High Speed Rough Classification for Handwritten Characters using Hierarchical Learning Vector Quantization

Yuuji Waizumi[†] Nei Kato[†] Kazuki Saruta[‡] Yoshiaki Nemoto[†]

†Graduate School of Information Sciences TOHOKU Univ.
Aza,Aoba,Aramaki,Aoba-ku,Sendai.980-77 Japan
‡Faculty of Human Literature and Social Sciences YAMAGATA Univ.
1-4-12,Kojirakawa-machi,Yamagata-shi,Yamagata,990 Japan.

Abstract

Today, high accuracy of character recognition is attainable using Neural Network for problems with relatively small number of categories. But for large categories, like Chinese characters, it is difficult to reach the neural network convergence because of the "local minima problem" and a large number of calculation. Studies are being done to solve the problem by splitting the neural network into some small modules. The effectiveness of the combination of Learning Vector Quantization(LVQ) and Back Propagation(BP) has been reported. LVQ is used for rough classification and BP is used for fine recognition. It is difficult to obtain high accuracy for rough classification by LVQ itself. In this paper, to deal with this problem, we propose Hierarchical Learning Vector Quantization(HLVQ). HLVQ divides categories in feature space hierarchically in learning procedure. The adjacent feature spaces overlap each other near the borders. HLVQ possesses both classification speed and accuracy due to the hierarchical architecture and the overlapping technique. In the experiment using ETL9B, the largest database of handwritten character in Japan, (includes 3036 categories, 607,200 samples), the effectiveness of HLVQ was verified.

1 Introduction

It is possible to obtain high accuracy using BP [1] for problem with relatively small number of categories, like figures and the English alphabet, but it is difficult for Chinese characters, which has thousands of categories, because of the "local minima problem" and BP's learning speed. To deal with this problem, the technique which combining LVQ and BP [2]-[5] has been proposed (see Fig.1). For this type of network to work effectively, it is necessary to gain high accuracy for rough classification.

LVQ can learn in high speed for certain applications, but for Chinese characters, which include thousands of categories, learning cannot terminate because of the calculation quantity. To solve this problem, it would be ideal to assign multiple categories to each codebook vector, but the generation method for teacher signals and the management of boundary data for adjacent clusters have not been established. Due to the reasons mentioned above, conventional LVQ for rough classification is inadequate.

In this paper, we propose Hierarchical Learning Vector Quantization(HLVQ). HLVQ possesses the following features, which allows it to gain high accuracy and speed for rough classification.

feature 1: add neurons hierarchically

feature 2: divide the feature space into a few regions hierarchically

feature 3: overlap the adjacent region near the border

Feature 1 and 2 is to gain speed for classification. In contrast to the conventional LVQ, which classifies categories in one layer, the HLVQ divides the feature space at $(N+1)$-th layer by using the feature space divided at N-th layer. This can reduce the calculation in classification significantly.

Feature 3 is to gain high accuracy for classification. It is difficult to gain high accuracy by dividing the feature space with a large number of categories into a few Voronoi regions represented by a few codebook vectors. The reason is that the feature space with a large number of categories has complicated borders. HLVQ solves this problem by dividing the feature space with overlapping near the border.

In the recognition experiment using ETL9B, a recognition rate of 99.6% is obtained when selecting an average of 170.0 categories from the 3036 categories. The average number of calculation to classify one unknown sample is 107.51. Compared to the K-means method under the condition of 256 centroids, the recognition rate is 98.03% when selecting an average of 193.5 categories from the 3036 categories.

In this paper, the details of HLVQ is described in Section 2. In Section 3, the recognition experiment is explained and reference is made to K-means for comparison purpose. In Section 4, we discuss the results obtained in Section 3. Finally in Section 5, we point out future works for this research.

