ガウス関数によるニューラルネットワークを用いた 紙幣認識システムと学習途上におけるパラメータの 最適化手法*

孫 柏青[†]・竹田 史章[†]

A Paper Currency Recognition System based on Neural Networks with Gaussian Functions and an Optimizing Method for Its Parameters on Way to Learning*

Baiqing SUN[†] and Fumiaki TAKEDA[†]

In this paper, in order to improve rejection capabilities of the paper currency recognition system for unknown currency patterns on promise of ensuring recognition capabilities for known currency patterns, a feed-forward neural network (FNN) with Gaussian activation function is proposed. The proposed activation function is a ridge-like function. Moreover, a hybrid-learning algorithm for optimizing the width parameters of the Gaussian function is proposed. In the network the Gaussian activation function instead of the sigmoid function is employed in all units of hidden and output layers. The algorithm consists of two steps, one is exploring local minima by employing the gradient descent search, and the other is extricating the search from local minima, in which a random search with the downhill simplex method is employed. The results of simulation reveal the potential effectiveness of the proposed activation function and the algorithm. The system with the proposed activation function and the proposed algorithm can recognize known currency patterns and reject the unknown currency patterns effectively.

1. Introduction

In recent years, along with the accelerative development of world economics incorporation course, the frontier trade and personal intercourse of various countries are increasingly frequent. Therefore, probabilities that the paper currencies of various countries are interweaved together rises increasingly. It is a new challenge for conventional bank machines such as readers and sorters. Hence it is necessary to develop the bank machines that can not only correctly recognize and classify known paper currencies, but also effectively reject unknown paper currencies.

It has been proved that neural networks are a class of effective tools on realizing nonlinear mapping according to a set of given input-output training samples, and there are too many stories of FNN being applied successfully on the areas such as pattern recognition, nonlinear adaptive systems communica-

tion, and signal processing. Therefore, in the field of paper currency recognition, many attempts using FNN have been done[1-8]. Generally, these systems can effectively recognize the known currency patterns having been learned, even if the input data has some differences from learning samples. However, they still suffer from poor rejection capabilities for unknown patterns.

In order to improve rejection capabilities of currency recognition systems, many methods such as the linear-template matching method, the multi-dimensional Gaussian probabilistic density function (PDF) method[7], and the neuro-template matching method [8] were applied. However, for the first one, although the rejection capabilities for unknown currency patterns are increased, the recognition capabilities for known currency patterns are abruptly decreased. For the second, it is difficult to estimate the data distribution in most cases. For the last, it is effective on rejecting unknown patterns, but the structure of the system is too complicated. Moreover, the improvement of the rejection capability of these methods do not stem from the neural network itself directly. In other papers, Aoba et al. proposed a combination of a three-layer perceptron and RBF networks[9], in which the perceptron were used to classification and many

* 原稿受付 2005年8月1日

[†] 高知工科大学 知能機械システム工学科 Department of Intelligent Mechanical Systems Engineering, Kochi University of Technology; 185 Miyanokuchi, Tosayamada-cho, Kami-gun, Kochi 782-8502, JAPAN
Key Words: paper currency recognition, optimizing algorithm, neural networks, Gaussian function.

RBF networks were used for validation of each feature area on paper currencies. The system construction is therefore complicated. Eccles proposed using probabilistic neural networks (PNN)[22] to reject unknown currencies. However, The dimensionality of PNN also is too large. It is difficult for them to be applied practically. It is essential to construct more effective and simpler network structures, which can possess these two capabilities simultaneously.

In this paper, a three-layer FNN with Gaussian activation functions is proposed as being a classifier in the currency recognition system. Furthermore, a new hybrid-learning algorithm is proposed to optimize the widths of the Gaussian function. Because of characteristics of the proposed activation function and the restricted activation by the width parameter sigma, the system reveals considerable potentials on recognition and rejection capabilities. In the system, the width parameter of Gaussian function distinctly influences its recognition and rejection capabilities[21]. Once the initial values of the widths are improper, the network will disaccord with converge criterion or performance targets of the system. It results from a complex error surface of the FNN being cast by many plateaus, basins and valleys, and the iteration search is easily to get stuck in local minima as the gradient descent method is used. The proposed algorithm for optimizing widths involves two steps, one is searching a local minimum by employing the conventional gradient descent method with a momentum term, and the other is extricating the iterative search from the local minimum, in which a random search with the downhill simplex method[18] is employed as the cost function get trapped in a bad local minimum. First, a random vector replaces the momentum terms in the increment terms of widths, then coefficients of the random vector and the gradient term are optimized by using the downhill simplex method. In this case, it is unnecessary to calculate derivative of widths, and the span and directions of the iteration have more possibilities than just using a random vector. In other words, efficiency of the exploring is greater than that of general random search methods.

