet and Digit Patterns**

Figure 9 shows examples of alphabet and digit patterns \((a1) \sim (c1)\), and their skeleton patterns extracted by conventional methods \((a2) \sim (c2) [11]\), and \((a3) \sim (c3) [6]\), and \((a4) \sim (c4)\) the WCL method. The WCL method can successfully extract the essential structural patterns.

![Examples of Alphabet and Digit Patterns](image)

**6.2 Alphabet Patterns with Edge Noise**

Figure 10 shows alphabet patterns with edge noise (a). The skeleton patterns obtained by (b) the conventional [6] and (c) the WCL method are shown. The WCL method can provide essential patterns.

**6.3 Japanese Kanji Characters**

Figure 11 shows examples of Japanese Kanji Characters and their skeleton pattern extracted by the improved WCL method. \( a = 2 \) and \( R_{m1,n} = 2 \) are used. Efficiency of the proposed
WCL method can be confirmed.

![Alphabet patterns](image1)

Fig. 10 Examples of alphabet patterns with edge noise and their skeleton pattern.

VII CONCLUSIONS

The WCL method has been improved for applications to complicated characters, such as Japanese Kanji characters. Furthermore, its implementation based on a PDP model has been proposed. Through computer simulation efficiency of the proposed WCL method has been confirmed.

REFERENCES


![Japanese Kanji characters](image2)

Fig. 11 Examples of Japanese Kanji characters and their skeleton patterns extracted by WCL method.