0-8186-7898-4/97 $10.00 © 1997 IEEE

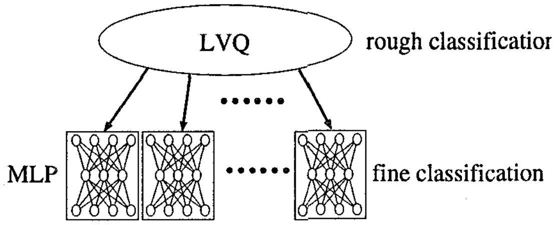

Figure 1: Network of Combining LVQ and MLP

2 Hierarchical Learning Vector Quantization(HLVQ)

2.1 Basic Concept

Hierarchical Learning Vector Quantization(HLVQ) divides a feature space hierarchically by a few codebook vectors. The regions divided at a layer is further divided at the lower layer hierarchically. Since a feature space with many categories has complicated border, HLVQ divides the feature space by the overlapping technique. The overlap is determined by the '*window*' used in LVQ2 [6] [7] (Fig. 2).

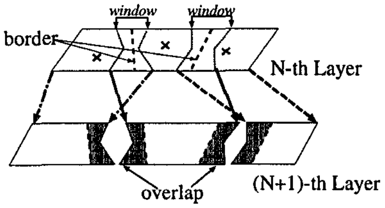

Figure 2: Dividing Feature Space by HLVQ

2.2 Structure of Network

HLVQ is constructed by neurons which have multiple codebook vectors. Neurons are arranged hierarchically in a tree-like structure(Fig.3).

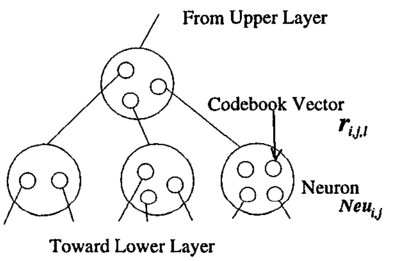

Figure 3: Structure of Network

2.3 Learning Algorithm

The learning algorithm is shown in Fig.4. In Fig.4, [Learning*] is the algorithm for each neuron shown in Fig.5. HLVQ learns while adding neurons hierarchically. The learning process is continued until all neurons satisfying the terminating condition. The learning procedure is described as follows:

[step1] Initialize Codebook Vectors

The codebook vectors in a neuron are initialized by the average vector of the training vectors in the region which will be divided by the neuron. This is to prevent the phenomenon that a particular codebook vector always wins and the other codebook vectors are hindered in the learning process.

[step2] Unsupervised Learning Vector Quantization

In the case of character recognition, supervised learning is available because all samples are labeled. To assign one codebook vectors to represent multiple categories, it is necessary to cluster the training vectors. For that purpose, unsupervised learning is conducted for those codebook vectors initialized in [step1].

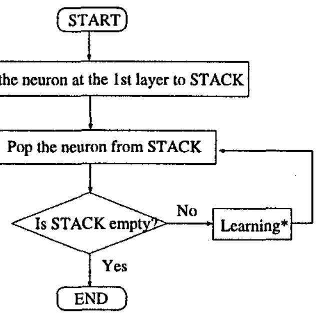
START
Push the neuron at the 1st layer to STACK
Pop the neuron from STACK
Is STACK empty?
No
Learning*
Yes
END

Figure 4: HLVQ Learning Algorithm

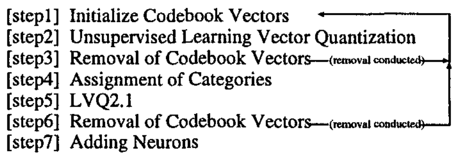
[step1] Initialize Codebook Vectors
[step2] Unsupervised Learning Vector Quantization
[step3] Removal of Codebook Vectors—(removal conducted)
[step4] Assignment of Categories
[step5] LVQ2.1
[step6] Removal of Codebook Vectors—(removal conducted)
[step7] Adding Neurons

Figure 5: Learning Algorithm for Each Neuron

[step3] Removal of Codebook Vectors

The number of codebook vectors is important for clustering. But the way of obtaining the suitable number has not been established, yet. In many cases, the

