Proceedings of the First International Conference on Machine Learning and Cybernetics, Beijing, 4-5 November 2002

REGION-MAPPING NEURAL NETWORK MODEL FOR PATTERN RECOGNITION

YAN-LAI LI [a], KUAN-QUAN WANG [a], DAVID ZHANG [b]

[a] Department of Computer Science & Engineering, Harbin Institute of Technology (HIT), Harbin, 150001,China
[b] Department of Computing, Hong Kong Polytechnic University, Kowloon, Hong Kong
E-MAIL: yanlai_l@yahoo.com, kqwang@hope.hit.edu.cn, csdzhang@comp.polyu.edu.hk

Abstract:

In general, the process for multilayer feedforward neural network in pattern recognition is composed of two phases: training and classifying. The aim of the training phase is to make the network output meet the desired output given by the training patterns as possible. It demands a map of point to point, which is so strict that it often causes the criterion inconsistence between training and classifying. Consequently the recognition rate would be decreased. Region-mapping model has changed the output space from one point to a certain supervisor region so that it has overcome the shortcoming of inconsistent problem between training and testing as common multilayer perceptron (MLP) does. Furthermore, it has saved much computing time by mapping the input data to an output area rather than an output point. This paper presents a Region-mapping model with quarter hyper globe as supervisor region. The gradient decent algorithm is applied to this model. In order to illustrate the effect of our propounded model, a hand-written letter recognition problem is put into experiment. Moment invariant features are used as input parameters. The simulation results show that the region-mapping model has much better characteristics than those common multiplayer perceptrons. Also, the quarter hyper globe rule is more reasonable than the hypercube one.

Keywords:
Region-mapping model; Supervisor region; Neural network; Pattern recognition

1 Introduction

Multilayer perceptron (MLP), one of the most popular neural network models, is a powerful tool in pattern recognition problem. When applying in the pattern recognition, its procedure is divided into two phases generally [1]. The first one is training or learning phase. Work of this phase is to calculate the weights of given network through the training pattern set. Assume that the number of input and output units is M and N, respectively. Also suppose there are K patterns belonging to C classes. The subscript set is I^c. Then the pattern set can be described as follows:

$$\bigcup_{c=1}^{C}\{(x_k^c, y_k^c) \big| x_k^c \in R^M, y_k{}^c \in R^N, k \in I^c\} \qquad (1)$$

where $x_k{}^c$ denotes the characteristic parameter vector, $y_k{}^c$ is the kth code of class c. The aim of training is to make the output meet $y_k{}^c$ when $x_k{}^c$ is inputted through the network. But actually it is quite difficult to achieve this goal. So a permitted arbitrary small error limit ε is given. The training process is considered to be finished when the sum of error of all the patterns satisfy:

$$E = \frac{1}{2}\sum_c\sum_k (y_k - y^c)^2 < \varepsilon \qquad (2)$$

The intuitionistic meaning of (2) is that the training phase stops when all the outputs of patterns fall in the suitable adjacent regions of the corresponding supervised signal. These regions of four-class pattern recognition problem are shown as Fig. 1, from which we can see that the shapes of the four regions are irregular and uncertain.

The second phase is classifying. In this phase, output is computed by all the fixed weights saved by the training phase and the classification results are given. For the four-class case, the classification criterion is: given a certain small positive number ε, suppose that the desired output is binary code. If the output lies in the range of $[1-\varepsilon, 1]$ or $[0, \varepsilon]$, then the pattern can be classified correctly, otherwise the classification fails. The Fig.1 (b) shows the classification criterion. When the network output of the pattern falls in the square adjacent region of supervised signal with each side ε, the pattern is classified to this class. Otherwise, it cannot be correctly classified. It can be seen from Fig.1 (b) that the classification criterion is certain, and not in accordance with the training criterion shown in Fig.1 (a). It is very fuzzy to map the input patterns of a class to an irregular region in the output space during the training phase, while to choose a fixed and regular region as output space during the classifying phase. Such criterion is also a reason to cause low recognition rates. This phenomenon is called ***Criterion Inconsistence*** between training and classifying. The problem can be solved if we adopt a predefined region in the output space instead of just a supervisor signal to train the network. Once the output of network falls in the certain region, the error is set to zero. Therefore, a new model named Region-mapping is propounded.

