Research on Rejection Capabilities of Paper Currency Recognition System with the Neural Network Employing Gaussian Function

Baiqing Sun

A dissertation submitted to
Kochi University of Technology
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Department of Engineering
Graduate School of Engineering
Kochi University of Technology
Kochi, Japan

March 2006

Abstract

In this research, in order to improve rejection capabilities of currency recognition systems for unknown currency patterns on premise of guaranteeing their recognition capabilities for known currency patterns, a new paper currency recognition system based on neural networks is proposed. The neural network is a three-layer feedforward neural network (FNN), in which a Gaussian function is employed as the activation function in hidden layer and output layer. The recognition system is composed of two parts. The first is preprocessing, including detecting edges, compressing data dimensionalities, and extracting digital features. The second one is recognition, in which the core is a neural network classifier, in which the ridge-like Gaussian activation function is used. It makes the classifier have the potential of rejecting unknown currency pattern. Outputs of the classier are evaluated according with a certain criterion, and hence the input currencies are judged whether be rejected or not, and which pattern belongs to.

In the procedure of preprocessing, in order to reduce dimensionality of paper currencies to be recognized effectively, pixels of currency images are grouped in some blocks, each of which are replaced 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, in order to further compress dimensionality of the network and improve robustness of the system, slab values 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. Because many mask patterns can be obtained depending on different combinations of the blocks, many corresponding slab values are generated to be representative of the digital features of these currencies. In our recognition system, the slab values of a currency image are the input vector of the neural network.

In the procedure of recognition, the slab values representing features of the currency are inputted the neural network employing the proposed Gaussian activation function, which is different with not only the simple radial basis function but also the Gaussian bars function. It is a ridge-like function in multi-dimensional space. 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. The directions, distributions and active ranges are controlled by corresponding weights, biases and widths, respectively. Hence the network with this activation function has more potential to improve rejection capabilities on promise of ensuring recognition capabilities of the system.

The experiments about the influences of the parameters of the Gaussian function to performances of the system are designed and executed. The corresponding results are analyzed. It can be found from these results that the performances of the system are very sensitive to variations of the width parameter. During the training procedure, in order to obtain satisfying performances, several learning algorithms are employed to optimizing the parameters. The improved back propagation algorithm is used to optimize the weights and biases of the network. The sequential gradient algorithm fitting for the proposed network are derived, and applied to optimize the width parameters. At the same time, it is found in this experiment that the different sequences of training currency samples influence the rejection capabilities of the system remarkably as using the sequential version of the gradient descent algorithm, and the reasons leading to this phenomenon are analyzed.

In order to search the appropriate width parameters in larger region and avoid the problem of local minima, a new hybrid-learning algorithm is also proposed. The algorithm is used to optimize the widths of the Gaussian function. The algorithm consists of two steps, one is searching local minima near start point by employing the sequential gradient descent method with a momentum term, because the sequential gradient descent algorithm is a stochastic searching method and is possible to escape the iterative search from local minima. If the iterative search still cannot shake off bindings of local minima by employing this first step solely, the second step of this algorithm is then activated. First, a random vector is mixed in increment terms of the width parameters to replace that momentum term, then coefficients of the random vector and the gradient term are optimized simultaneously using the downhill simplex method. In this case, derivative calculation is unnecessary, and the span and directions of the iteration search have more possibilities. It hence can explore optimal parameters in a larger range and extricate the search from local minima more quickly and easily.

The results of the experiments show that using the proposed algorithm, the iteration search span and directions of increments of the widths have more combinations, it can extricate the search from local minima more quickly and easily and explore optimal solutions of widths in a larger range. Moreover, the proposed recognition system using the algorithm is insensitive for the change of initial range of width parameters, and it can improve the rejection capabilities for unknown currency patterns on promise of guaranteeing the recognition capabilities for known currency patterns.

Contents

Chapter 1
INTRODUCTION

In recent years, along with the accelerative developments of world economics incorporation course, the start of Euro area, and the increase of Asia economics, frontier trade and personal intercourse of various countries are frequent increasingly. Traveling people always take many countries of paper currency. Probabilities that the paper currencies of various countries are probably interweaved together therefore rises increasingly. It is a challenge for conventional paper currency recognition systems. However, the focus of most of the conventional currency recognition systems and machines is on recognizing counterfeit currencies. It is not enough for practical businesses. The reason is that in most of banks, especially those internationalized banks, there are large quantities of cash belonging to many different countries need to be process, and it is possible that all of them are real cashes. The situation that cashes belonging to different countries mixes together is possible to occur. It cannot be processed with conventional currency recognition systems. For example there is a

currency recognition system, which can correctly recognize and classify the 20 kinds of currency with different par values or coming from different countries because this system 'knows' these currencies so much. Therefore this system is responsible to recognize these 20 kinds of currency specially. However, some pieces of currency not belonging to those 20 kinds, which is unknown for the system, are mixed together with those 20 kinds of currency to input this system accidentally. All of these unknown currencies should be rejected in ideal situation. But because there is not any relative information of these unknown data in the system, it is possible to lead to that these currencies are misrecognized as a certain kind in those 20 kinds of currency. In practice this situation is not permitted to take place. Hence it is necessary to develop the currency recognition systems and bank machines that can not only realize correct recognitions and classifications for known paper currencies, but also complete effective rejections for 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 neural networks being applied successfully on the areas such as pattern recognition, nonlinear adaptive system, communication, signal processing, and so on. Therefore, in previous researches of ours and some other researchers, many attempts have been done to classify paper currencies using neural networks. Generally, these systems can effectively recognize the known currency patterns having been learned, even if the input data has some differences from the learning samples. However, it still suffers from rejection capabilities for unknown currency patterns.

In order to improve rejection capabilities of currency recognition systems, many methods such as the post processing with linear-template matching method[4], the multi-dimensional Gaussian probabilistic density function (PDF) method[8], and the neuro-template matching method[7] 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 structures of the system are too complicated. Moreover, the improvement of the rejection capability of these methods do not stem from the neural network itself directly. In other papers, *M. Aoba* proposed a combination of a three-layer perceptron and RBF networks[9], in which the perceptron were used to classification and many RBF networks were used to validation for each feature area on paper currencies. The system construction is therefore complicated. *Eccles, N. J.* proposed using probabilistic neural networks (PNN)[37] 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, in order to improve rejection capabilities of currency recognition systems for unknown currency patterns on premise of guaranteeing their recognition capabilities for known currency patterns, a new paper currency recognition system based on neural networks are proposed. The neural network used in this system is a three-layer feedforward neural network (FNN), in which a Gaussian function is employed as the activation function of each unit in hidden layer and output layer. The proposed Gaussian activation function is different with conventional Gaussian functions such as radial basis function and Gaussian Bar function, it is a ridge-like function. Just because of the characteristics of this kind of ridge-like function, the proposed recognition system shows its potentials on improve rejection capabilities for unknown currencies.

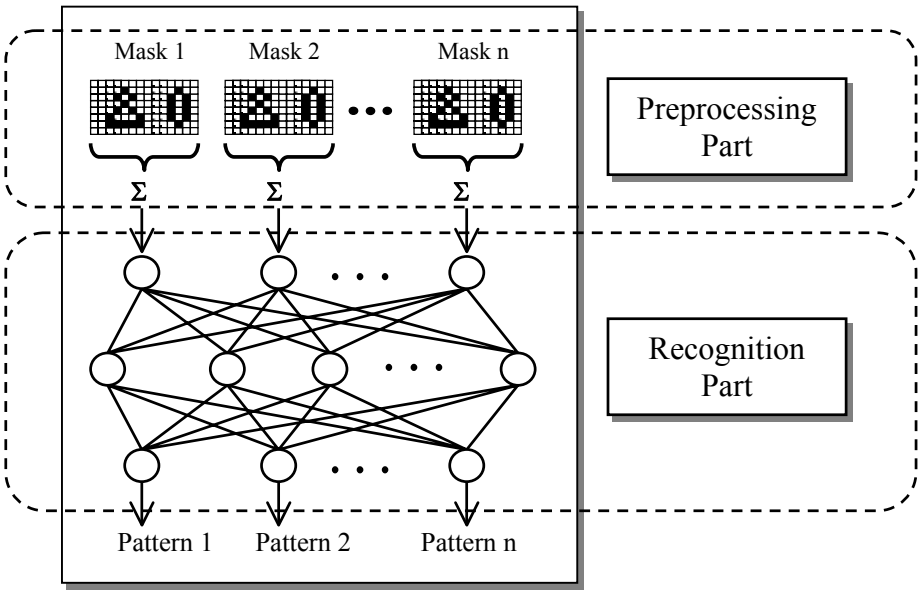


Fig.1.1. Structures of the proposed currency recognition system

The proposed paper currency recognition system is composed of two parts. The first one is the part of preprocessing, in which for collected currency image data the recognition system achieves the tasks including detecting edges, compressing data

dimensionalities, and extracting digital features. In the procedure of extracting currency features, the techniques of masks and slab values are employed. During this processing all currency images are transformed to a set of slab values using masks to be representative of the digital features of these currencies. The slab values of a currency image are the input vector of the classifier mentioned next. The second one is the part of recognition, in which the core is a neural network classifier. This neural network is the three-layer FNN just mentioned above, in which the ridge-like Gaussian activation function is used. It makes the classifier have the potential of rejecting unknown currency pattern. Outputs of the classier are evaluated according with a certain criterion, and hence the input currencies are judged whether be rejected or not, and which pattern belongs to. The structures of the proposed currency recognition system are illustrated in Fig.1.1.

During the training procedure of this neural network, In order to obtain satisfying performances of the proposed recognition system, several learning algorithms are employed to explore optimal solutions of parameters. During this period the influences of the parameters of the proposed network to performances of the recognition system are analyzed. The influences of sequences of training samples to the performance of the system are discussed, and its reasons are also analyzed. Moreover, it is found that the width parameter of the Proposed Gaussian function is very sensitive to rejection capabilities for unknown currency patterns of the recognition system.

In order to search the appropriate width parameters in larger region and avoid the problem of local minima caused by using back propagation algorithm, a new hybrid-learning algorithm is also proposed. The algorithm is used to optimize the widths of the Gaussian function for improving recognition and rejection capabilities of this currency recognition system. The algorithm consists of two steps, one is searching local minima near start point by employing the sequential gradient descent method with a momentum term, because the sequential gradient descent algorithm is a stochastic searching method and is possible to escape the iterative search from local minima. If the iterative search still cannot shake off bindings of local minima by employing this first step solely, the second step of this algorithm is then activated. It is extricating the iterative search from local minima. First, a random vector is mixed in increment terms of the width parameters to replace that momentum term, then coefficients of the random vector and the gradient term are optimized simultaneously using the downhill simplex method. In this case, derivative calculation is unnecessary, and the span and directions of the iteration search have more possibilities than just using a random vector. In other words, its search efficiency is greater than that of general random search methods. It hence can explore optimal parameters in a larger range and extricate the search from

local minima more quickly and easily.

The results of the experiments show that using the proposed algorithm, the iteration search span and directions of increments of the widths have more combinations, it can extricate the search from local minima more quickly and easily and explore optimal solutions of widths in a larger range. Moreover, the proposed recognition system using the algorithm is insensitive for the change of initial range of width parameters, and it can improve the rejection capabilities for unknown currency patterns on promise of guaranteeing the recognition capabilities for known currency patterns.

In the last part of this paper, to verify the practicability of the proposed currency recognition system on business developments and applications, some validation experiments are executed on the real-time system, and The results of these experiments also reveal the considerable potentials of the proposed recognition system on these aspects.

This paper is organized as follows. In chapter 2, we address on the preprocessing part of the recognition system. The contents include collection of currency images, initial compression of image data, detection method of edges, extraction of digital features using slab values and masks. Chapter 3 describes the recognition part of the proposed recognition system including hardware and software. In this chapter the structures of the network are introduced. The characteristics of the Proposed Gaussian activation function are indicated. It is also addressed on the details of how to use the relative learning algorithm to optimize the parameters of the proposed network. The structures and operation modes of the real-time system is introduced in chapter 4. Chapter 5 is the part of experiments including analysis for the experiment results. Chapter 6 involves the conclusions of this paper.

Chapter 2
PRE-PROCESSING

In real world, there are many kinds of object need to be recognized. It is not a difficult task for human kind. However, it is difficult for modern computer recognition systems. Input data of the most of recognition systems always contains some amount of noise, and the dimensionality of them will be very high. Therefore, in many practical applications the choice of pre-processing will be one of the most significant factors in determining the performance of the final system.[10]

Reasons for pre-processing may be easier subsequent analysis, improving classification performance through a more stable representation, removing redundant or irrelevant information. One of the most important forms of pre-processing involves a reduction in the dimensionality of the input data. In neural network recognition systems, one of the principal motivations for dimensionality reduction is that it can help to alleviate the worst effects of the curse of dimensionality.