The results of the experiment show that the proposed currency recognition system is effective on improving rejection capabilities for unknown currencies on promise of ensuring recognition capabilities for known currency patterns. Moreover, the proposed algorithm is insensitive to the change of initial range of width parameters, and it can extricate the search from local minima more quickly and easily and explore optimal solutions of widths in larger range.

This paper is organized as follows. In section 2, we address on the preprocessing part of the recognition system. Section 3 describes Gaussian activation function and structures of the network. Optimization of the weights and biases of the network is introduced in section 4. Section 5 is addressed on the detail of how to use the proposed algorithm to optimize widths

parameters of the Gaussian function. Section 6 and 7 are the computer simulations and conclusions respectively.

2. Pre-processing of Data Compression and Feature Extraction

Preprocessing is indispensable for correctly recognizing paper currency patterns, one of which is dimensionality compression of the image data. It is beneficial for reduction of network inputs. A network with fewer inputs has fewer adaptive parameters to be determined, leading to the network with better generalization properties[10]. The other is the procedure of extracting features.

In this system, the technique called pixel averaging is applied to reduce dimensionality of paper currencies to be recognized. It involves grouping blocks of pixels together and replacing each of them with a single effective pixel whose gray-scale value is given by the average of the gray-scale values of the original pixels in the block. After that, to further compress dimensionality of the network and improve robustness of the system, slab values, which is given by the sum of the gray-scale values of the transformed pixels[11], are applied to represent digital features of the paper currency image. In this procedure, for acquiring more features of a currency image, a mask set is utilized. It involves several mask patterns, each of which differs from the others and covers a different area of the currency image. There is a slab value corresponding to each of mask patterns.

Fig. 1 Processing of masks and slab values

The procedure of masks and slab values processing is illustrated in Fig.1. First, depending on the categories of paper currencies to be recognized, a specified area called as feature area is selected on the currency images. Then the feature area is divided into many blocks. After that, we can define a mask pattern that is composed of several the blocks mentioned above. Finally, the slab value of each mask pattern is calculated, which is the sum of the gray-scale values of all

but the blocks covered by the mask pattern locating in the feature area. Because many mask patterns can be obtained depending on different combinations of the blocks, many corresponding slab values are generated symptomizing the paper currencies to be recognized. In our recognition system, each of slab value of the currency image is one of inputs of the network.

3. Activation Function and Structures of the Network

3.1 Gaussian Activation Function

In order to improve rejection capabilities for unknown patterns on the premise of ensuring effective recognition capabilities for known patterns of this currency recognition system, a three-layer FNN differing from those conventional FNN is used as a classifier. The activation function in the hidden and output layers of the network is a Gaussian function, not the conventional sigmoid function. Its formula is shown as following

$$f(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right), \quad (1)$$

where σ is the width parameter of the Gaussian function. Hence all the weighted inputs fed to each unit in the hidden and output layers are summed up, and the output is governed by

$$y_i = f(\mathbf{X}_i, \xi_i) = \exp\left[-\frac{(\mathbf{W}_i \mathbf{X}_i - \theta_i)^2}{2\sigma_i^2}\right] \quad (2)$$

where y_i is the output of the i th unit for input vector $\mathbf{X}_i$, $\xi_i \in \{\mathbf{W}_i, \theta_i, \sigma_i\}$ is a set of parameters of the i th unit to be optimized, in which $\mathbf{W}_i$ and θ_i are the weight vector and the bias respectively. The output of a single Gaussian unit governed by Eq.(2) with two input variables is illustrated in Fig.2(c), in which (a) and (b) illustrate that of RBF and Gaussian bar function. The equations of them are

$$y_i = \exp\left[-\frac{\|\mathbf{X}_i - \theta_i\|^2}{2\sigma_i^2}\right]$$

and

$$y_i = \sum_j w_{ij} \exp\left[-\frac{(x_{ij} - \theta_{ij})^2}{2\sigma_{ij}^2}\right]$$

respectively. Comparing the outputs of three functions in this figure, the shape of the proposed function is different with that of RBF. It is a ridge-like function. Its activation is stretch out to infinity along the ridge, but is restricted by the width parameter on the orthogonal direction of the ridge. Its active range lies on the value of the width parameter σ . Several crossed these ridges can construct not only many bump-like peaks just like the output of RBF at intersections of these ridges, but also many unclosed ridges with different directions. It can be realized by the

combination of hidden-layer units. For instance, two ridges being as in Fig.2(c) with orthogonal directions can form the shape just like that of Gaussian bar function shown in Fig.2(b). However, $3N$ parameters per unit are needed to train for a Gaussian bar unit with N input variables. But in the case of the proposed function, there are only $N+2$ parameters. Generally, a neural network with fewer adaptive parameters to be determined requires fewer training samples, leading to reducing of the learning procedure. Moreover, it will lead to the network with better generalization properties. Furthermore, in practical applications a large memory space is indispensable for the network with large number of parameters, it will increase the cost of the practical system. The directions, distributions and active ranges are controlled by corresponding weights, biases and widths, respectively. The network is active on the data distributing not only in the peaks but also in the ridges. The system can more accurately recognize the data outside of those closed peak spaces. Hence the network with this activation function has more potential to improve rejection capabilities on promise of ensuring recognition capabilities of the system.