The remained parts of this paper are arranged as follows: In section 2, the Region-mapping model and its algorithm is described. In section 3, we put this model in the application of English character and Arabian numbers recognition. Six invariant moments are calculated as feature parameters. Finally, a brief conclusion is drawn in section 4.

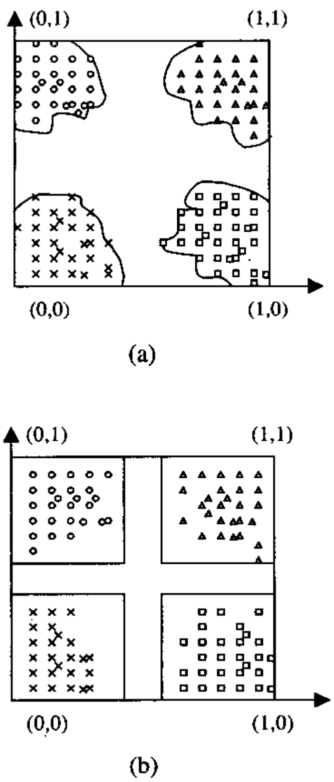

Fig.1. Sketch map of the criteria for the general MLP. (in two-dimension condition)

(a) The criteria for the training phase of MLP

(b) The criteria for the classifying phase of MLP.

2 Region-mapping Model

2.1 Description

Region-mapping model is first proposed by Wang in 2000 [2]. The model demands the character parameter region of the input space be mapped to a corresponding region of the output space, rather than a point, which is the essential difference between the region-mapping model and the general MLP. Suppose that the region formed by the character parameters belonging to c classes is denoted as $U^c \subset R^M$, the MLP model fulfills the following map:

$$f\left(U^c\right)= y^c,\ U^c \subset R^M,\ y^c \subset R^N, c=1, \cdots C \quad (3)$$

Where y^c is a kind of code of the cth class.

$$N = ceil(\log_2 C)\ , \quad (4)$$

where $ceil(x)$ denotes the smallest integer that is greater than or equal to x.

For the Region-mapping model, an output region V^c, called ***Supervisor Region***, should be predefined for each class. Its output space is $[0, 1]^N$. There are three principles to select a V^c:

1) $V^c \subset [0,1]^N$, and $\bigcup\limits_c^C V^c \subset [0,1]^N$.

2) $\min d(V^c,V^{c'}) \geq \sigma$, for each $c \neq c'$, where σ is a positive constant and $d(V^c,V^{c'})$ denotes the distance between V^c and $V^{c'}$.

3) $shape(V^c) = shape(V^{c'})$, for each $c \neq c'$, where $shape(V^c)$ means the shape of the predefined supervisor region.

The principles are to make sure that the output regions of each class can be separated strictly.

Theoretically, V^c can be arbitrarily selected as long as it satisfies the principles mentioned above. The supervisor region firstly proposed is hypercube for the considering of convenience of calculating and analyzing. However, it cannot reflect the distribute rule of the patterns so exactly. In this paper, we take quarter hyper-globe as V^c, the two-dimension sketch map of which is shown in Fig.2. Though it may make the computation a little complex, it is better in accordance with the distribute rule of the patterns. Moreover, it can be described more clearly.

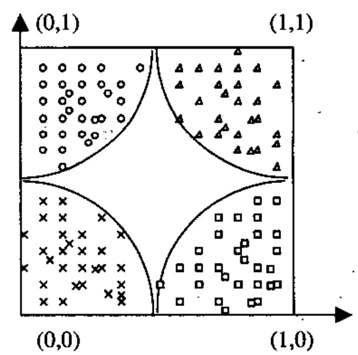

Fig.2. Two-dimension sketch map of a quarter hyper globe V^c (consistent rule for training and classifying).

After defining the supervisor region V^c, the criterion of both the training and the classifying phase is consistent.