In paper currency recognition systems, if the images of currencies are used to recognize, because of the huge initial dimensionality of the images, for correctly recognizing the patterns of them, preprocessing is indispensable certainly. One of the important forms of which involves compression in the dimensionality of the image data. It is also 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 of paper currencies.

The pre-processing of the currency recognition in this paper is composed of the aspects as following:

1. Collecting and transforming currency images. During this part, the relative currency images are collected by a scanner with appropriate mode and resolution, considering the relationship and tradeoff of accurate recognition and the scale of following classifier.

2. Detecting the edges of the transformed images. For further processing the currency images, the area of currency image should be found out correctly from the collected images with background. Therefore, edges of the currency are detected with two thresholds in this part.

3. Extracting respective features of currencies. After deciding the correct area of the currencies, features of the currencies are extracted. In this procedure, the mask processing and slab values are applied, in which the mask area and the mask patterns should be decided and the respective slab values should be calculated. The slab values of each mask pattern just represent the feature of the currency image.

Of course there should be including other processing before the feature extraction, such as rotating askew image, whose details are not introduced in this chapter. We will explain the three parts mentioned above in detail in the following.

2.1 Data Collection and Transformation

In order to recognize paper currencies effectively, the currency images are collected with appropriate spatial resolutions and brightness resolution, where the spatial resolution describes how many pixels comprise a digital image or how many dots are in each inch of a digital image. Every pixel in a digital image represents the intensity of the original image at the spatial location where it was sampled. The concept of the brightness resolution addresses how accurately the digital pixel's brightness can represent the intensity of the original image. With higher brightness resolution, a digital image appears more natural and more continous, in which more information is included in it. As the brightness resolution decreases, the image appears coarser and more mechanical. In our recognition sytem, considering scale of image data and complexity of processing, the saving mode of currency images is bitmap (BMP) mode. The spatial resolution of them is 75 dpi and the brightness resolution is 8-bit with 256-level gray-scale.

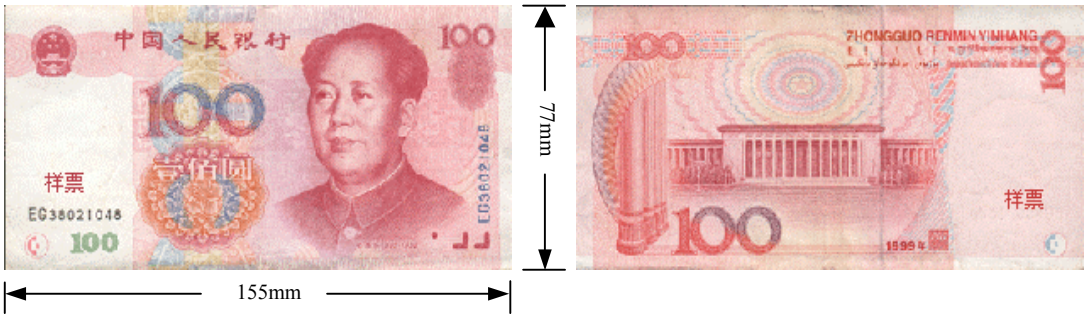


Fig. 2.1 Real images of Renminbi 100

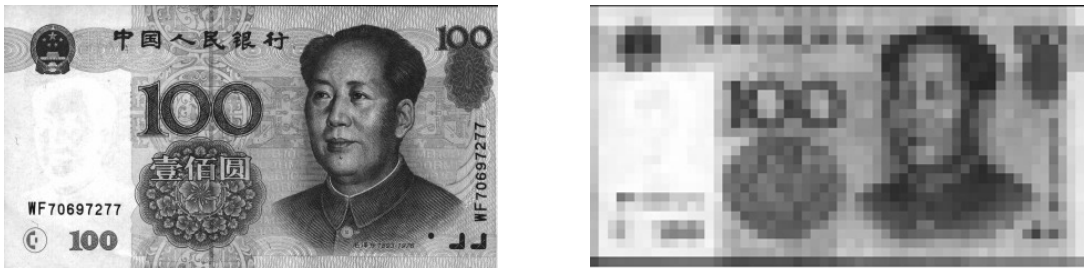
For every par value of paper currencies, there are four directions of images to be collected. A sample of RENMINBI with par value 100¥ is shown in Fig.2.1 and 2.2, in which Fig.2.1 shows real bill images and Fig.2.2 shows collected 4 patterns of bill image.

For every collected currency image, there are so many pixels that the data scale of them is too large. If we directly take each pixel as the input to a neural network, it would give large quantities of weight for every unit in the hidden layer. It reveals that a very large training sample set is indispensable to ensure the weights are well determined, and huge computational resources are also needed to find a suitable minimum of the cost function. In practice such a method is entirely impractical. In this currency recognition system, the technique called pixel averaging is applied to reduce

Fig.2.2 Collected currency images of 4 different directions

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 as shown in the following equation

$$y = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (2.1)$$



Bit Map Data **Transformed Data**

Fig.2.3 Image of a transformed currency

where x_i is the gray-scale value of the *i*th initial pixel in the block, y is the gray-scale value of a block in the transformed image, n is the quantity of the initial pixels included in the block. Then each initial currency image was divided into 32×216 pixel blocks. The transformed data combined with header file including some image information such as pattern number, currency kind, direction, and so on was called NN data. The scale of the currency images was compressed effectively by the NN data. The right image of Fig.2.3 shows the transformed currency.

2.2 Edges Detection

After getting the transformed data of the paper currencies, the next step is to detect the edges of currency image out of black background. First, the sum of pixel values of each row and column of the currency image are calculated. Then we select two appropriate thresholds called 'cho' and 'tan' manually being as the offsets of gray scale level projection in the long and short side of the currency, and hence the first and the last row

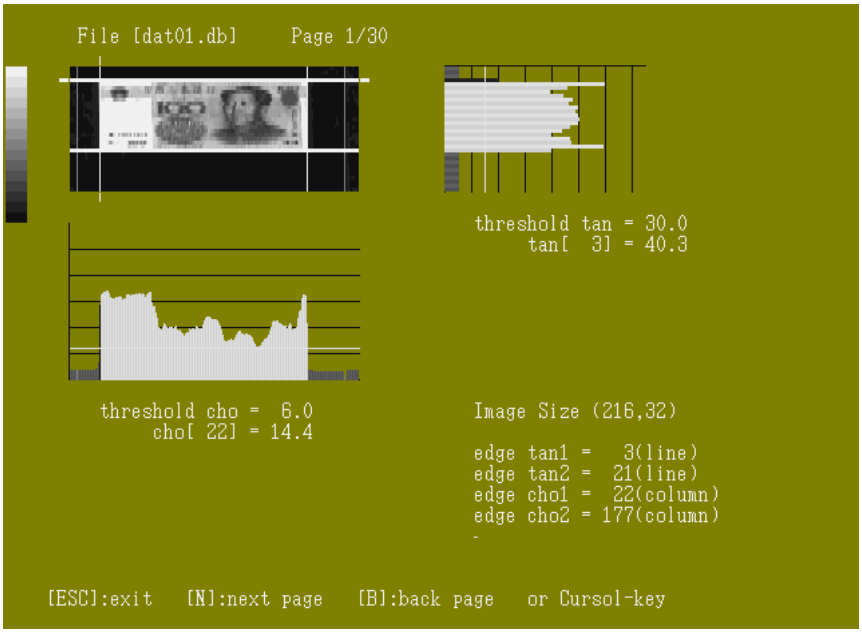


Fig. 2.4 Edges Detection with Thresholds

(or column) whose gray scale level is greater than the selected respective threshold were considered as the long or short edges of the currency image. Fig 2.4 illustrates a menu of the edges detection procedure of the data analysis system with two thresholds. As can be seen from this figure, the thresholds 'cho' and 'tan' are 6.0 and 30.0 respectively. The pixel values of the first column and row are 14.4 and 40.4, which are greater than the corresponding thresholds. They are therefore decided as a part of the edge of the currency image. You can see the area of this piece on the interface being from the line 3 to line 21 and from column 22 to column 177.

2.3 Feature Extraction

Feature extraction or selection is a pivotal procedure considerably for currency recognition, which effects on design and performance of the classifier intensively. If the differences of selected features are so large, it can easily construct a classifier with good recognition performance. It is difficult to get it with the contrary situation. The essential task of feature extraction and selection is how to find the correspondingly effective features out of many pending features. In our research, the technique of slab and mask processing is applied to extract the features of currencies.

2.3.1 Slab Value

The slab value of an image, given by the Eq.2.2, is the sum of the gray-scale values of all pixels in the image.[11] It can be used to represent the feature of the image to be recognized.

$$V_s = \sum_{i=1}^{n} Vp_i \qquad (2.2)$$

where V_s is the slab value of the image, Vp_i is the gray-scale value of the *i*th pixel, and *n* is the total number of pixels in the image. Fig.2.5 illustrates a black-white image, in which each block represents a pixel. The gray-scale value of a white and a black block are 0 and 1 respectively. Therefore, the slab value given by Eq.2.2 of this image is 25 because there are 25 black blocks in this image. As the slab value is applied, it can

sufficiently reduce the dimensionality of input data and improve robustness of the recognition system. Even if there are some differences of the image, it cannot influence the slab value of the image.

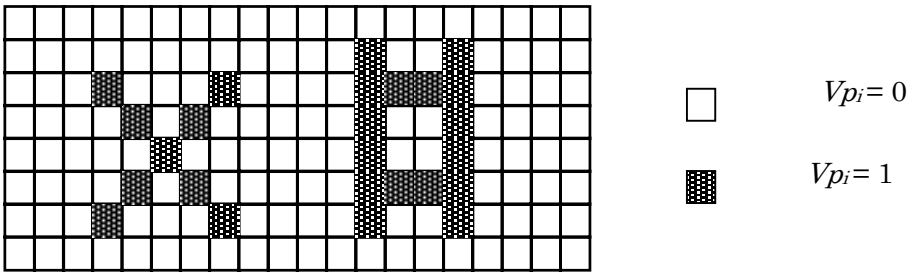


Fig.2.5 Slab value of an image

However, there is a problem that should be faced, which is as shown in Fig.2.6. As can be seen, although there are apparently differences between the two images, the slab values of them are equivalent. If these two slab values are input in the classifier as the unique feature of these two images, it cannot be classified at all. In order to solve this problem, we introduce masks to process the images on the next section.

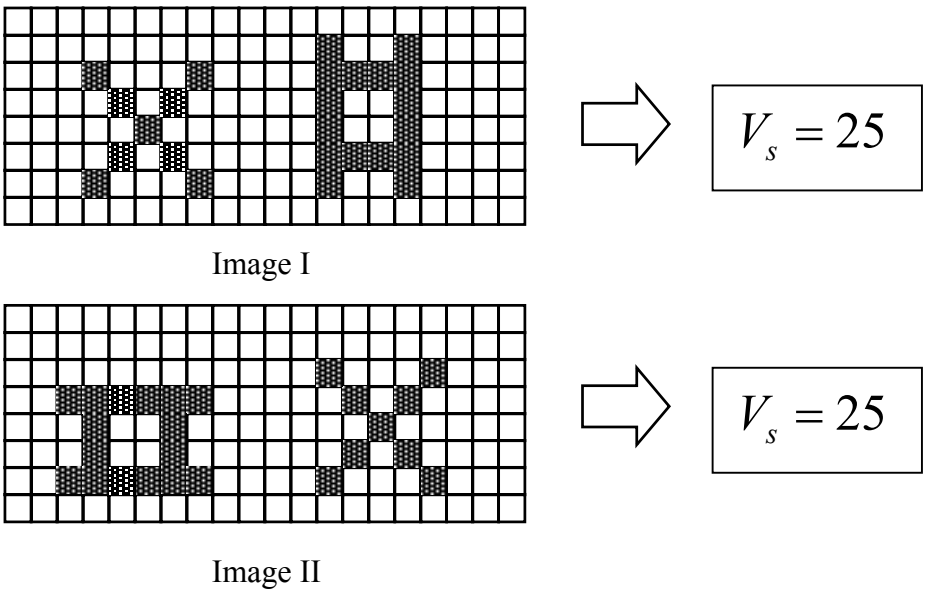


Fig. 2.6 slab values of different images

2.3.2 Mask processing

Just as mentioned in last section, sometimes it is possible that the slab values of two completely different images are equivalent. It will lead to difficulties for recognitions. How to resolve this problem? A mask processing is introduced in this section to attempt to solve the problem.