Fig. 2 Outputs of different activation function with two input variables

3.2 Structures of the Network

In this system, a three-layered FNN is used, its structure is decided as following rules. The number of units at input layer is equal to that of mask patterns in a mask set being used, in other words, it is the same with the quantity of slab values generated by the mask set; The number of units at output layer equals to that of currency patterns to be recognized; the number of units in hidden layer is between that of the units in input layer and output layer.

4. Optimizing Weights and Biases of the Network

The aim of the network learning is that for the mapping of a network between inputs and outputs: $y = f(\mathbf{X}, \boldsymbol{\xi})$, $\boldsymbol{\xi} \in A$, if given a set of training patterns $(\mathbf{X}_m, \hat{y}_m)$, $m = 1, \dots, M$, exploring a set of optimal $\boldsymbol{\xi}^*$ to minimize the cost function (error function or objective function) of the network given by

$$E(\boldsymbol{\xi}) = \frac{1}{M} \sum_{m=1}^M (\hat{y}_m - f(\mathbf{X}_m, \boldsymbol{\xi}))^2 \quad (3)$$

where $E(\cdot)$ is the cost function of the system.

For the weight vectors $\mathbf{W}$ and bias θ , a modified gradient descent algorithm is employed. Then the $\boldsymbol{\xi} \in \{\mathbf{W}, \theta\}$ updates at the k th iteration is shown in the following equation

$$\Delta \boldsymbol{\xi}(k) = -\alpha \nabla E(\boldsymbol{\xi}(k)) + \beta \Delta \boldsymbol{\xi}(k-1) + \gamma \Delta \boldsymbol{\xi}(k-2) \quad (4)$$

where α , β , and γ are learning rate, momentum factor and oscillation factor, respectively.

5. Widths Optimization with the Hybrid-learning Method

The width parameter of Gaussian function controls its active range of the hyper-surface, it is considerably important for generalization capabilities of the network. Depending on our previous research[21], improper widths will lead to bad recognition and rejection capabilities, and more seriously, the iteration cannot converge and the network cannot be used at all. Therefore, we proposed a hybrid-learning algorithm to explore the optimal width parameters of Gaussian function. It includes two steps, one is exploring local minima by employing the conventional gradient descent method adding a momentum term, and the other is extricating the iteration search from the local minima by employing a random search with the downhill simplex method. First, once the cost function is distinguished to getting trapped in a bad local minimum, a Gaussian random vector with respective coefficients is mixed in the increment terms of width parameters. In this case, the cost function is possibly greater or less than that of the local minimum and disaccording with a certain criterion, Then instead of reselecting random vector, the coefficients of the random vector and the gradient term are simultaneously optimized using the downhill simplex method to find a point, in which the cost function is not only less than that of the local minimum but also according with the certain criterion. Because the downhill simplex method is efficient on optimizing parameters that are not very much, and it is unnecessary to calculate derivative of any parameters, it is more possible to extricate from local minima and find better points.

Before addressing on the details of this hybrid-learning algorithm, the overview of the downhill simplex

method is provided in the next subsection.

5.1 Downhill Simplex Method

The downhill simplex method is an efficient iterative algorithm to solve unconstrained minimization problems for several variables[18]. Where a simplex is defined by $N + 1$ linearly independent corners in the N -dimensional search space (a triangle in the 2-dimensional space). Each simplex defines a solution in the search space.

The simplex method does not need to calculate derivatives. It intends to enclose the minimum inside an irregular volume defined by a simplex. The operations of changing the simplex optimally with respect to the objective function values found at the corners of the simplex are contraction, expansion and reflection, each determining new simplex corners by linear combinations of selected existing corners[16]. Its iterative steps are indicated briefly as follows.

First, it calculates the corners whose objective function value is highest (highest corner) and lowest (lowest corner) and calculates the centroid of all but the high corner. Then the high corner is reflected through the centroid calculated above. If the function value of the current point is smaller than that of the existing lowest one, it continues to attempt an expansion in that direction. Otherwise, it tries contractions of all but the lowest corner toward the direction of the lowest corner. In the simplex method, it is certain that there must be a corner whose function value is equal to that of the smallest one or less than that of all corners in last simplex in each step of iteration.

Fig. 3 Simplex method in 2-dimensional space

Fig.3 illustrates the downhill simplex method in a 2-dimensional space (plane of x_1 - x_2), in which P_h , P_m and P_l are corners of a current triangle simplex according with $f(P_h) > f(P_m) > f(P_l)$. P is the centroid of P_m and P_l , P_r and P_e are the reflection and expansion points respectively of P_h along the direction through centroid P . P_{ch} and P_{cm} represent respectively the contraction points of P_h and P_m toward P_l .