Now, the pattern set needed by network training can be denoted as:

$$\bigcup_{c=1}^{C}\left\{\left\{x_k^c, V^c\right\} \middle| x_k^c \in R^M, V^c \subset R^N, k \in I^c\right\}. \quad (5)$$

Because essentially the neural network is a non-linear continuous map, it is more reasonable to map a region to a region than to a point, from the point of view of nonlinear transformation of function. The region-mapping model fulfills the following map:

$$f(U^c) \subset V^c \quad U^c \subset R^M,\ c=1, 2, \cdots, \text{C}. \quad (6)$$

2.2 Algorithm For Training the Region-mapping Model

Here we discuss only about the neural network with one hidden layer. The supervisor region V^c selected in this paper can be described by the radius WD^c and the vertexes $P^c = (p_1^c, p_2^c, \cdots, p_N^c)$, where p^c_i (i=1, 2, $\cdots N$) is binary code of the cth class, i.e., p^c_i is equal to '0' or '1'.

$$V^c = \left\{ (y_1, y_2, \cdots y_N) \left| \sqrt{\sum_i^N (y_i^c - p_i^c)^2} < WD^c \right. \right\}, \qquad (7)$$

where $WD^c = 0.5 - \varepsilon$, ε is a small positive number.

The algorithm of region-mapping model can be deduced on the basis of gradient decent algorithm, except that the error function has changed.

Usually, the error function adopted by the multilayer perceptron is:

$$E = \frac{1}{2} \sum_{k=1}^{K} d(y_k - \tilde{y}_k), \qquad (8)$$

where y_k is the actual output of the kth pattern x_k, while $\tilde{y}_k$ is the desired output. $d(y_k - \tilde{y}_k)$ is the square distance between y_k and $\tilde{y}_k$,

$$d(y_k - \tilde{y}_k) = \sum_{n=1}^{N} (y_{kn} - \tilde{y}_{kn})^2 . \qquad (9)$$

While the error function adopted by the region-mapping model is:

$$E = \frac{1}{2} \sum_{k=1}^{K} d(y_k, V_k), \qquad (10)$$

where $d(y_k, V_k)$ denotes the square distance between the point y_k and the supervisor region V_k. when the input x_k belongs to the cth class, $V_k = V^c$.

From the form of the error function, we can see that if the kth pattern belongs to the cth class, the error will be zero when the output falls into V^c. In contrast with the requirement of falling at a point, this condition has been relaxed much. This character can make the convergence speed up.

The weight adaptation is:

$$w_{ij}^{(l)}(t+1) = w_{ij}^{(l)}(t) + \eta_{ki}^{(l)}(t) \cdot \delta_{ki}^{(l)}(t) \cdot y_{kj}^{(l)}(t) + \alpha \cdot (w_{ij}^{(l)}(t) - w_{ij}^{(l)}(t-1)) \qquad (11)$$

where η is learning rate, α is momentum factor, $w_{ij}^{(l)}$ is the weight from the jth neuron of the l-1 layer to the ith neuron of the lth layer.

$$\delta_{ki}^{(l)} = \begin{cases} y_{ki}^{(l)} \cdot (1 - y_{ki}^{(l)}) \cdot E_{ki}, & l \in OutputLaye \\ y_{ki}^{(l)} \cdot (1 - y_{ki}^{(l)}) \cdot \sum_p \delta_{kp}^{(l+1)} \cdot w_{pi}^{(l+1)}, & l \in HiddenLaye \end{cases} \qquad (12)$$

where

$$E_{ki} = \begin{cases} 0, & \sqrt{\sum_i (y_i^c - p_i^c)^2} < WD^c \\ \left| \sqrt{\sum_i (y_i^c - p_i^c)^2} - WD^c \right|, & else \end{cases} \quad i = 1, 2, \cdots, N \qquad (13)$$

To accelerate the speed of the converging, most of the speed-up algorithms for MLP are available for the region-mapping model. Readers interested in it can deduce the formulations similarly with them.

2.3 Training Process

The whole training process can be concluded as:

1) Get the patter set as

$$\bigcup_{c=1}^{C} \left\{ (x_k^c, c) \left| x_k^c \in R^M, k \in I^c \right. \right\}. \qquad (14)$$

2) Define the output node number of the network according to the class number of the patterns or to the designing need. Define the V^c and get the training pattern set as (5).