Let us observe Fig.2.7, which shows the same images in Fig.2.6 with a mask. It can be seen from this figure that the same parts of two images are covered by a mask, which is rectangle including 32 pixels. In this case if the summations of values of all pixels in the two images except those covered by the mask are calculated respectively, it can be seen that these values are 20 and 15 for image I and image II, respectively. These two values not only are different with the slab values obtained in last section but also are different by each other. It reveals that the problem of different images being with the same slab values can be solved using the same mask to process these images. In this case, the slab value has to be redefined, which is the sum of the gray-scale values of all pixels except those covered by the mask in the image. These new slab values are used to replace the original slab values to be representative of the digital features of the image.

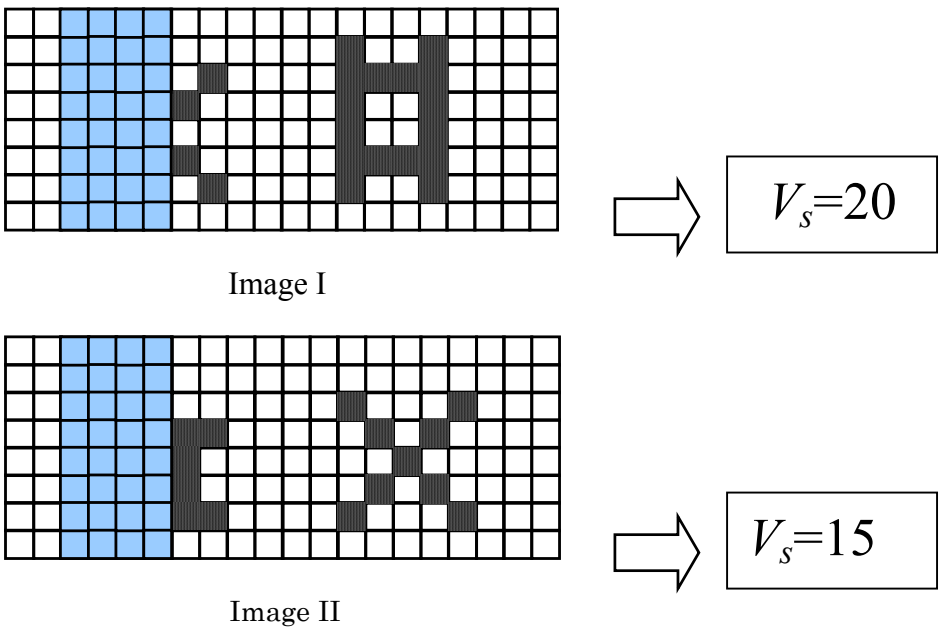


Fig. 2.7 slab values of different images covered by a mask

However, if a mask is used to process images, it is still a problem that there is only one slab value being as the digital feature of each image. Furthermore, for a multi-layer neural network a slab value is apparently not enough to represent the whole features of the recognized objects, and it is not enough distinctly for being recognized them effectively. In order to resolve this problem, a mask set, which includes many the different masks mentioned above, is applied to process the recognized objects. All of these masks differ from the others and covers different areas of the currency image. There is a corresponding slab value for each of masks. Therefore, there are many slab values representing the different features for one currency image. An example of a mask set including two different masks processing an image is shown in Fig. 2.8. As can be seen from this figure, if a mask set including two masks is used to process an image, two different slab values can be obtained, in which one is 15 and the other is 13. they are used to be representatives of the image together.

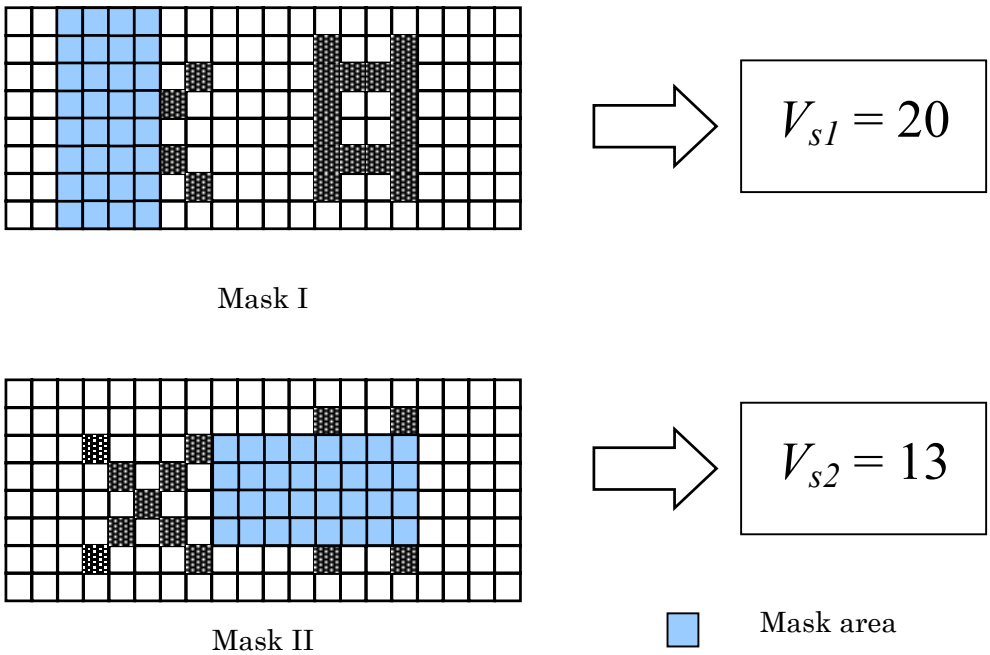


Fig. 2.8 Slab values with different masks

2.3.3 Procedure of feature extraction

After having introduced the slab values and the mask processing, the whole procedure of feature extraction using these two techniques are introduced in this section.

First, depending on the categories of paper currencies to be recognized, a specified area called as feature area is selected on the currency images. The size of this feature area should be smaller than that of the recognized currency image with the smallest size. 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. The procedure of masks and slab values processing is illustrated in Fig.2.9. 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.

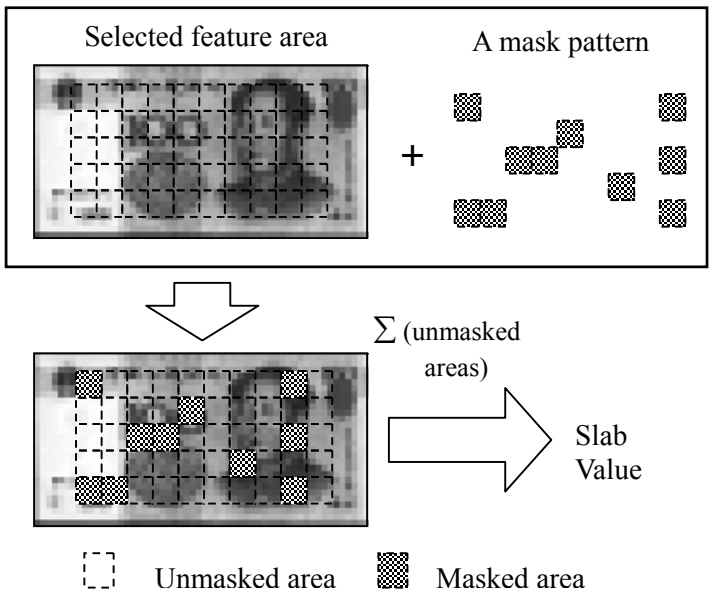


Fig.2.9 Processing of masks and slab values

Chapter 3

RECOGNITION SYSTEM DESIGN WITH NEURAL NETWORKS

3.1 Introduction

As mentioned in the last chapter, after getting features of currencies, it is essential to recognize the pattern of the currencies on the base of these features, which should be practised by an effctive recognition system called classifier. About the design of classifiers, it is mainly divided two categories. One is the conventional design method based on prapobility and statistics theories, which has substantial theoy bases and was successfully applied in many reconition fields, such as image recognition, voice recognition, electronic signals rcognition, and so on. However, the complicated calculating restricts the application range of it, and as more and more complex of

recognition objectives and more and more enhanced recognition demands, the conventional pattern recognition methods have not been satisfied with modern demands in some fields.

In recent twenty years, along with the burgeoning of artificial intelligence (AI) techniques, the pattern recognition techniques are developed adequately. The AI techniques, such as fuzzy techniques and neural networks, are diffusely applied to the design of classifiers. The performance of the classifier employing AI is therefore improved obviously. Especially in recent years neural networks has emerged as a practical technology with successful applications in many fields increasingly. The majority of these applications are concerned with problems in pattern recognition. The network architectures including multilayer perceptrons, Hopfield network, self-organizing network, and so on are more widely applied to solve problem in pattern recognition. Here the neural network is the artificial neural network (ANN), which is a system using physical practicable system to imitate the structures and functions of human brains. The advantages of the ANN are result from the characteristics of it shown as following

Distributed memory and redundancy. The information in ANN is memorized distributedly in large number of neurons. It means the network is redundancy, which leads to that the memory of the network is fault tolerant.

Parallel processing. The input information of a neural network usually is processed in parallel style by large number of neurons.The neural network is not only a processor, but also a memorizer.

Flexibility and self-organizing. The value of weights between each of neurons is variable as the variation of input information. Self-organizing is that the connections between each of neurons are also variable as the variation of input information.

Robustness. Some errors and noises cannot deteriorate the performance of neural networks because of strong interconnections between neurons.

The topological structure of a neural network is another important characteristic. It consists of two kinds: Feed-forward network, and Feedback network. Each unit only accepts input from last layer and output to next layer in the feedforward network. There is not feedback. The feed-forward network generally is composed of several layers. Inputs of the ith layer are just connected with outputs of the i-1th layer. In the feedback network all units are calculating unit that can accept inputs from and output to environments simultaneously. The connections between them are bi-directional.

It is an essential characteristic of ANN to improve the performance of it by obtaining knowledge from environments. Generally, this performance improvement is achieved by regulating parameters of them step by step. There are three kinds of learning mode including supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning is the kind of learning studied in most current research in machine learning, statistical pattern recognition, and artificial neural networks. It is learning from examples provided by a knowledgeable external supervisor. The learning system regulates its parameters according to the error between ideal outputs and actual outputs. The training data consist of pairs of input objects and desired outputs. The output of the function can be a continuous, or can predict a class label of the input. The task of the supervised learner is to predict the value of the function for any valid input object after having seen only a small number of training examples.

Unsupervised learning is a method distinguished from supervised learning by the fact that there is not a supervisor. In unsupervised learning, a data set of input objects is gathered, then is typically treated as a set of random variables. A joint density model is then built for the data set. Unsupervised learning can be used in conjunction with Bayesian inference to produce conditional probabilities for any of the random variables given the others, and be also useful for data compression.

Reinforcement learning is different from any of supervised or unsupervised learning. It is learning what actions can lead to maximize a reward. The learner is not told which actions to take but instead must discover which actions yield the most reward by trying them. Reinforcement learning is a computational approach to understanding and automating goal-directed learning and decision-making.

Generally, in each iteration of deterministic search the search direction is the direction that make the cost function non-increasing, thus the cost function always can converge to a local minimum nearby the initial values in the parameter space. Since it is possible that there are many low-qualitative local minima on the surface of the cost function, it is significant to sieve appropriate initial points of parameters. The classical back propagation (BP) algorithm with the gradient descent method, and the conjugate gradient method belong to this method.

The stochastic search is explore the optimal solution during the stochastic changing of parameters, in which the direction of each iteration search is stochastic, the search can make progress along with the direction of increasing the cost function with certain conditions. This method can prevent the search from plunging into low-qualitative local minima, but it is time-consuming. Therefore, the better scheme is an effective

combination of the stochastic search and deterministic search. In section 3.5, a new hybrid search method including these two searches will be introduced in detail.

3.2 Conventional recognition system

In our previous researches, we also used the three-layer feed-forward neural network to realize the recognition and classification for many kinds of paper currency and obtained the correspondingly good results on aspect of recognizing currencies having been learned. However, rejection capabilities of this system for unknown patterns were unsatisfied in previous experiments. The rejection capability for unknown patterns is an important performance index for the practical paper currency recognition system.