5.2 Hybrid Learning Algorithm

The illustrations of the flow of the proposed learning algorithm are shown in Fig.4.

Fig. 4 Flowchart of the learning algorithm

On the first step of our proposed algorithm, it uses the gradient descent method for searching local minima near the starting point. On base of Eq.(3), the update formula of the width vector σ is given by

$$\sigma(k) = \sigma(k-1) + \Delta\sigma(k) \quad (5)$$

$$\Delta\sigma(k) = -\alpha \cdot \nabla E(\sigma(k)) + \beta \cdot \Delta\sigma(k-1) \quad (6)$$

where α and β are the learning rate and momentum coefficient of widths σ respectively. $\nabla E(\sigma(k))$ is the gradient of the cost function with respect to σ at the k th iteration.

After that, as the cost function gets stuck in a local minimum and its value still disaccords with the convergence criterion, the second stage of our algorithm, a random search with downhill simplex method is executed to extricate the search from the local minimum. After finding a better point on the error surface with the second stage, then the gradient search is employed again. The two stages are executed alternately until the convergence criterion is accorded. Afterwards, the criterions of getting stuck into and having extricated from local minima mentioned in the flowchart are described in the next subsection.

5.2.1 Criterions of Being Stuck into and Extricated from a Local Minimum

In searching a local minimum along the gradient direction, the criterions of getting trapped into local minima is given by

$$\frac{|E_c(\sigma(k)) - E_c(\sigma(k-1))|}{E_c(\sigma(k))} < \varepsilon_T \quad (7)$$

where $0 < \varepsilon_T \ll 1$. It means that as the ratio of the increment of the cost function caused by the variable

σ and itself is less than a specified value ε_T , it is considered getting trapped into a local minimum. $E_c(\cdot)$ denotes the error function of the training samples in a certain class, which leads to the cost function $E(\cdot)$ is stuck into a bad minimum. The gradient searching is paused and the cost function value of that time is recorded and marked as $E_{c0}(\cdot)$. Then the second stage of the algorithm is started to execute.

In the second stage, if the difference of E_{c0} mentioned above and the current cost function value of the $E_c(k')$ accords with the formula given by

$$E_{c0} - E_c(k') > \lambda E_{c0}, \quad 0 < \lambda < 1. \quad (8)$$

It is considered having extricated from local minima. It reveals that by using the second stage we can find a point whose cost function value is less than that of the local minimum trapped in the last stage. In other words, we can find a better point having lower energies on the error surface.

5.2.2 Random Search with Simplex

Once the gradient iteration search is ascertained being stuck into a local minimum, the increment of σ shown in Eq.(6) is transformed by

$$\Delta\sigma(k) = -\alpha \cdot \nabla E(\sigma(k)) + \beta \cdot \mathbf{G}(k) \quad (9)$$

where $\mathbf{G}(k)$ is a Gaussian random vector, which has the same dimensionalities with σ . During searching the appropriate σ in this stage, $\nabla E(\sigma(k))$ is constant and frozen at the ultimate value of the last stage having stuck in a local minimum in gradient search. For a decided $\mathbf{G}(k)$, the cost function $E(\cdot)$ of the system is just the function of α and β marked as $E(\alpha, \beta)$. In this case, the downhill simplex method is used to explore a set of optimal α^* and β^* , which accord $E(\alpha^*, \beta^*)$ with the criterion shown in Eq.(8).

Since these two parameters are adjustable, the span and direction of the increment have much more combinations and possibilities. The simplex method in each iteration can find a new simplex in which there must be a corner whose function value is less than or equal to that of the smallest one in last simplex. Hence it is more easily to extricate the search from local minima and arrive to a better point on the error surface. The detail of this stage is described as following:

Step 1 Generate a Gaussian random vector whose dimension is same with that of parameter σ , and set a maximum iteration steps K .

Step 2 Generate an initial triangle simplex on the plane of α - β by selecting randomly 3 pairs of α and β , in which the corner is $\mathbf{P}_i(\alpha_i, \beta_i)$, $i = 1, 2, 3$.

Set the reflection coefficient $C_r > 0$ the expansion coefficient $C_e > 1$, and the contraction coefficient $C_c \in (0, 1)$.

Calculate the error function value $E(\mathbf{P}_i(\alpha_i, \beta_i))$ of each corner ($i = 1, 2, 3$). Let $k = 1$.