3) Initialize the network weights and thresholds.

4) Input the pattern set and calculate forwards to obtain the error sum. If the error sum is zero, the training finishes. Else go to 5).

5) Set k=1.

6) Input pattern k and calculate forwards.

7) Calculate δ_{ki} of the output layer and the hidden layer according (12). Adapt the network weights and thresholds according to (11).

8) If k = K, go to 4); if k < K, set k = k+1 and go to 6).

2.4 The Training And The Classifying Criterion of The Region-mapping Model

It can be seen from the training algorithm that the network training stops when the error function is aero. That means that all the training patterns fall into the corresponding supervisor region.

In contrast with the classifying process of the MLP, the actual classifying process takes the supervisor regions V^c, c=1, 2, $\cdots C$ as input parameter as well as the network weights and thresholds.

If the network output of the pattern to be tested falls into a certain supervisor region, it can be classified to this class, else the pattern can not be classified correctly.

Therefore, the classifying criterion keeps in consistent with the training criterion, which guarantees the recognition rate.

3 Character Recognition with Region-mapping Model

The Character recognition system with Region-mapping (R-M) model consists a preprocessor and a neural network classifier as shown in Fig.3.

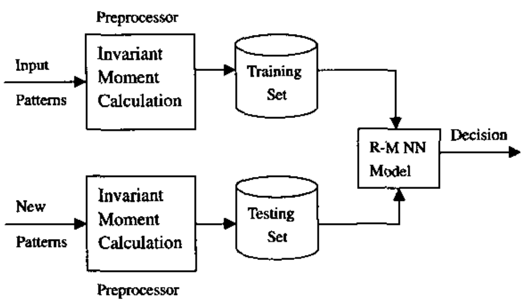

Fig.3. The character recognition system with R-M neural network model

For various reasons, the patterns may be deformed due to translating, rotating or scaling. The aim of the preprocessor is to create an output for an input pattern in such a way so that the preprocessed output remains unchanged or almost unchanged even if the input pattern is rotated, scaled and translated.

In this experiment, the pattern database consists twenty-six lower case letters a - z and ten Arabian numbers 0 – 9. Each class generates fifty image patterns, each is presented by a 13*13 binary pixel where a '1' presents an *on-pixel* and a '0' presents an *off-pixel*. Altogether there are 900 training patterns and another 900 testing patterns.

3.1 Moment Invariant Calculation

For each image, its gray distribution function is $f(x, y)$. The $(p+q)$ order moment is defined as:

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y). \qquad (15)$$

The center moment of m_{pq} is:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x, y), \qquad (16)$$

where $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$.

Furthermore,

$$\mu_{10} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^0 f(x, y) = m_{10} - \frac{m_{10} \cdot m_{00}}{m_{00}}$$

$$\mu_{11} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^1 f(x, y) = m_{11} - \frac{m_{10} \cdot m_{01}}{m_{00}}$$

$$\mu_{20} = \sum_x \sum_y (x-\bar{x})^2 (y-\bar{y})^0 f(x, y) = m_{20} - \frac{m_{10} \cdot m_{10}}{m_{00}}$$

$$\mu_{02} = \sum_x \sum_y (x-\bar{x})^0 (y-\bar{y})^2 f(x, y) = m_{02} - \frac{m_{01} \cdot m_{01}}{m_{00}}$$

$$\mu_{30} = \sum_x \sum_y (x-\bar{x})^3 (y-\bar{y})^0 f(x, y)$$

$$= m_{30} - 3\bar{x} \cdot m_{20} + 2m_{10} \cdot \bar{x} \cdot \bar{x}$$

$$\mu_{12} = \sum_x \sum_y (x-\bar{x})^1 (y-\bar{y})^2 f(x, y)$$

$$= m_{12} - 2\bar{y} \cdot m_{11} - \bar{x} \cdot m_{02} + 2\bar{y} \cdot \bar{y} \cdot m_{10}$$

$$\mu_{21} = \sum_x \sum_y (x-\bar{x})^2 (y-\bar{y})^1 f(x, y)$$

$$= m_{21} - 2\bar{x} \cdot m_{11} - \bar{y} \cdot m_{20} + 2 \cdot \bar{x} \cdot \bar{x} \cdot m_{01}$$

$$\mu_{03} = \sum_x \sum_y (x-\bar{x})^0 (y-\bar{y})^3 f(x, y)$$

$$= m_{03} - 3\bar{y} \cdot m_{02} + 2m_{01} \cdot \bar{y} \cdot \bar{y}\ . \qquad (17)$$

The normalized center moment is

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^r},\ p+q = 2, 3, \cdots \qquad (18)$$

where $r = \frac{p+q}{2} + 1$.