The neural network used in our previous research is a three-layer feed-forward perceptrons, in which the sigmoid function is used as the activation function in its hidden and output layers. For a unit in the hidden and output layers, its input-output mapping is shown as following

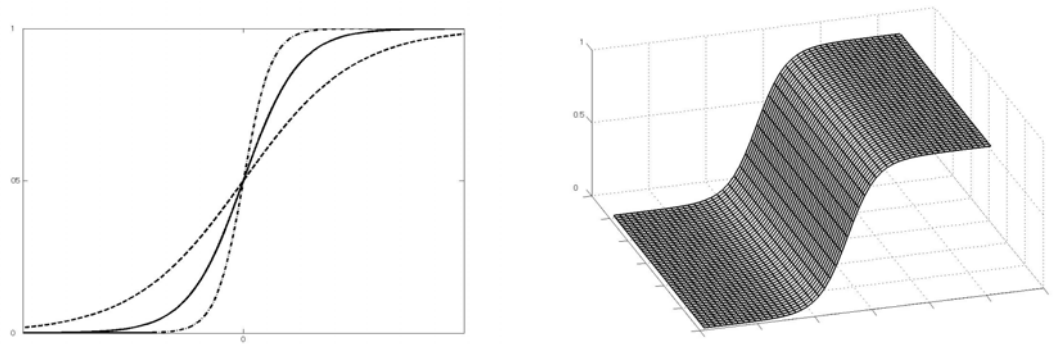
$$s = \sum_{i=1}^{p} w_i x_i - \theta = W^T X - \theta \tag{3.1}$$

$$y = f(s) \tag{3.2}$$

where $x_i \in \mathbf{R}$, $X \in (x_1, x_2, \cdots, x_p)$ is inputs of the unit. $w_i \in \mathbf{R}$, $W \in (w_1, w_2, \cdots, w_p)$ is the corresponding weights of the inputs. $\theta \in \mathbf{R}$ is a threshold. y is the output of the unit. $f(s)$ is sigmoid activation function of given in the following equation

$$f(s) = \frac{1}{1 + \exp(-\alpha \cdot s)} \tag{3.3}$$

where α is slope factor regulating the declining angle of the curve. Its 2-dimensional and 3-dimensional shapes are illustrated in Fig.3.1



(a). 2-dimensional (b). 3-dimensional

Fig.3.1 Illustration of sigmoidal function

Fig.3.1(a) illustrates curves of the sigmoid function with different slope factors in 2-dimensional space. As can be seen from the illustration, the sigmoid function is non-local, and its activation is infinite, i.e. they are non-zero in infinite domain. The decision regions for classification are formed by cutting the parameter space with the hyperplanes. The system is quite false especially far from the training data regions where separating hyperplanes, extending to infinity, enforce arbitrary classifications. A multi-layer perceptron with sigmoid activation function can separate the classes by using hidden units, which form hyperplanes in the input space as indicated in Fig.3.2. As mentioned above, representations of the hidden unit in the multi-layer percepotron depend on weighted linear summations of the inputs, transformed by sigmoid function, a monotonic activation function. Thus the activation of a hidden unit in a multi-layer perceptron is constant on surfaces, which consist of parallel (d-1)-dimensional hyperplanes in d-dimensional input space.[10] just results from the behaviors mentioned above, the conventional classifier using this kind of neural network can not realize the precise classification, and is unsatisfied with its rejection capabilities for unknown currency patterns.

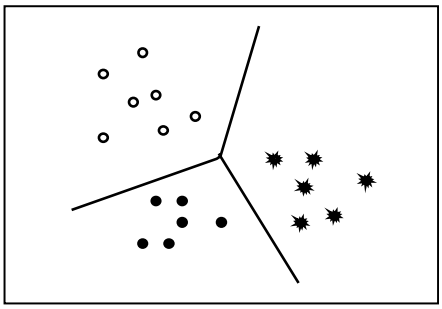


Fig.3.2 Schematic example of classifying three classes with sigmoid function

In order to improve rejection capabilities of currency recognition systems, some combined methods including post-processing such as the linear-template matching

method[4], and the multi-dimensional Gaussian probabilistic density function (PDF) method[8], and multi-NN method such as the neuro-template matching method[7] were applied. However, for the first method, the mean and variance of input data of each leaning class were applied to form a linear template used to reject unknown patterns. Although the rejection capabilities for unknown currency patterns are increased, the recognition capabilities for known currency patterns are abruptly decreased. For the second one, the probabilistic densities of learning currency vectors, which are used to judge the evaluated currencies belong to which class or not, are estimated as the multi-dimensional Gaussian PDF. However, it is difficult to estimate the data distribution in most cases. For the last one, the many neural networks with the same constructions are used to be the template for each currency class having been decided. It is effective on rejecting unknown patterns, but the structures of the system are too complicated. Moreover, the improvement of the rejection capability of these methods do not stem from the neural network directly. Although the performance of the system was improved, the performance of the network could not make progress any more.

In other papers, some researchers proposed the combination of a three-layer perceptron and RBF networks[9], in which the perceptron were used to classification and a RBF network was used to validation for each feature area on paper currencies. The system construction of this combination is too complicated. Furthermore, Eccles proposed using probabilistic neural network (PNN)[37] to reject unknown currencies. However, and the dimension of PNN will be large obviously as the quantity of training data is increased. It is difficult for them to be applied on practical systems.

It is essential to construct a more effective and simpler network structures, which can simultaneously possess the recognition capability for known currency patterns and the rejection capability for unknown currency patterns.

3.3 Proposed recognition system

As mentioned in last section, it is necessary to find a more effective neural network simultaneously having both recognition and rejection capabilities. In this section the proposed neural network are introduced indetail, including the proposed activation function, and the structure of the neural network.

3.3.1 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, in which the activation function of each unit in hidden and output layers is gaussian function, not the conventional sigmoid function. The formula Gaussian function is given as following

$$f(x) = \exp(-\frac{x^2}{2\sigma^2}) \tag{3.4}$$

where σ is width parameter of the Gaussian function. The illustrations of gaussian functions with different widths and altitudes are shown in following Fig.3.3.

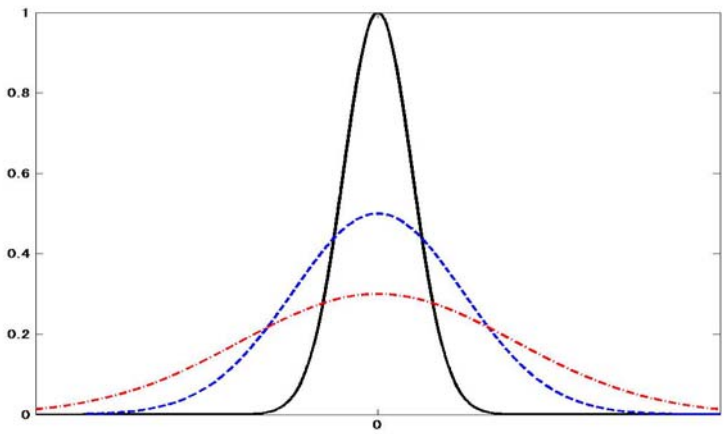


Fig.3.3 Gaussian function

As can be seen from it, its function value approaches zero as the width approaches infinity. The Gaussian function, differing from the sigmoidal function, is a kind of typical localized function. The active range of the Gaussian function is controlled by the width parameter. The range of activation will be larger as the widths are larger with the same altitude. Hence based on Eq.(3.1) and Eq.(3.2), all the weighted inputs fed to each unit in the hidden and output layers are summed up, and produce outputs governed by

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{X}, \xi) = \exp[-\frac{(\boldsymbol{W} \cdot \boldsymbol{X} - \theta)^2}{2\sigma^2}] \tag{3.5}$$

where y is the output of each unit for input vector $\boldsymbol{X}$ in hidden and output layers, $\xi \in \{W, \theta, \sigma\}$ is a set of parameters to be optimized. The output of a single gaussian unit governed by Eq.(3.5) as a function of two input variables is a ridge-like function shown in Fig.3.4. Comparing it with the Fig.3.1(b), which illustrates a sigmoid function with two inputs and one output, we can find there are some differences between them. The activation of the sigmoidal unit is non-local, the input space is directly divided two parts by a single sigmoidal unit. For the proposed Gaussian unit, its output is a ridge-like function. On the direction of $\boldsymbol{WX} = \theta$ the ridge stretch out to infinity, and on the other direction the active range is localized and is controlled by the width parameter σ. The unit can accept the data distributing along with the direction of the ridge, and reject the data distributing along with the orthogonal directions of the ridge by regulating the width parameter–sigma. On the other hand, this ridge-like function also can be considered as be obtained by adding the two sigmoid units on mutually reversed directions together. Hence the network with this activation function has potentials to improve rejection capabilities on promise of ensuring recognition capabilities of the system.

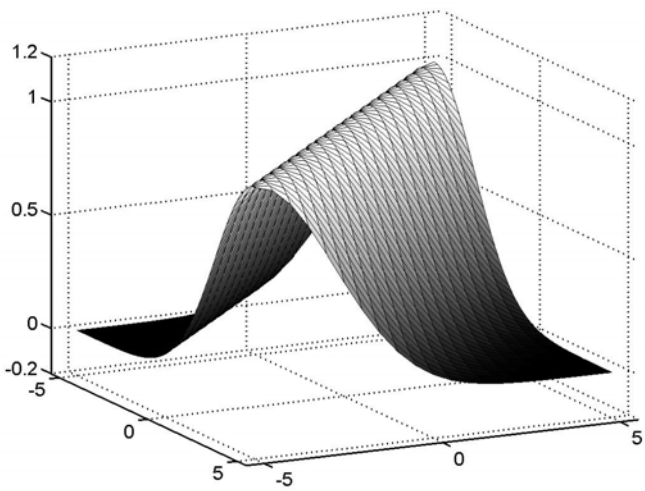


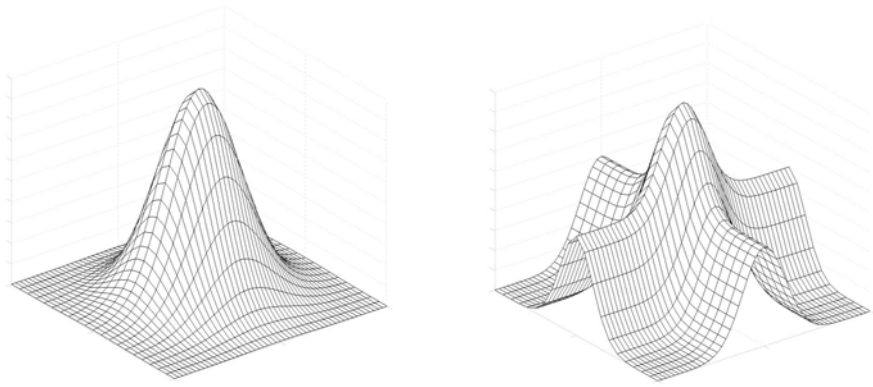
Fig.3.4 Separating surface of a proposed gaussian function with two input variables and a single output

3.3.2 Comparisons with similar activation functions

Once the Gaussian function is mentioned in neural networks, we will consider radial basis function (RBF) networks, which is a type of artificial network for applications to problems of supervised learning. It is non-parametric models. The RBF network is similar with the MLP in that it is a multi-layer, feed-forward network. However, unlike the MLP, the hidden units in the RBF are the "Radial Basis Function", whose output is governed by the Eq.(3.6), a statistical transformation based on a Gaussian distribution.

$$y_i = \exp[-\frac{\| \boldsymbol{X}_i - \boldsymbol{\theta}_i \|^2}{2\sigma_i^{\ 2}}] \tag{3.6}$$

It is well known a RBF network use distance to a prototype vector followed by transformation with a localized function (Gaussian function). The activation of an activation function is therefore constant on concentric hyperspheres. So it is easy for us to attempt to use it to improve the rejection capability of the currency recognition system. Moreover, in our initial stage we ever attempted to use it in our system to improve rejection capabilities for unknown patterns. However, the result of experiments revealed the recognition capabilities for known patterns decreased obviously comparing with that of the sigmoidal activation function in this case. If the system cannot ensure the recognition ability, it is no any meaning to discuss improving the rejection ability. It



(a). RBF (b).Gaussian bar function

Fig.3.5 Separating surfaces of the RBF unit and Gaussian bar function unit with two input variables and one output

is caused by the characteristic of the basis function of RBF formed completely closed hyperspheres in any direction at input space. The responses of the RBF are completely localized. By comparing the Eq.(3.5) and Eq.(3.6), it can be found that the $\boldsymbol{\theta}_i$ in Eq.(3.6) is a vector representing the center of certain class in input variables, and the θ_i in Eq.(3.5) is just a scalar, it is a bias or a threshold of the proposed activation function. Thus the output of the proposed function is a ridge-like hypersurface, not a hypersphere.

$$y_i = \sum_j w_j \exp[-\frac{(x_j - \theta_{ij})^2}{2\sigma_{ij}^2}] \tag{3.7}$$

A semi-local function, Gaussian bars function given in Eq.(3.7), was also considered. For a unit with two input variables and one output, as the activation function is RBF, and gaussian bars function respectively, the corresponding separating surfaces are illustrated in following.