Step 3 Decide the highest corner $\mathbf{P}_h$, the second highest corner $\mathbf{P}_m$ and the lowest corner $\mathbf{P}_l$, where h, m

and $l \in \{1, 2, 3\}$, according with

$$\begin{aligned} E(\mathbf{P}_h) &= \max\{E(\mathbf{P}_1), E(\mathbf{P}_2), E(\mathbf{P}_3)\} \\ E(\mathbf{P}_m) &= \max\{E(\mathbf{P}_i) | i \neq h\} \\ E(\mathbf{P}_l) &= \min\{E(\mathbf{P}_1), E(\mathbf{P}_2), E(\mathbf{P}_3)\} \end{aligned}$$

Then calculate the centroid of $\mathbf{P}_m$ and $\mathbf{P}_l$, let

$$\bar{\mathbf{P}} = \frac{1}{2}(\mathbf{P}_m + \mathbf{P}_l)$$

Step 4 Execute the reflection operation, let $\mathbf{P}_4 = \bar{\mathbf{P}} + C_r(\bar{\mathbf{P}} - \mathbf{P}_h)$, and calculate $E(\mathbf{P}_4)$.

Step 5 If $E(\mathbf{P}_4) < E(\mathbf{P}_l)$, then execute expansion operation. Let $\mathbf{P}_5 = \bar{\mathbf{P}} + C_e(\mathbf{P}_4 - \bar{\mathbf{P}})$, and calculate $E(\mathbf{P}_5)$, then jump to Step 6;

If $E(\mathbf{P}_l) \leq E(\mathbf{P}_4) \leq E(\mathbf{P}_m)$, then let $\mathbf{P}_h = \mathbf{P}_4$, and jump to Step 8;

If $E(\mathbf{P}_4) > E(\mathbf{P}_m)$, then execute contraction operation. Set $E(\mathbf{P}_{h'}) = \min\{E(\mathbf{P}_h), E(\mathbf{P}_4)\}$, Where $h' \in \{h, 4\}$. Let $\mathbf{P}_6 = \bar{\mathbf{P}} + C_c(\mathbf{P}_{h'} - \bar{\mathbf{P}})$, and calculate $E(\mathbf{P}_6)$, then jump to Step 7.

Step 6 If $E(\mathbf{P}_5) < E(\mathbf{P}_4)$, let $\mathbf{P}_h = \mathbf{P}_5$, and jump to Step 8;

Otherwise let $\mathbf{P}_h = \mathbf{P}_4$, and jump to Step 8.

Step 7 If $E(\mathbf{P}_6) \leq E(\mathbf{P}_{h'})$, then set $\mathbf{P}_h = \mathbf{P}_6$ and jump to Step 8;

Otherwise execute contraction operation, let $\mathbf{P}_i = \mathbf{P}_i + C_c(\mathbf{P}_l - \mathbf{P}_i)$, where $i \in \{h, m\}$, and calculate $E(\mathbf{P}_i)$.

Step 8 Being as the current cost function value, let $E(\mathbf{P}^*) = \min\{E(\mathbf{P}_i) | i \in \{h, m, l\}\}$.

If $E(\mathbf{P}^*)$ accords with the criterion in Eq.(8), jump out from this algorithm and end it; Otherwise let $k = k + 1$.

Step 9 If $k = K$, then let $k = 0$ and return to Step 1; Otherwise let $\mathbf{P}_l = \mathbf{P}_1$, $\mathbf{P}_m = \mathbf{P}_2$ and $\mathbf{P}_h = \mathbf{P}_3$, and then return to Step 3.

6. Simulation

In experiments of this paper currency recognition system, 3 kinds of paper currency, US dollars (12 patterns), Thai banknotes (10 patterns) and Renminbi (Chinese paper currency, 10 patterns) are recognized. Among them, Renminbi is used as known patterns for the system, which means the network is trained with Renminbi, then the US dollars and Thai banknotes are as the unknown patterns for this recognition system. The constitution of the network follows the rules mentioned in section 3.2. Since a mask set with 50 mask patterns is used to extract 50 slab values for each piece of paper currency, there are 50 units in input layer. Then there are 10 units in output layer because of having 10 classes of training samples. The structure of the network is $50 \times 30 \times 10$.

The criterions ε_T and λ in Eq.(7) and (8) are 10^{-4} and 0.03, respectively. The other conditions of learning are shown in Table 1. The initial values of the widths are generated in the ranges of (0.5, 0.6), (0.45, 0.7), (0.2, 0.9) and (0.2, 1.2) respectively.

First, the network was trained with the proposed

Table 1 Conditions of learning

Initial Learning Rate α	0.05
Momentum Coefficients β	0.80
Oscillation Coefficient γ	-0.0001
Reflection Coefficient C_r	1.0
Expansion Coefficient C_e	1.2
Contraction Coefficient C_c	0.7
Convergence Criterion	Average of square error < 0.0001
Maximum Iteration No.	20000

hybrid-learning algorithm using Renminbi patterns. When the initial widths were in the first two ranges, the second stage of the algorithm was not activated. In other words, in this case the first stage of the algorithm, the gradient search, activated independently to converge the network. In the cases of last two ranges, the cost function cannot accord with the convergence criterion using the first stage of the algorithm by itself. Then the second stage, the random search with the simplex, was activated and extricated the search from local minima. The two stages were active alternately to converge the cost function. In the experiments, we also discovered that the learning rate is a key factor that it should be set carefully when the search returns to the gradient search from the second stage of the algorithm again. If it is too large, the search cannot converge and will oscillate.