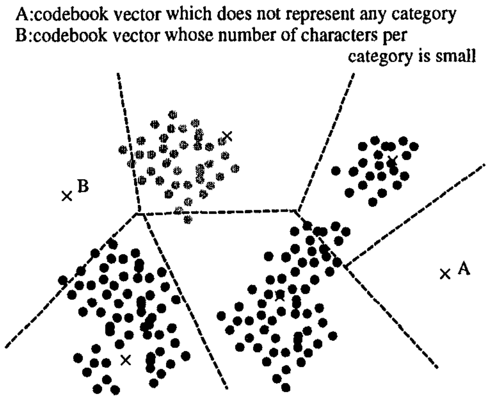
A:codebook vector which does not represent any category
B:codebook vector whose number of characters per category is small
B
A

Figure 6: Conditions of Removal

techniques of adding or removing [8]-[11] the number of codebook vectors is adopted to fix the number of codebook vectors. HLVQ adopts the way of removing codebook vectors to obtain suitable clusters. Codebook vectors satisfying the following conditions are removed. (Fig.6)

Condition of removal

When a neuron has more than three codebook vectors:

$$\begin{aligned} condition\ 1 &: CharN(i,j,l) = 0 & (1) \\ condition\ 2 &: \frac{CharN(i,j,l)}{CateN(i,j,l)} < \varepsilon & (2) \\ \varepsilon &: constant & (3) \end{aligned}$$

where $CharN(i,j,l)$ is the number of characters represented by codebook vector r_l in $neuron_j$ at i-th layer, $CateN(i,j,l)$ is the number of categories represented by codebook vector r_l in $neuron_j$ at i-th layer, and they are calculated using the 1-NP(nearest prototype) rule.

Condition 1 means that there is no training vector in the region represented by the codebook vector; thus, making it useless. In character recognition, a codebook vector should represents the cluster of vectors for the corresponding categories, but the codebook vector satisfying condition 2 is not representative of any particular categories, and can be regarded as useless. If the above conditions are satisfied, these vectors are removed and the algorithm goes back to [step1].

[step4] Assignment of Categories

After the unsupervised learning in [step2], supervised learning will be conducted. For supervised learning, it is necessary to assign categories to the codebook vectors. Each codebook vector represents some categories for rough classification. Due to the fact that there is no conventional category assignment in LVQ, we will use the 1-NP rule for the network created by unsupervised learning in [step1] and [step2].

[step5] LVQ2.1

The supervised learning(LVQ2.1) will be conducted using the teacher signals generated by [step4].

[step6]Removal of Codebook Vectors

[step6] is the same as [step3].

[step7] Adding Neurons

Adding neurons is the most characteristic process in HLVQ. HLVQ divides the feature space hierarchically by a few codebook vectors. Since the borders of Voronoi region created by the codebook vectors are extremely simple, and it is expected that the ideal borders between categories are very complicated, HLVQ does not define a definite border but divides the feature space with overlapping regions(Fig.2). The width of overlap is defined by W and (4).

Overlap width

$$w = W + (1 - W)(1 - \frac{c(i,j)}{CATE}) \qquad (4)$$

where

w : Overlap width
W : constant
$c(i,j)$: the number of categories learned by $neuron_j$ at i – th layer
$CATE$: total number of categories

If the trainig vector v fall into the region as defined by (5), v is belonged into both the nearest codebook vectors r_1 and r_2.

Overlap

$$d_1/d_2 > w \qquad (5)$$

where

d_1,d_2 are the distances between v and r_1,r_2 respectively.

If the number of categories of trainig vectors represented by codebook vector r_l is greater than the terminating condition,$StopCnum$, a child neuron of r_l is added at lower layer and learned using the algorithm shown from [step1] to [step7].

2.4 Classification Algorithm by HLVQ

In HLVQ, unknown data x is classified while it descends the tree whose nodes are neurons. The path descending the tree can split into no more than three paths at a neuron.

Assuming that r_1,r_2,r_3 are the nearest codebook vectors to x, $d_1, d_2, d_3 (d_1 < d_2 < d_3)$ are distances between x and r_1,r_2,r_3 respectively, and only r_1 and r_3 have child neuron at lower layer. r_1 always sends x to its child neuron. If the condition, as defined by (6), is satisfied, r_2 outputs the candidates. If the condition, as defined by (7), is satisfied , r_3 sends x to its child neuron, too.