By comparing the shapes of them, it can be found that there are not only a central bump just like that of RBF, but also many other ridges stretching out to infinity for Gaussian bar functions. These ridges can be removed by the action of the next layers units, which effectively provide a soft threshold to get the co-isolated central bumps. The space distribution of the separating surface of Gaussian bar function makes it more flexible on classification than RBF. However, for a unit of Gaussian bar functions with N input variables, there are $3N$ adjustable parameters need to be learned and regulated. It is too much. It will be influence the speed of learning procedure and possibly make training slower. Moreover, it is possible to influence generalization properties with more learning parameters. So we attempted to use this proposed activation function.

Now consider the proposed function again. Just as mentioned above the proposed function is a ridge-like function just like in Fig.3.4, The activation of the proposed function is non-localized along the direction of ridges and it is localized on the other directions. We attempt to add two pair of the orthogonal ridges with the same parameters except for positions together, which can be realized by superposing the outputs of 4 proposed hidden-layer units. The illustration of the separating surface is shown as in Fig.3.6. As can be seen from it, there are 4 bump-like structures just like that of RBF at the intersections between them, and there are also several ridges present which stretch out to infinity. The distribution, the size, and the activation range of these central peaks are all decided by the distribution, the direction, and forms of the corresponding ridges. The directions, the positions, and the forms of these ridges are influenced by the corresponding parameters of these units, e.g. weights, bias and width. It can be imagined in multi-dimensional input space, several proposed hidden units can

form some closed spaces just like that formed by RBF at some areas, and it still has considerable recognition capabilities at the other areas relying on the existence of the ridges. It is essential for those data do not be included in those closed spheral spaces. It makes the units be able to further classify this sort of data. Furthermore, for a unit of the proposed function with *N* input variables, there are only *N*+2 adjustable parameters need to be learned and regulated.

Therefore, with the appropriate position, direction, and width, the proposed neural network has effective recognition and rejection capabilities. It is more flexible on aspect of improving both recognition capabilities for known currencies and rejection capabilities for unknown currencies. It can find the tradeoff of these two capabilities.

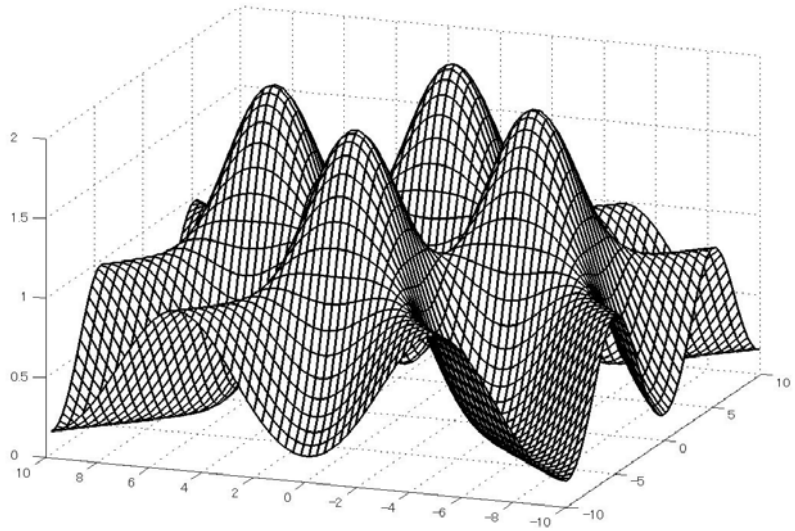


Fig.3.6 Separating surface of adding two pair of the orthogonal ridges with the same parameters except for positions together

3.3.3 Structures of the network

In this system, a three-layered FNN is used, its structure is decided as following rules. The number of nodes 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 nodes at output layer equals to that of currency patterns to be

recognized; the number of nodes in hidden layer is between that of the nodes in input layer and output layer.

3.4. Parameter determination with optimizing algorithms

In this section, the problem of learning in neural networks will be formulated in term of minimization of an error function *E*. this error is a function of the adaptive parameters including weights, biases, and widths in the network. the problem of minimizing continuous, differentiable functions of many variables is one which has been widely studied, and many of the conventional approaches to this problem are directly applicable to the training of neural networks. In this section several of employed algorithms practically in this system will be explained in detail.

Before introducing the algorithms in detail, the error function and its error surface will be reviewed.

3.4.1 Error function and error surface

The central goal in network training is not to memorize the training data, but rather to model the underlying generator of the data, so that the best possible predictions for the output vector can be made when the trained network is subsequently presented with anew value for the input vector. The aim of the network training is that for the mapping of a network between inputs and outputs: $y = f(\boldsymbol{X}, \boldsymbol{\xi})$, $\boldsymbol{\xi} \in \Lambda$, if given a set of training patterns $(\boldsymbol{X}_m, \hat{y}_m)$, $m = 1 \sim M$, exploring a set of optimal $\boldsymbol{\xi}^*$ to minimize the error function (cost function or objective function) of the network given by

$$E(\boldsymbol{\xi}) = \frac{1}{2}\sum_{m=1}^{M}\sum_{k=1}^{K}(\hat{y}_m^k - y^k(\boldsymbol{X}_m, \boldsymbol{\xi}))^2 \tag{3.8}$$

where $E(\cdot)$ is the cost function of the network, which is the sum-of-squares error function given by a sum over all patterns in the training set. $y^k(\boldsymbol{X}_m, \boldsymbol{\xi})$ denotes the

output of unit k as a function of the input vector $\boldsymbol{X}_m$ and the parameter vector $\boldsymbol{\xi}$, K is the number of outputs. The quantity $\hat{y}_m^k$ is the target value for output of unit k when the input vector is $\boldsymbol{X}_m$.

The problem addressed in this section is to find appropriate parameters including weights, biases, and widths, which minimize the error function defined in Eq.(3.8). If there is a simple geometrical picture illustrating the error minimization process, it will be useful. However, for most of general networks, in particular those with more than one layer of adaptive parameters, the error function will typically be a highly non-linear function of the parameters, and there may exist many minima all of which satisfy

$$\nabla E = 0 \tag{3.9}$$

where ∇E denotes the gradient of E in parameter space. The minimum for which the value of the error function is smallest is called the global minimum while the other minima are called local minima. Even if for those local minima, their function value may be smaller than that of the convergence criterion. It is therefore possible to having many points that satisfy the convergence criterion including the global minimum if it is existent. On the base of the Eq.(3.8), the error surface depends on structure of networks, training data, activation function, and error function itself deservedly. As mentioned in reference[38] that error surface has greatest diversity close to the center. The difference of the surface is large in that area, and most of local minima distribute there. Far from the center flat horizontal planes occupy large areas. If the range of random initial weights, biases and widths is too broad then it is likely that the starting point lies somewhere on the flat area, and as the result the network can not be trained with gradient-based or local search method.

In this research, the activation function of hidden layer and output layer is the proposed Gaussian function, which is an unmonotonic function as shown in Fig.3.3. The gradient at the peak of it is zero. Comparing to the conventional sigmoid network with same structures, the error surface of the network with proposed activation function is more complex than of the network with conventional sigmoid functions. There will be much more local minima or saddle-points. If we want to stride across these local minima and saddle-points and find the global minimum, there are two selections. One is sieving carefully the initial range of optimizing parameters, thus training can converge to a point according with the criterion quickly. The other is employing a global search that it is insensitive for the initial range of parameters, but it is possible to influence the speed of training.

In this research, some local search methods and their improved methods are employed firstly, thus the initialization of parameters are done carefully. Afterwards, a new global search method combining gradient search and random search together is proposed.

3.4.2 Initialization of corresponding parameters

All of the algorithms considered in our research begin by initializing the parameters including weights, biases, and widths in the proposed network to some randomly chosen values. As we have known that optimization algorithms which proceeding by a steady monotonic reduction in the error function can be stuck in local minima. Therefore, an appropriate choice of initial values of parameters is potentially important in leading the training algorithm to explore a good set of parameters, and in addition it is possible to lead to improvements in the speed of training. In this case, even deterministic algorithms such as gradient descent, which have possibility of escaping from local minima, can show strong sensitivity to the initial conditions. The majority of initialization procedures in current use involve setting the weights to randomly chosen small values. The aim of using random values is in order to avoid problems due to symmetries in the proposed network.

The initial weight values are chosen to be small so that Gaussian activation functions are not driven into the noneffective regions where is far from the centers of Gaussian functions, gradient of the error function is too small, and is a very flat error surface. As mentioned in section 3.3, the distributions of the proposed ridge-like hypersurface are determined by biases, which decide the center of the ridge-like function, the initial bias values are therefore chosen according to distributions of training samples. It also should be small. The widths, which are strongly influence performances of the system, control the active regions of the proposed Gaussian function. The choice of initial width values is therefore prudent very much. The large quantities of experiment should be done.

3.4.3 Back propagation (BP) algorithm

Back propagation algorithm, which was popularized in a paper by Rumelhart, Hinton and Williams (1986), is applied on training of feed-forward neural networks extensively. The application of the BP algorithm involves to stages: during the first stage the input is presented and propagated forward through the network to calculate the output value for

each unit. This output is then compared with the targets, resulting in an error signal for each output unit. The second stage involves a backward pass through the network during which the error signal is passed to each unit in the network and the appropriate parameter changes are made.[29]

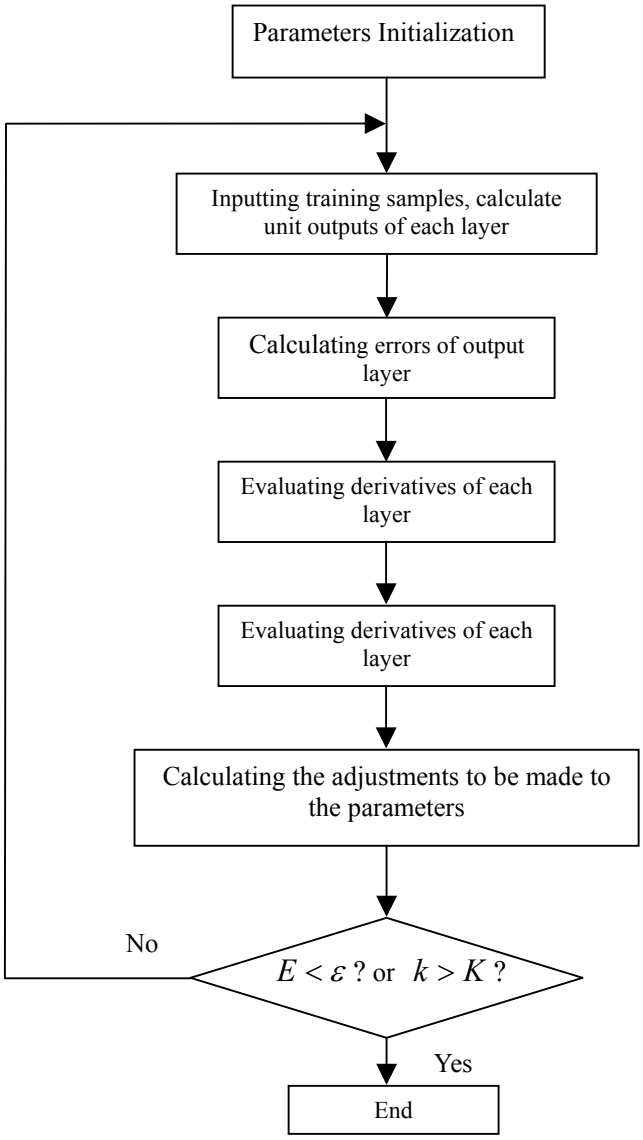


Fig. 3.7 Flowchart of Back-propagation algorithm

Before the development of Back-propagation algorithm, the network training has a serious problem called the credit assignment problem. It is just like mentioned by C. M. Bishop, if an output unit procedures an incorrect response when network is presented with an input vector, we have no way of determining which of the hidden units should be regarded as responsible for generating the error, so there is no way to determine which parameters to adjust or by how much. The Back-propagation algorithm resolved this problem successfully. An iterative procedure is always involved in most of optimizing algorithms with adjustments to the parameters being made in a sequence of steps. Each such step can be distinguished between two distinct stages. In the first stage, the derivatives of the error function with respect to the parameters must be evaluated. The back-propagation algorithm just provides a computationally effective method for evaluating these derivatives. Because of errors being propagated backwards through the network, the term back-propagation is to specially describe the evaluation of derivatives. In the second stage, the derivatives are then used to calculate the adjustments to be made to the parameters needed to implement the steepest descent, conjugate gradient, and quasi-Newton algorithms. The flowchart of BP algorithm is illustrated in Fig.3.7.