Fig.5 shows the error response curves of the training procedure, in which (1) shows the total error of all the training samples, and (2), (3), (4) and (5) show the respective error curves of the Renminbi Patterns RMB100H, RMB50T, RMB20T and RMB10T. As can be seen from Fig.5(1), in whole convergence procedure, the second stage of the algorithm is activated four times, whose positions are also illustrated in the respective subfigures. Comparing all the subfigures, it can be found because the error of the pattern RMB100H got trapped into a local minimum, the second stage of the algorithm was activated firstly at *activated point 1* shown in Fig.5(2), where the error was 0.500069. Depending on the second stage, the error was decreased to 0.481053 and was extricated from the local minimum successfully. From this new point, the first stage was restarted again and its error converged. However, this new point led to the increase of the error of pattern RMB10T and got trapped into a new local minimum. At *activated point 2*, where the error of RMB10T is 0.500004, the second stage of the algorithm was activated again. When the error escaped from this local minimum, its value is 0.362579. The errors of pattern RMB100H, RMB50T and RMB20T therefore were increased simultaneously. This transition also resulted in that the total error function was increased to about 2.5 as shown in Fig.5(1). From this new point, using the first stage of the algorithm, the error of RMB100H converged quickly, but unfortunately the errors of RMB50T

(1) Total error

(2) Error of RMB100H

(3) Error of RMB50T

(4) Error of RMB20T

(5) Error of RMB10T

Fig. 5 Error curves of the training procedure

and RMB20T were stuck in local minima. Hence the second stage was activated in succession at *activated point 3* and *activated point 4*. Their errors hence were decreased from 0.500259 and 0.499998 to 0.299979 and 0.457219 respectively. Then the first stage was restarted again and their errors converged to the point according with the criterion. The new point found by the fourth activation led to the increase of the error of RMB10T, but it converged by inactivating the second stage of the algorithm.

After training the network, to verify the recognition capability of the system for known patterns, 10 patterns of Renminbi being as the known patterns, each of which involves 100 pieces differing from the training samples (including some worn, smutched and incomplete currencies) were inputted the system. The recognition results are shown in Table 2.

Table 2 Recognition ratios for the known patterns

Patterns	Initial range of width parameters			
	(0.5, 0.6)	(0.4, 0.7)	(0.2, 0.9)	(0.2, 1.2)
RMB100H	98%	97%	96%	98%
RMB100T	100%	100%	100%	100%
RMB50H	95%	100%	100%	100%
RMB50T	100%	99%	100%	100%
RMB20H	99%	100%	99%	99%
RMB20T	99%	96%	99%	99%
RMB10H	100%	100%	100%	100%
RMB10T	97%	99%	97%	94%
RMB5H	100%	100%	100%	99%
RMB5T	99%	99%	100%	100%
Average	98.7%	99%	99.1%	98.9%

As can be seen from Table 2, for each of initial range of the width parameters, although the recognition ratio for each pattern has a little difference among them, the average recognition ratios for all patterns of Renminbi are almost identical. They are greater than 98%. It reveals that given a appropriate initial range, optimized widths of Gaussian function can be found just using the first stage of the algorithm to earn good recognition capabilities. Moreover, even if the initial range is not ideal, the system also can obtain the satisfied recognition effects by alternately utilizing the two stage of the algorithm.

Therefore, to verify the rejection capabilities of the system for unknown patterns, 12 patterns of US dollar and 10 patterns of Thai banknotes were also recognized by the system, there are 200 and 100 pieces for each pattern of US dollars and Thai banknotes, respectively. Table 3 and Table 4 show the corresponding rejection ratios of them. In this case, all patterns of US dollar and Thai banknote are unknown patterns for the network. They all should be rejected ideally.

Form Table 3 and Table 4, you can see that as the initial range varies from (0.5, 0.6) to (0.2, 1.2), the average rejection ratio for Thai banknotes increases 1.8 percentages. It decreases about 2 percentages for US dollars. However, once the initial range is (0.2, 0.9), the rejection ratio for the pattern US1T is only

Table 3 Rejection ratios for unknown Thai banknotes

Thai banknotes	Initial range of width parameters			
	(0.5, 0.6)	(0.45, 0.7)	(0.2, 0.9)	(0.2, 1.2)
BG1000H	97%	100%	100%	100%
BG1000T	100%	100%	100%	100%
BG500H	100%	88%	98%	96%
BG500T	82%	97%	99%	100%
BG100H	99%	100%	100%	100%
BG100T	100%	100%	82%	100%
BG50H	100%	100%	100%	100%
BG50T	99%	100%	94%	99%
BG20H	100%	100%	99%	100%
BG20T	100%	100%	100%	100%
Average	97.7%	98.5%	97.2%	99.5%