Condition of splitting path

$$d_1/d_2 > w_2 \quad (6)$$
$$d_1/d_3 > w_3 \quad (7)$$

where

$$w_2, w_3 \ : \ constant$$

The number of neurons through which each unknown sample passes increases as the value of w_2,w_3 increases. Therefore the number of candidates is greater.

The black and gray circles in Fig.7 indicate selected codebook vectors. The black circles represent output candidates, while the gray circles indicate codebook vectors through which the unknown data is passed onto the child neuron.

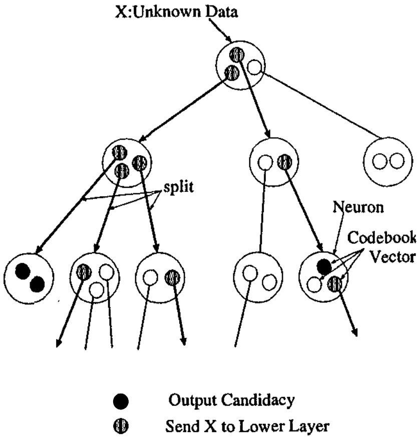

Figure 7: Classification Algorithm by HLVQ

3 Classification Experiment

In this section, we discuss classification experiment for ETL9B[12]. ETL9B is the largest handwritten character database published by Electrotechnical Laboratory of Japan in 1985. There are 200 data sets where each set has 3036 categories (one sample per category). As preprocessing, normalization, contour extraction and feature vector extraction[13] are conducted. HLVQ is compared with K-means method, which is commonly used for rough classification. The number of centroid for K-means is decided based on the desired number of distance calculations. The odd-numbered sets in ETL9B are used as training samples for HLVQ, the average vectors of the odd-numbered sets in ETL9B are used as training samples for K-means. The even-numbered sets in ETL9B are used as unknown samples for both HLVQ and K-means.

Table 1: The number of neurons and codebook vectors

	the number of neurons	the number of codebook vectors
non-overlap	669	3764
with overlap	1441	7525

3.1 Classify All 3036 Categories in ETL9B

3.1.1 Parameters

Parameters of HLVQ are decided as follows: The number of codebook vector in a neuron before learning is 10, condition of destruction $\epsilon = 2(2)$, the overlap width of adjacent regions $W = 1.0(non\text{-}overlap), 0.96(with hoverlap)$(ean.4), condition of termination of learning, *StopCnum*, equals 50. The number of centroids for K-means is 256. The criterion of split(6,7) is from 0.84 to 1.00.

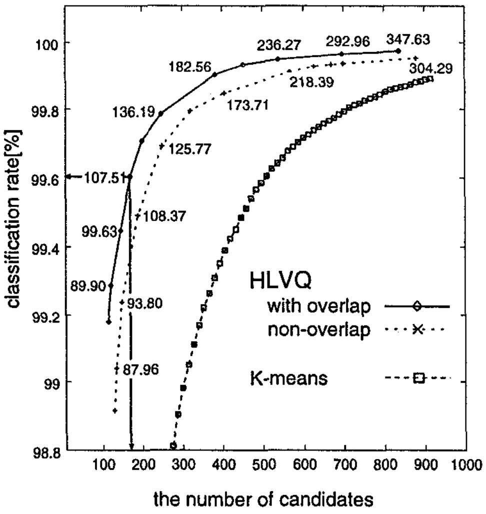

Figure 8: Accumulated Classification Rate for All 3036 Categories in ETL9B

3.1.2 Experimental Results

The results are shown in Fig.8. The numbers in the figure are the average number of distance calculations to classify one unknown sample by HLVQ. The highest classification rate is gained by HLVQ with overlap. When selecting an average of 170.07 categories from the 3036 categories, the classification rate reachs 99.60%. The average calculation to classify one unknown sample is 107.51. The classification rate of K-means reaches 98.03% when selecting an average of 193.45 categories from the 3036 categories. The number of calculations to classify one unknown sample is the same as the number of centroids. The number of neurons and codebook vectors are shown in Table 1.

4 Discussion

The overlap of adjacent regions defined by $W(4)$ indicates the ability to decrease category in each neuron. By making the overlap of adjacent regions smaller, the ability to decrease category increases in each neuron. So, the number of neurons and codebook vectors differs for with or without overlap, as shown in Table 1. The number of neurons and codebook vectors is greater with overlap, but the classification rate achieve with overlap higher than that of without overlap.