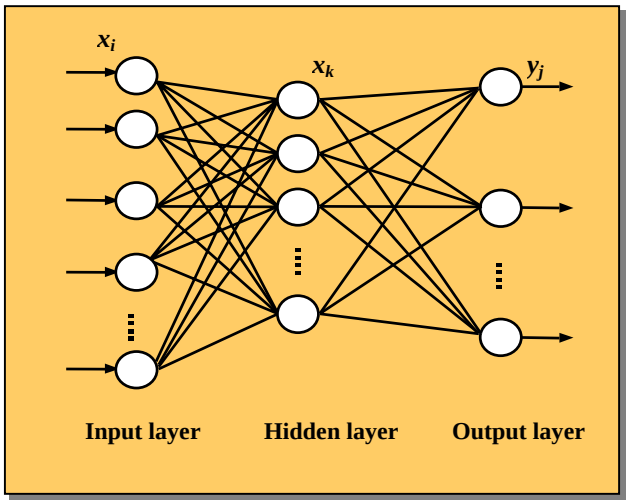


Fig.3.8 Schematic illustration of a proposed 3-layer network

Since the Gaussian function as Eq.(3.5) is employed as the activation function in hidden and outtput units of the network, the derivatives of the parameters of the network including weights, biases, and widths are evaluated using the BP algorithm as shown in following. As can be seen from the Fig.3.8, it is a typical structure of three-layer feed-forward neural networks. The network consists of an input layer, a hidden layer

and an output layer. In which x_i (i = 0, 1, 2, … , m-1) is the output of the ith unit in input layer, x_k (k = 0, 1, 2, … , n-1) is the output of the kth unit in hidden layer, and y_j (j = 0, 1, 2, … , l-1)is the output of the jth unit in output layer. Being based on Eq.(3.8), we also can get the cost functions shown as following

$$E_p = \frac{1}{2}\sum_{j=0}^{l-1}(y_j - t_j)^2 \tag{3.10}$$

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{j=0}^{l-1}(y_j - t_j)^2 \tag{3.11}$$

where t_j is the teacher value of the jth output unit of the network. E_p is the sum-of-squares error of all output units for the training sample p. E is the sum-of-squares error of all training samples. On the base of Eq.3.5, we can get

$$y_j = \exp[-\frac{(\sum_{k=0}^{n-1} w_{kj}x_k - \theta_j)^2}{2\sigma_j^2}] \tag{3.12}$$

$$x_k = \exp[-\frac{(\sum_{i=0}^{m-1} w_{ik}x_i - \theta_k)^2}{2\sigma_k^2}] \tag{3.13}$$

where w_{kj} are weights between the jth unit of output layer and the kth unit of hidden layer. θ_j and σ_j are the bias and width of the jth unit in output layer. w_{ik} are weights between the kth unit of hidden layer and the ith unit of input layer. θ_k and σ_k are the bias and width of the kth unit in hidden layer. Set

$$u = \frac{(\sum_{k=0}^{n-1} w_{kj}x_k - \theta_j)^2}{2\sigma_j^2} \tag{3.14}$$

Calculating the partial derivatives of E with respect to the corresponding parameters (w_{kj}, θ_j , σ_j) in the output layer are shown as following

$$\because \quad \frac{\partial E_p}{\partial y_j} = \sum_{j=0}^{l-1}(y_j - t_j), \quad \frac{\partial y_j}{\partial u} = -e^{-u} = -y_j,$$

and $$\frac{\partial u}{\partial w_{kj}} = \frac{x_k \cdot \sum_{k=0}^{n-1} w_{kj} x_k - \theta_j}{\sigma^2},$$

$$\frac{\partial u}{\partial \theta_j} = -\frac{\sum_{k=0}^{n-1} w_{kj} x_k - \theta_j}{\sigma^2},$$

$$\frac{\partial u}{\partial \sigma_j} = -\frac{(\sum_{k=0}^{n-1} w_{kj} x_k - \theta_j)^2}{\sigma^3}$$

$$\frac{\partial E_p}{\partial w_{kj}} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial u} \cdot \frac{\partial u}{\partial w_{kj}}$$

$$\frac{\partial E_p}{\partial \theta_j} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial u} \cdot \frac{\partial u}{\partial \theta_j}$$

$$\frac{\partial E_p}{\partial \sigma_j} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial u} \cdot \frac{\partial u}{\partial \sigma_j} \qquad \therefore$$

$$\frac{\partial E_p}{\partial w_{kj}} = -\sum_{j=0}^{l-1} (y_j - t_j) \cdot y_j \cdot \frac{\sum_{k=0}^{n-1} w_{kj} x_k - \theta_j}{\sigma^2} \cdot x_k \tag{3.15}$$

$$\frac{\partial E_p}{\partial \theta_j} = \sum_{j=0}^{l-1} (y_j - t_j) \cdot y_j \cdot \frac{\sum_{k=0}^{n-1} w_{kj} x_k - \theta_j}{\sigma^2} \tag{3.16}$$

$$\frac{\partial E_p}{\partial \sigma_j} = \sum_{j=0}^{l-1} (y_j - t_j) \cdot y_j \cdot \frac{(\sum_{k=0}^{n-1} w_{kj} x_k - \theta_j)^2}{\sigma_j^3} \tag{3.17}$$

Calculating the partial derivative of E with respect to the corresponding parameters(w_{ik}, θ_k , σ_k) in the hidden layer are shown as following, set

$$v = \frac{(\sum_{i=0}^{m-1} w_{ik} x_i - \theta_k)^2}{2\sigma_k^2} \tag{3.18}$$

$$z = \sum_{k=0}^{n-1} w_{kj} x_k - \theta_j \tag{3.19}$$

$$\frac{\partial E_p}{\partial w_{ik}} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial u} \cdot \frac{\partial u}{\partial z} \cdot \frac{\partial z}{\partial x_k} \cdot \frac{\partial x_k}{\partial v} \cdot \frac{\partial v}{\partial w_{ik}}$$

$$\frac{\partial E_p}{\partial \sigma_k} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial u} \cdot \frac{\partial u}{\partial z} \cdot \frac{\partial z}{\partial x_k} \cdot \frac{\partial x_k}{\partial v} \cdot \frac{\partial v}{\partial \sigma_k}$$

$$\frac{\partial E_p}{\partial \theta_k} = \frac{\partial E_p}{\partial y_j} \cdot \frac{\partial y_j}{\partial u} \cdot \frac{\partial u}{\partial z} \cdot \frac{\partial z}{\partial x_k} \cdot \frac{\partial x_k}{\partial v} \cdot \frac{\partial v}{\partial \theta_k}$$

where $\frac{\partial u}{\partial z} = \frac{\sum_{k=0}^{n-1} w_{kj} x_k}{\sigma_j^2}$, $\frac{\partial z}{\partial x_k} = \sum_{k=0}^{n-1} w_{kj}$, and $\frac{\partial x_k}{\partial v} = -x_k$

$$\frac{\partial v}{\partial w_{ik}} = \frac{\sum_{i=0}^{m-1} w_{ik} x_i - \theta_k}{\sigma_k^2} \cdot \sum_{i=0}^{m-1} x_i$$

$$\frac{\partial v}{\partial \sigma_k} = -\frac{(\sum_{i=0}^{m-1} w_{ik} x_i - \theta_k)^2}{\sigma_k^3}$$

$$\frac{\partial v}{\partial \theta_k} = -\frac{\sum_{i=0}^{m-1} w_{ik} x_i - \theta_k}{\sigma_k^2}$$

$$\therefore \quad \frac{\partial E_p}{\partial \sigma_k} = -\sum_{j=0}^{l-1} (y_j - t_j) \cdot y_l \cdot \frac{\sum_{k=0}^{n-1} w_{kj} x_k}{\sigma_j^2} \cdot (\sum_{k=0}^{n-1} w_{kj}) \cdot x_k \cdot \frac{(\sum_{i=0}^{m-1} w_{ik} x_i - \theta_k)^2}{\sigma_k^3} \tag{3.20}$$

$$\frac{\partial E_p}{\partial \theta_k} = -\sum_{j=0}^{l-1} (y_j - t_j) \cdot y_j \cdot \frac{\sum_{k=0}^{n-1} w_{kj} x_k}{\sigma_j^2} \cdot (\sum_{k=0}^{n-1} w_{kj}) \cdot x_k \cdot \frac{\sum_{i=0}^{m-1} w_{ik} x_i - \theta_k}{\sigma_k^2} \tag{3.21}$$

$$\frac{\partial E_p}{\partial w_{ik}} = \sum_{j=0}^{l-1} (y_j - t_j) \cdot y_j \cdot \frac{\sum_{k=0}^{n-1} w_{kj} x_k}{\sigma_j^2} \cdot (\sum_{k=0}^{n-1} w_{kj}) \cdot x_k \cdot \frac{\sum_{i=0}^{m-1} w_{ik} x_i - \theta_k}{\sigma_k^2} \cdot \sum_{i=0}^{m-1} x_i \tag{3.22}$$

The derivatives evaluated above, which are the derivatives of the optimizing parameters with respect to E_p, are the first stage of the BP algorithm. The differences of the derivatives with respect to E_p and E will be explained in detail in next section. The second stage then will be executed to get optimal parameters minimizing the cost

function.

3.4.4 Gradient descent

Batch and sequential versions The gradient descent, sometimes also called steepest descent, is one of the simplest training algorithm of networks, and is applied the most extensively. There are two versions including batch version and sequential version. At the kth step, the parameter vecter ξ is updated iteratively according with the case that it is moved a short distance in the direction of the greatest rate of decrease of the error function, i.e. in the direction of the negative gradient at somewhere of the error surface in the parameter space. In the batch version, the increment of the parameter is governed by

$$\Delta\xi(k) = -\eta\nabla E\,|_{\xi(k)} \tag{3.23}$$

where ∇E represents the gradient with respect to all training samples E. In sequential version, the gradient of the error function is evaluated for just one sample at a time, and the parameters are updated using

$$\Delta\xi(k) = -\eta\nabla E_p\,|_{\xi(k)} \tag{3.24}$$

where the different training samples p in the training set can be selected sequentially or randomly. η is learning rate or called step size. It controls the moving distance of the parameter in each update and should be small sufficiently. The new value of the parameter vector at kth iteration thus is shown as following

$$\xi(k) = \xi(k-1) + \Delta\xi(k) \tag{3.25}$$

As mentioned above in the sequential gradient descent the parameter is updated with every training sample inputted in the network, so it has many potential advantages over the batch gradient descent in which is the parameter is updated with all of training samples in the training set. The advantages include: one is that it has possible for the parameter exploring to escape from local minima since this approach is a stochastic algorithm. The other is that redundancies of the training set do not influence the computation scales of training.

In this research, the sequential version of gradient descent is used relying on its

advantages. But just because of being stochastic in each iterative search, the different orders of training samples will lead to find out the different solution with the different performances. We also met this problem in our experiments. In batch version of gradient descent, each update of the parameter is for all samples in the training set, so the direction in each step of iteration is not as stochastic as that of sequential version. It can resolve this problem to a certain extent degree. However, it is easily to be stuck in a bad local minimum and remain there indefinitely. An appropriate tradeoff between these two versions is therefore made in this research to solve the problem that sample orders influence performance of the network. The training set is divided into several blocks and each of them is used sequentially to evaluate updates as if it is one training sample. The details can be found in chapter 5.

Learning rate As mentioned above, The learning rate η in Eq.(3.24) and Eq.(3.25) should be small enough to guarantee that the average direction of iterative search in parameter space is the approximately contrary with that of the local gradient in order to realize the steady decrease of the error function. In the case of with a constant η, If it is too large, the overshoot will be occurred, and it will induce an increase and instabilities of the error function that manifest as a divergence of the error. On the contrary, if it is too small the iterative search will be extremely slow and difficult to stride over adjacent local minima. The adaptively chosen learning rate is therefore applied in this research. First, in each step of iteration, the average error in last certain steps including this step are calculated, and it is compared with the last average error. If it is greater than the last one, which means the present direction of search cannot induce the decrease of the error, the corresponding counter will be increased 1. Once this counter is a certain value, being as a punishment, the learning rate will be decreased a little. Conversely, if it is less than the last one, another corresponding counter will be increased 1. Once this counter is a certain value, being as reinforcement, the learning rate will be increased a little.