Table 4 Rejection ratios for unknown US dollars

US dollars	Initial range of width parameters			
	(0.5, 0.6)	(0.45, 0.7)	(0.2, 0.9)	(0.2, 1.2)
US100H	100%	100%	99%	99.5%
US100T	98.5%	98%	98.5%	89.5%
US50H	100%	100%	99%	98%
US50T	100%	100%	100%	98%
US20H	100%	100%	100%	99.5%
US20T	99%	95%	93.5%	90%
US10H	100%	100%	100%	100%
US10T	98.5%	98.5%	100%	100%
US5H	99.5%	100%	99%	99.5%
US5T	100%	100%	100%	100%
US1H	99.5%	100%	94.5%	98%
US1T	99.5%	97%	81%	88.5%
Average	99.54%	99.04%	97.04%	97.13%

81% that means 38 pieces of US1T are misrecognized as some patterns of Renminbi. It reveals that for getting good performances of the system, any of the appropriate network structure, the proper initial range and effective algorithm is indispensable.

It has mentioned that when the initial range of the widths is (0.2, 0.9) and momentum coefficient is 0.8 shown in Table 1, the network cannot converge by only using the gradient search. Nevertheless, if the momentum coefficient in the increments of widths is 0.084, the network also can converge according with the criterion only using the gradient search. In this case, the rejection ratios of the system for US dollars and Thai banknotes are shown in Table 5. It shows although the cost function accords with the convergence criterion, the performance of the system is unsatisfactory relying on converge to a bad local minimum.

Comparing Table 5 with Table 3 and 4, it reveals that it is achievable to find a good solution using the proposed algorithm in exploring suitable widths of the Gaussian function. To further verify rejection capabilities of the proposed system, the contrastive experiments in which the sigmoid activation function is used in the FNN with the same restrictions therefore were executed. The results are shown in Table 6.

It shows that the rejection capabilities of the proposed system are all increased about 10 percentages

Table 5 Rejection ratios for a specified momentum

US dollars	Initial range of width parameters		Thai Bbanknotes
	(0.2, 0.9)		
US100H	93%	100%	BG1000H
US100T	95.5%	66%	BG1000T
US50H	71.5%	71%	BG500H
US50T	100%	49%	BG500T
US20H	82%	100%	BG100H
US20T	98.5%	97%	BG100T
US10H	61.5%	100%	BG50H
US10T	100%	100%	BG50T
US5H	80%	93%	BG20H
US5T	100%	100%	BG20T
US1H	94.5%		
US1T	87.5%		
Average	88.67%	87.6%	Average

Table 6 Contrastive Rejection Percentages

Activation Function	Average Rejection Percentages	
	US Dollar	Thai Banknote
Sigmoid Function	89.58%	87.1%
Gaussian Function	98.19%	98.2%

for US dollars and Thai banknotes than that of using the sigmoid activation function. It proves the structure of the system is more effective than that of the network with the sigmoid function on rejection capabilities. Furthermore, the rejection capabilities of the proposed system are compared with that of a combined recognition system, which consists of a three-layer perceptron with sigmoidal functions and several linear templates whose number is equivalent to the classes of training samples. For each class of learning samples, the mean and variance of slab values of each mask pattern are calculated, then the corresponding means and variances are combined linearly to produce a specified template, which includes the upper and lower limits of the corresponding class of input data. For each input data, it is necessary not only to calculate outputs of the perceptron, but also to compare each input data of the network with these templates. In other words, it has to synthetically consider the results of these two components in the recognition system once a paper currency is recognized. The results are shown in Table 7. In this case, besides of US dollar and Thai banknote, banknotes of Canada and Australia are also used as unknown currencies for two systems. There are 28 and 24 patterns of them respectively, but there is only one piece in each pattern of them. In this case the initial range of widths of the proposed function is (0.5, 0.6). Comparing this table with Table 6 on US dollar and Thai banknote, it can be found that rejection capabilities for them are improved as the linear templates are combined with the sigmoid neural network, but it is still lower than that of the proposed method. It reveals from Table 7 that

Table 7 Comparison of rejection capabilities with different methods

Nation	Rejection ratio	
	NN + linear template matching	Proposed method
US	96.67%	99.54%
Thai	93.1%	97.7%
Can	87.5%	100%
Aus	89.29%	95.83%

rejection capabilities of the proposed system are better than that of the combination method. Moreover, the proposed recognition system is just composed of the neural network, thus its structure is simpler than that of the combined recognition system.

7. Conclusion

In this paper, we proposed a currency recognition system based on the neural network, in which a Gaussian activation function whose active range is controlled by the width parameter was employed in all units in hidden and output layers. Moreover, a hybrid-learning algorithm is proposed to explore the optimal width parameters of the Gaussian function. The activation function is a ridge-like function, and the hybrid algorithm consists of the gradient descent search and a random search with the downhill simplex method.