The following invariant moments are calculated by the second and third order moments [3-4]:

$$\phi_1 = \eta_{20} + \eta_{02} \qquad (19)$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \qquad (20)$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \qquad (21)$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \qquad (22)$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \cdot [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \qquad (23)$$

$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{03} + \eta_{21}) \qquad (24)$$

The above ϕ_1 - ϕ_6 has invariant characteristics with rotation, translation and scale.

3.2 Simulation Result

After preprocessing the input pattern data, the invariant moments are passed the network as the input parameters.

In order to investigate the performance of the proposed Region-mapping model and to evaluate the recognition accuracy of the system, it is first trained with a set of training data and then tested with a different set of testing data. Each pattern is passed through the preprocessor and produces outputs. These preprocessed outputs are used to train the neural network. The test data set consists 900 test patterns, which are the rotated and translated or scaled versions of the exemplar pattern. The general MLP with the famous BP algorithm, and the Region-mapping model with hypercube supervisor region is taken into consideration as comparisons with the proposed model in this paper.

In this experiment, the neural network structure is composed of six input nodes, six output nodes and 16 hidden nodes. The number of training pattern is 900. All the models are trained by the famous gradient descend algorithm. Let learning rate $\eta = 0.8$ and the momentum factor $\alpha = 0.1$ to get best behavior of all the cases.

The contrast results are listed in Table 1, from which we can see that the capability of Region-mapping model is much better than that of the general MLP and the Region-mapping model with hypercube supervisor region.

Table 1. Comparison results for character recognition with different models and different V^c

Model	Learning Epochs	Error	Recognition Rate
MLP	120074	0.052	87.14%
R-M 1	8950	0.002	92.32%
R-M 2	4689	0.0001	97.01%

* In the table, "R-M 1"denotes the Region-mapping model with hyper cube supervisor region, and "R-M 2" means the Region-mapping model with quarter hyper globe supervisor region.

4 Conclusions

In this paper, a Region-mapping neural network model with quarter hyper globe supervisor region is presented. It is a more effective model in classification problems in contrast with the multilayer perceptron (MLP). It allocates an area or a region for each class in the output space so that all the patterns concentrate in their demanded region after successful training of the neural network model. Since neural network is a continuous map, it is a more reasonable criterion to make the classifying consistent with the training process by taking the region where patterns concentrate in the output space as the output. Moreover, the region-mapping model can save the number of output units in contrast with the conventional manner of coding, which can minish the scale of the network. Thereby the converging speed is accelerated. In order to make it reflect the pattern distribution more exactly, a kind of more reasonable V^c, quarter hyper globe supervisor region is put forward in this paper. The proposed model with quarter hyper globe supervisor region is put into the application of the character recognition system to verify its behavior. Experimental results have shown that the proposed model has much better behavior than the general MLP and the Region-mapping model with hyper cube supervisor region. We believe that the model will have bright application future in pattern recognition field.

Acknowledgements

The research was partly supported by National Natural Science Foundation (NSFC)/High-Tech 863-306 fund and The Central/Departmental Fund of Hong Kong Polytechnic University.

References

[1] Licheng, J. System theory of neural networks, Xi Dian University, 1995.

[2] Xuefeng Wang, Xuequan Sun. Region-mapping Model of Feedforward Neural Network for Pattern Recognition. Journal of Harbin Institute of Technology, 73-77, 32(5), 2000.

[3] Hu M K, Visual pattern recognition by moment invariants, IRE Trans. Information Theory, 1962, IT (8): 179-187.

[4] Wood J, Invariant pattern recognition: a review. Pattern Recognition, 1996, 29 (1): 1-17