Momentum and Oscillation terms A simple and applied widely method to improving the iterative properties of the back-propagation algorithm is to add a momentum term, which increases inertia and smoothes out oscillations of the iteration effectively, to the gradient descent formula Eq.(3.24) and Eq.(3.25). The update of the parameter governed by

$$\Delta\xi(k) = -\eta\nabla E\,|_{\xi(k)} + \alpha\Delta\xi(k-1) \tag{3.26}$$

$$\Delta\xi(k) = -\eta\nabla E_p\,|_{\xi(k)} + \alpha\Delta\xi(k-1) \tag{3.27}$$

where $0 < \alpha < 1$, it is the momentum parameter. α is more adjacent to 1, the inertia of the system is greater and the overshoot is more serious. The Z-transform form of Eq.(3.27) is given by

$$\frac{\Delta\xi(z)}{\nabla E_p(z)} = \frac{-\alpha}{1-\eta z^{-1}} \tag{3.28}$$

where it is a typical representative of a IIR digital low-pass filter. It effectively strengthens low frequent components and weakens high frequent components of signals. It therefore can effectively restrain oscillations near the minimum. Furthermore, in order to stride over local minima and escape the search from local minima, a oscillation term is combined into Eq.(3.27) and the formula is then given by

$$\Delta\xi(k) = -\eta\nabla E_p\,|_{\xi(k)} + \alpha\Delta\xi(k-1) + \beta\Delta\xi(k-2) \tag{3.29}$$

where β is the oscillation parameter, -1< β < 0. it means that the fractions of last two steps are included in the current iteration. The itrtation has considerable memories for the last two iterations. In other words, iterations of the last two steps are affecting the current iteration cosiderablely.

Optimizing of weights and biases For the weights and biases of the proposed network, the improve back-propagation algorithm as shown in Eq.(3.29) is used to optimize them. The updates of them are therefore given by

$$\Delta w_{ij}(k) = -\eta_\omega\nabla E\,|_{w_{ij}(k)} + \alpha_\omega\Delta w_{ij}(k-1) + \beta_\omega\Delta w_{ij}(k-2) \tag{3.30}$$

$$\Delta\theta_j(k) = -\eta_\theta\nabla E\,|_{\theta_j(k)} + \alpha_\theta\Delta\theta_j(k-1) + \beta_\theta\Delta\theta_j(k-2) \tag{3.31}$$

where w_{ij} denotes the weight of the *j*th unit of current layer and the *i*th unit at last layer. θ_j is the bias of the *j*th layer of current layer. α, β and γ are the learning rate, momentum factor and oscillation factor, respectively. Generally, the values of learning rates, momentum parameters and oscillation parameters of weights and biases are different.

3.4.5 Conjugate gradient

According to our research, the width selection of the gaussian function is so important that it influences the convergence and the generalization capability of the

neural network. For acquirement of a set of appropriate widths to optimize the performance of the system, the conjugate gradient algorithm is performed to train the widths by minimization of the error function. For general non-linear error functions, the local Hessian matrix need not be positive definite. The search directions defined by the conjugate gradient algorithm need not then be descent directions. It means the direction in each iterative search of widths is not that of negative gradients just like in gradient descent, but that of conjugate gradients. The search directions of width selected iteratively that are noninterfering in the sense that successive minimizations along theses directions do not undo the process made by previous minimizations. In the *k*th iteration the width gradients with respect to the error function is given by

$$g(k) = \nabla_{\sigma} E(\sigma)\Big|_{\sigma=\sigma(k)} \tag{3.32}$$

Hence, the width σ are updated on the base of the following equations

$$\sigma(k) = \sigma(k-1) + \eta_{\sigma} \cdot d(k) \tag{3.33}$$

$$d(k) = \begin{cases} -g(k), & k=0 \\ -g(k) + \gamma_{k-1} d(k-1), & k \geq 1 \end{cases} \tag{3.34}$$

where

$$\gamma_{k-1} = \frac{g(k)^T g(k)}{g(k-1)^T g(k-1)} \tag{3.35}$$

As indicated in last subsection, the form of gradient descent with momentum involves learning rate determining the step length and momentum parameter controlling the momentum. Because of selected the initial values and the applied algorithm in this non-linear optimization problem, it is not easy to be enmeshed in local minima of the error function. A problem with gradient descent is how to determine value of the momentum parameter. Its larger or smaller value will influences the convergence properties at all. It only can be determined with experiences or reduplicative experiments. The conjugate gradient method can be regarded as a form of gradient descent with momentum, in which the momentum parameter is determined automatically at each iteration by parameter γ_{k-1}. Just because of the selected initial values and the applied algorithm in this non-linear optimization problem, the iterative search of widths of the proposed activation function is not easy to be enmeshed in local minima of the error function.

3.5 Hybrid-learning algorithm for widths

The width parameter of Gaussian function controls active range of the hyper-surface, it is considerably important for generalization capabilities of the network. Depending on our previous researches, improper widths will lead to bad recognition and rejection ratio, and more seriously, the iteration cannot converge and the network cannot be used at all. Therefore, we explore the optimal width parameters of Gaussian function applying a hybrid-learning algorithm. It includes two steps, one is searching local minima by employing the conventional gradient descent method adding a momentum term, and the other is extricating the iteration search from local minima. As the cost function is distinguished to getting trapped in a local minimum, a random search with the downhill simplex method is employed. First, A random vector is mixed in the increment terms of width parameters, then coefficients of the random vector and the gradient term are optimized simultaneously using the downhill simplex method.

Before addressing on the detail of this hybrid–learning algorithm, the overview of the downhill simplex method is provided in next subsections.

3.5.1 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 simplex size is continuously changed and mostly diminished to make it small enough to contain the minimum with the desired accuracy.[16] 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. Its iterative steps are indicated briefly as following.

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.

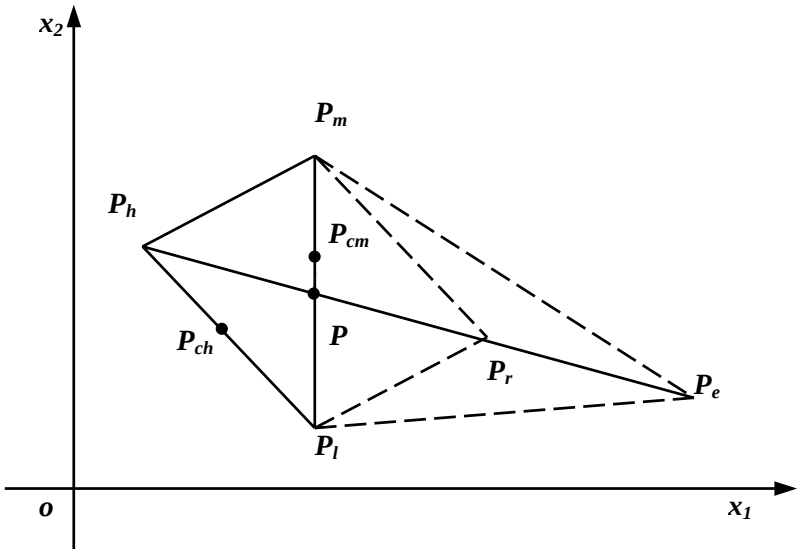


Fig.3.9 Simplex method in 2-dimentional space

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.3.9 illustrates the downhill simplex method in a 2-dimentional space (plane of x_1-x_2), in which $\boldsymbol{P}_h$, $\boldsymbol{P}_m$ and $\boldsymbol{P}_l$ are corners of a current triangle simplex according with $f(\boldsymbol{P}_h) > f(\boldsymbol{P}_m) > f(\boldsymbol{P}_l)$. $\boldsymbol{P}$ is the centroid of $\boldsymbol{P}_m$ and $\boldsymbol{P}_l$, $\boldsymbol{P}_r$ and P_e are the reflection and expansion points respectively of $\boldsymbol{P}_h$ along the direction through centroid $\boldsymbol{P}$. $\boldsymbol{P}_{ch}$ and $\boldsymbol{P}_{cm}$ represent respectively the contraction points of $\boldsymbol{P}_h$ and $\boldsymbol{P}_m$ toward $\boldsymbol{P}_l$.

3.5.2. Proposal of hybrid-learning algorithm

The illustrations of the flow of the proposed learning algorithm are shown in Fig.3.10.

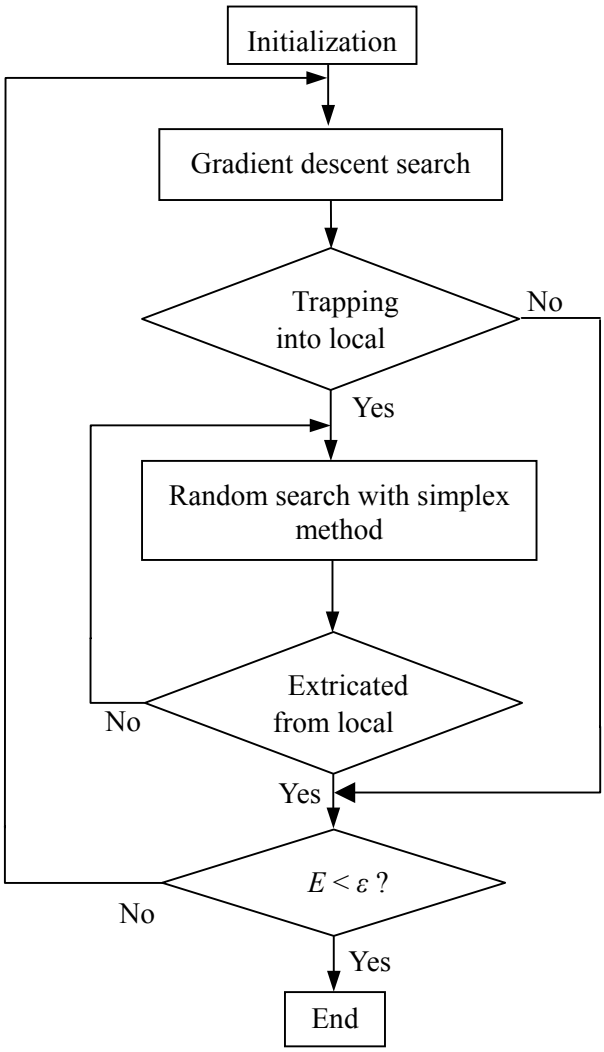


Fig.3.10 Flowchart of the learning algorithm

On the first step of our proposed algorithm, it uses the gradient descent method for searching local minima near the starting point. The update formula of the width vector $\boldsymbol{\sigma}$ is given by

$$\sigma(k) = \sigma(k-1) + \Delta\sigma(k) \tag{3.36}$$

$$\Delta\sigma(k) = -\alpha \cdot \nabla E(\sigma(k)) + \beta \cdot \Delta\sigma(k-1) \tag{3.37}$$

where α and β are the learning rate and momentum coefficient of widths $\boldsymbol{\sigma}$ respectively. $\nabla E(\sigma(k))$ is the gradients of the cost function with respect to $\boldsymbol{\sigma}$ at the kth iteration.

After that, as the cost function gets stuck in local minima and the 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 local minima. 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.

3.5.3 Criterion of being stuck into a local minimum

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

$$\frac{|E_c(\sigma(k)) - E_c(\sigma(k-1))|}{E_c(\sigma(k))} < \varepsilon_T \tag{3.38}$$

where $0 < \varepsilon_T << 1$. It means that as the ratio of the increment of the cost function caused by the variable $\boldsymbol{\sigma}$ 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_0. Then the second stage of the algorithm is executed.

3.5.4 Criterion of being extricated from local minima

In the second stage, if the difference of E_0 mentioned above and the current cost

function value of the $E(k')$ accords with the formula given by

$$E_{c0} - E_c(k') > \lambda E_{c0}, \quad 0 < \lambda < 1 \tag{3.39}$$

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.

3.5.5 Steps of random search with simplex

Once the gradient iteration search is ascertained being stuck into a local minimum, the increment of σ shown in Eq.(5) is transformed by

$$\Delta\sigma(k) = -\alpha \cdot \nabla E(\sigma(k)) + \beta \cdot \boldsymbol{G}(k) \tag{3.40}$$

where $\boldsymbol{G}(k)$ is a gaussian random vector which has the same dimensions with $\boldsymbol{\sigma}$. During searching the appropriate $\boldsymbol{\sigma}$ 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 $\boldsymbol{G}(k)$, 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^* that according $E(\alpha^*, \beta^*)$ with the criterion shown in Eq.(3.38).