5 Conclusion

In this paper, we have proposed Hierarchical Learning Vector Quantization for rough classification of large scale handwritten Chinese characters. HLVQ has the following features:

feature 1: add neurons hierarchically

feature 2: divide the feature space into a few regions hierarchically

feature 3: overlap the adjacent region near the border

In the recognition experiments, we demonstrated that HLVQ can gain high accuracy and speed in rough classification for Chinese characters.

The number of candidates for one unknown data is shown in Fig.9. Fig.9 shows the example to select an average of 188.21 categories from 3036 categories (99.49%) by non-overlap HLVQ. We can see from Fig.9 that variance of the number of candidates is large, and still needs improvement.

To solve the problem,

- improve the learning algorithm for codebook vectors
  - how to assign categories to codebook vector
  - how to update codebook vector
- find a better way to define the parameters.

are being studied.

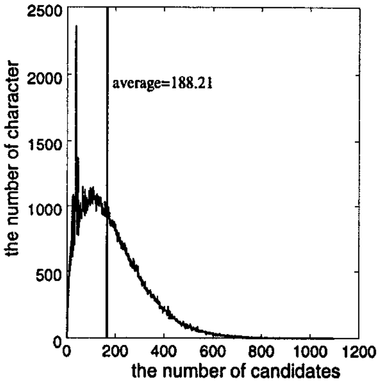

Figure 9: Number of Candidates

References

[1] Rumelhart D.E., Macclelland J. E. and PDP-research group: "Parallel Distributed Processing",1,2 MIT Press, Cambridge, MA(1986).

[2] Yoshio MORI ,"Aiming at a large scale neural network", TECHNICAL REPORT OF IEICE, **PRU88**-59,pp.87-94 (1988)

[3] T.Kohonen: "Self-organization and Associate Memory(2nd Edition)", Spring-verlag, pp.199-202 (1989).

[4] Masayuki ARAI , Kenzo OKUDA and Jyuichi MIYAMICHI , "Thousands of Hand-Written Kanji Recognition by "HoneycombNET-II"", Transactions of IEICE **J77-D-II**, No. 9, pp.1708-1715 (1995-09).

[5] Kazuki SARUTA,Nei KATO, Masato ABE and Yoshiaki NEMOTO, "High Accuracy Recognition of ETL9B Using Exclusive Learning Network-II(ELNET-II)," IEICE TRANS. INF. & SYST., VOL.E79-D,No.5,MAY 1996.

[6] Teuvo Kohonen,"Improved version of learning vector quantization" In *Proceedings of the International Joint Conference on Neural Networks*, pp.545-550, San Diego,June 1990.

[7] Teuvo Kohonen,"Staistical pattern recognition revisited." In *Advanced Neural Computers*, pp.137-144,1990.

[8] B.Frizke,"Growing cell structures - A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no.9, pp.1441-1460,1994.

[9] J.Diederich,"Connectionist recruitment learning, " in *Proc. 8th Eur. Conf. Artificial Intelligence.* Londo:Pitman, 1988, pp.351-356

[10] Y.Q.Chen,D.W.Thomas,and M.S.Nixon,"Generating-shrinking algorithm for learning arbitrary classification," *Neural Networks*,vol.7, no.9, pp.1477-1489, 1994.

[11] M.C.Moze and P.Smolensky, "Using relevance network size automatically," *Connect. Sci.*, vol.1, no.1, pp.3-16, 1989.

[12] Taiichi SAITO, Hiromitsu YAMADA and Kazuhiko YAMAMOTO, "On the Data Base ETL9B of Handprinted Character in JIS Chinese Characters and Its Analysis", Transactions of IEICE **J68-D**, No. 4, pp.757-772 (1985).

[13] N.Sun,M.Abe and Y.Nemoto,"A Handwritten Character Recognition System by Using Imaproved Directional Element Feature and Subspace Method," IEICE Trans. **J78-D-II**, No. 6, pp.922-930, 1995.