The results of the simulation reveal that the currency recognition system with the proposed the activation function and the algorithm is effective on rejecting unknown currency patterns on premise of ensuring recognition capabilities for known currency patterns having been learned. Moreover, it shows that the width parameter of the proposed activation function is important to influence the performance of the recognition system. The rejection capabilities were improved if the widths are optimized effectively. It also reveals that with the proposed algorithm the span and directions of iteration search of widths have more combinations. It can extricate the search from local minima and explore optimal width parameters in larger ranges to improve the recognition and rejection capabilities of the system.

References

- [1] F. Takeda and S. Omatu: High speed paper currency recognition by neural networks; *IEEE Trans. on Neural Networks*, Vol. 6, No. 1, pp. 73–77 (1995)
- [2] F. Takeda, S. Omatu and S. Onami: Recognition system of US dollars using a neural network with random masks; *Proceedings of the International Joint Conference on Neural Networks*, Vol. 2, pp. 2033–2036 (1993)
- [3] F. Takeda and S. Omatu: A neuro-paper currency recognition method using optimized masks by genetic algorithm; *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 5, pp. 4367–4371 (1995)
- [4] F. Takeda and S. Omatu and Y. Matsumoto: Development of high speed neuro-recognition unit and application for paper currency; *The International Workshop on Signal Processing Application and Technology*, pp. 49–56 (1998)
- [5] F. Takeda, T. Nishikage and Y. Matsumoto: Characteristic extraction of paper currency using symmetrical masks optimized by GA and neuro-recognition of multi-national paper currency; *Proceedings of IEEE World Congress on Computation Intelligence*, Vol. 1, pp. 634–639 (1998)
- [6] F. Takeda, T. Nishikage and S. Omatu: Banknote recognition by means of optimized masks, neural network, and genetic algorithm; *Engineering Applications of Artificial Intelligence* 12, pp. 175–184 (1999)
- [7] F. Takeda and M. Tanaka: A rejecting method of non-objective currencies using multi-dimensional Gaussian PDF for neuro-multi national currency recognition; *Trans. IEE of Japan*, Vol. 118-D, No. 12, pp. 1361–1369 (1998)
- [8] F. Takeda and T. Nishikage: Development of a neuro-templates matching recognition method for banknotes; *Trans. IEE of Japan*, Vol. 121-C, No. 1, pp. 196–205 (2001)
- [9] M. Aoba, T. Kikuchi and Y. Takefuji: Euro banknote recognition system using a three-layered perceptron and RBF network; *IPSJ Journal*, Vol. 41, No. 6, pp. 1234–1244 (2000)
- [10] C. M. Bishop: *Neural Network for Pattern Recognition*, Oxford University Press (1995)
- [11] B. Widrow, R. G. Winter and R. A. Baxter: Layered neural nets for pattern recognition; *IEEE Transaction Acoustic, Speech & Signal Preprocessing*, Vol. 36, No. 7, pp. 1109–1118 (1988)
- [12] B. Norial, et al.: A hybrid algorithm for finding the global minimum for error function of neural networks and its applications; *Neural Networks*, Vol. 7, No. 8, pp. 1253–1265 (1994)
- [13] A. Frosini, M. Gori and P. Priami: A neural network-based model of paper currency recognition and verification; *IEEE Transactions on Neural Networks*, Vol. 7, No. 6, Nov. pp. 1482–1490 (1996)
- [14] X. Wang, Z. Tang, et al.: Two-phase pattern search-based learning method for multi-layer neural network; *IEEJ Trans. EIS*, Vol. 124, No. 3, pp. 842–851 (2004)
- [15] Z. Tang and J. K. Gary: Deterministic global optimal FNN training algorithm; *Neural Networks*, Vol. 7, No. 2, pp. 301–311 (1994)
- [16] <http://rkb.home.cern.ch/rkb/AN16pp/node262.html>
- [17] S. Ma and C. Ji: Performance and efficiency: Recent advances in supervised learning; *Proceedings of the IEEE*, Vol. 87, No. 9, pp. 1519–1535 (1999)
- [18] J. A. Nelder and R. Mead: A simplex method for function minimization; *Computer Journal*, Vol. 7, pp. 308–313 (1965)
- [19] P. Patrick: Minimization methods for training feed forward neural networks; *Neural Networks*, Vol. 7, No. 1, pp. 1–11 (1994)
- [20] T. L. Fine: *Feedforward Neural Network Methodology*, Springer Press (1999)
- [21] B. Sun and F. Takeda: Proposal of neural recogni-

tion with Gaussian function and discussion for rejection capabilities to unknown currencies; *Proceeding of International Conference on Knowledge-based Intelligent Information & Engineering Systems*, pp. 859–865 (2004)

- [22] N. J. Eccles: Neural network for banknote recognition and authentication; *United States Patent*, EP0660276 (1995)