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 $\alpha - \beta$ by selecting randomly 3 pairs of α and β, in which the corner is $P_i(\alpha_i, \beta_i)$, 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 of each corner

$$(E(P_i(\alpha_i, \beta_i)), i = 1, 2, 3, \quad \text{let } k = 1.$$

Step.3 Decide the highest corner P^h, the second highest corner P^m and the lowest corner P^l, where *h, m* and $l \in \{1, 2, 3\}$, according with

$$E(P^h) = \max\{E(P_1), E(P_2), E(P_3)\}$$

$$E(P^m) = \max\{E(P_i) \mid i \neq h\}$$

$$E(P^h) = \min\{E(P_1), E(P_2), E(P_3)\}$$

Then calculate the centroid of P^m and P^l, let

$$\overline{P} = \frac{1}{2}(P^m + P^l)$$

Step.4 Execute the reflection operation, let $P_4 = \overline{P} + C_r(\overline{P} - P^h)$, and calculate $E(P_4)$.

Step.5 If $E(P_4) < E(P^l)$, then execute expansion operation. Let $P_5 = \overline{P} + C_e(P_4 - \overline{P})$, and calculate $E(P_5)$, then jump to Step.6;

If $E(P^l) \leq E(P_4) \leq E(P^m)$, then let $P^h = P_4$, and jump to Step.8;

If $E(P_4) > E(P^m)$, then execute contraction operation. Set $E(P^{h'}) = \min\{E(P^h), E(P_4)\}$, Where $h' \in \{h, 4\}$. Let $P_6 = \overline{P} + C_c(P^{h'} - \overline{P})$, and calculate $E(P_6)$, then jump to Step.7.

Step.6 If $E(P_5) < E(P_4)$, let $P^h = P_5$, and jump to Step.8;

Otherwise let $P^h = P_4$, and jump to Step.8.

Step.7 If $E(P_6) \leq E(P^{h'})$**,** then set $P^h = P_6$ and jump to Step.8;

Otherwise execute contraction operation, let $P^i = P^i + C_c(P^l - P^i)$,

where $i \in \{h, m\}$, and calculate $E(P^i)$.

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

If $E(P^*)$ accords with the criterion in Eq.(3.39), 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 $P^l = P_1$, $P^m = P_2$ and $P^h = P_3$, and then return to Step.3.

3.6 Creterion of network outputs

The structures and the training alogrithms haven been addressed in the last sections very much. It is well known that after training there are a series of output of the trained neural network for a input pattern during the evaluation procedure, how to determine wherther this input partten should be rejected or not, and which class should be belonged to will discussed in this section.

As mentioned in section 3.3.2, because of emplying the proposed Gaussian function in all of ouput units, the outputs of them for any input data are all greater than 0 and less than 1. In order to judge the pattern of inuput data, two thresholds Th1and Th2 are selected satifying

$$0 < \text{Th1}, \text{Th2} < 1 \tag{3.41}$$

For the network there are several output units in output layer, each of which corresponds with a class having been decided in training procedure. As a input data is feed to the trained network, output values of these output untis are differnet. If a certain unit whose output value is the largest in all output units, but is less than Th1, this input data will be rejected as unknown patters. Ortherwise, the unit whose output value is less than this largest one but is greater than the other units will be detected. If the differnece of these two values is less than Th2, this input data will also be rejected. Otherwise, if this differnece is graeater than Th2, this input pattern will be considered belonging to

the calss represented by the unit whose output is largest. This is formulized by

$$O_{max} > \text{Th1 and } O_{max} - O_{smax} > \text{Th2} \tag{3.42}$$

where O_{max} denots the largest output, and O_{smax} denots the sub-largest output. If the outputs of a input pattern are accordant with the criterion Eq.(3.42), this pattern will be considered as belonging to a certain known class, otherwise it will be rejected.

Chapter 4
REAL-TIME SYSTEM

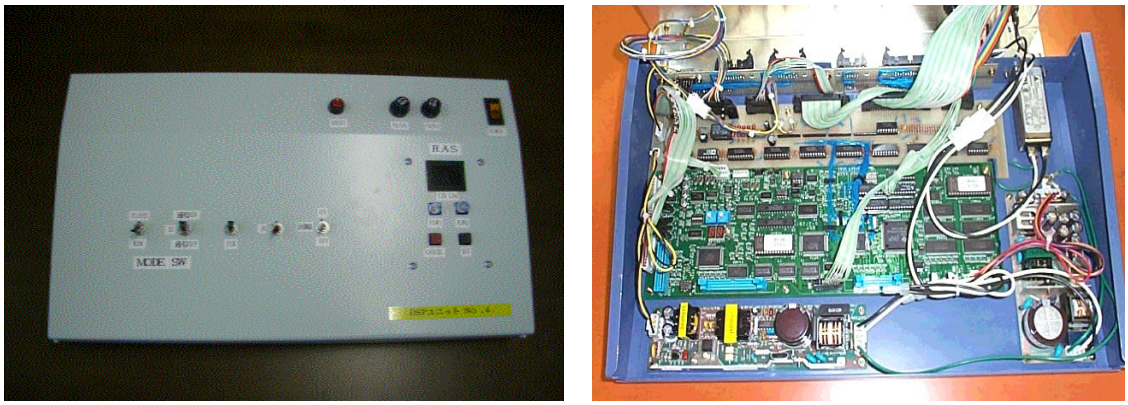
After finishing design of the proposed currency recognition system, in order to further prove the practicability of the system on business developments and applications, such as currency readers and sorters as shown in Fig.4.1, the verified and contrastive experiments of the proposed system are also executed on the real-time system, which are composed of digital signal processor, memories, input/output interfaces, and some essential auxiliary equipments. The control panel and the inside structures of this real-time system are shown in Fig.4.2. The configurations of this real-time system are explained in section 4.1. The introductions of operation modes including the communication between the real-time system and PC, the learning procedure, and the evaluation procedure of this real-time system are described in detail in section 4.2. The relative experiments will be addressed in the next chapter.

(a). Currency reader (b). Currency sorter

Fig.4.1 photographs of currency readers and sorters

4.1 Configurations of the real-time system

The kernel part of this system is a digital signal processor (DSP), which is TMS320C31. Some memories and interfaces are also constructed in this system. The



(a). Control panel (b). Inside structures

Fig.4.2 Photographs of the real-time system

configurations of this system are illustrated Figure 4.3. As can be seen from this figure the DSP connects any kind of instruments by address bus and data bus. The system can communicate with PC with serial ports, and exchange data directly by A/D and D/A converters. The operating status can be displayed on LED. The memory of this system is composed of three parts: flash memory, EPROM, and SRAM, in which the SRAM is main memory, all data accessing to DSP will be through here; the EPROM is used to store the communication programs between this system and PC. All weights, biases, widths and currency information are memorized in the flash memory.

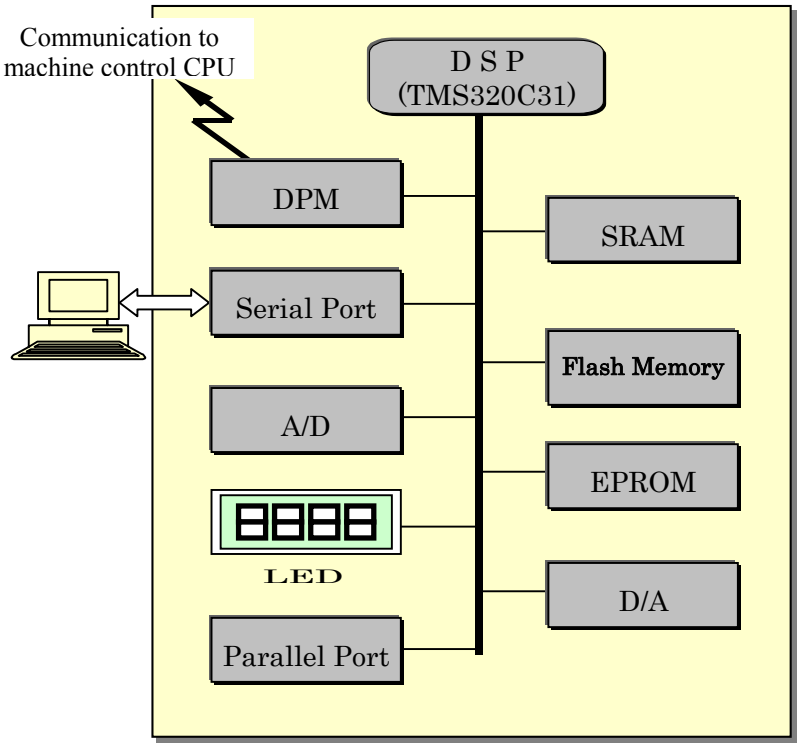


Fig.4.3 configuration illustration of the real-time system

4.2 Functions of the real-time system

In this real-time system, there are three kinds of operating mode being performed, which are communication mode, learning mode, and evaluation mode. The different

combinations of switches on the panel of the system can determine which operating mode is selected. For the communication mode, all the initial parameters of the neural network including weights, biases, and widhts, information of the mask set and paper currency images, and another parameters, are transferred from PC, and are then saved in the flash memory. Then in the learning mode, the program of neural networks and all of the learning data, which are saved on the flash memory, are loaded to the SRAM and from where to upload them to the DSP and start to execute learning operations for getting the optimal network parameters. During the learning procedure, all of the weights, biases, and widths are updated continually in the flash memory until the error function is accordant with the convergence criterion and the learning is finished. Finally, for the evaluation mode, the evaluation program of the network, the converged parameters, and testing or real target data are also loaded to SRAM, which is a bridge of connecting the memories and the DSP, and then are uploaded to the DSP to execute evaluation procedure. The Details of each operation mode are presented on next sections.

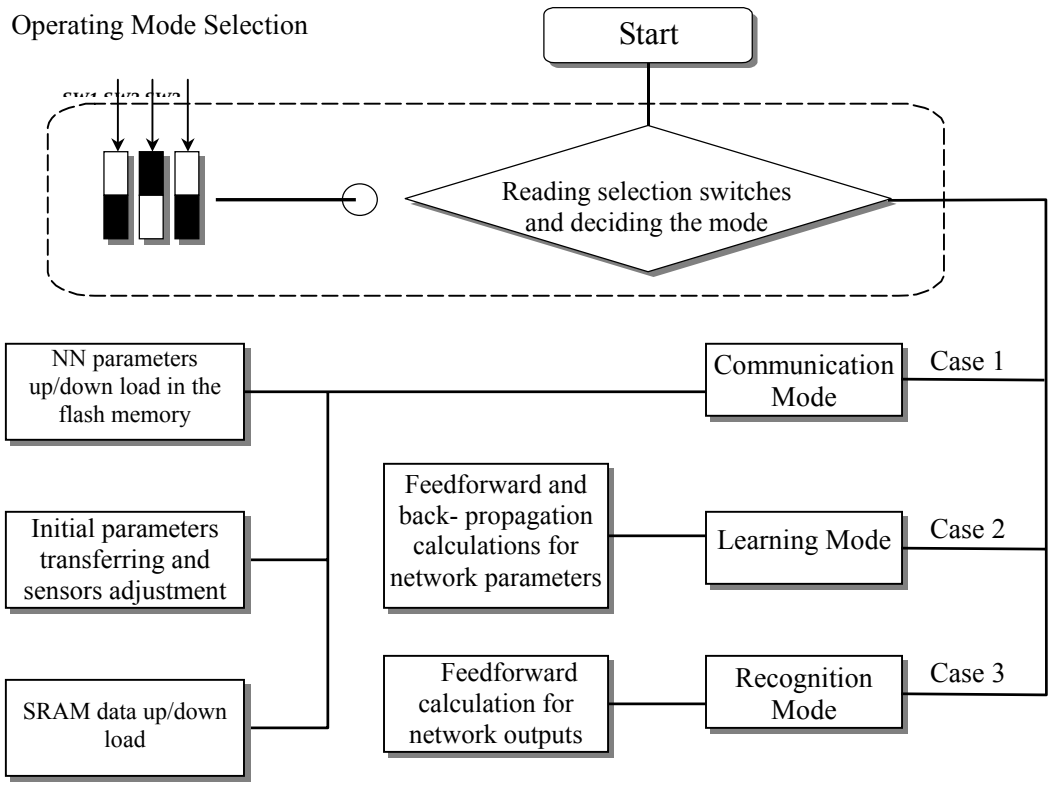


Fig.4.4 Operating mode selections of the real-time system

4.2.1 Communication mode

During the communication procedure of this system, all of the information being relative with the network is download to this real-time system for PC. These information includes:,

1). Structure information of the proposed network. Layers of the network, the number of units in every layer, the activation function, and so on.

2). Programs of the algorithm. Learning programs, evaluating programs, and preprocessing programs

3). Initial information of the network. Initial values of weights, biases, and widths; values of learning rates, momentum parameters; classes and quantities of training samples.

4). Preprocessing information. Size, position, and quantities of masks in the mask set.

5). Image information of paper currencies. Including training samples and paper currencies need to be evaluated.

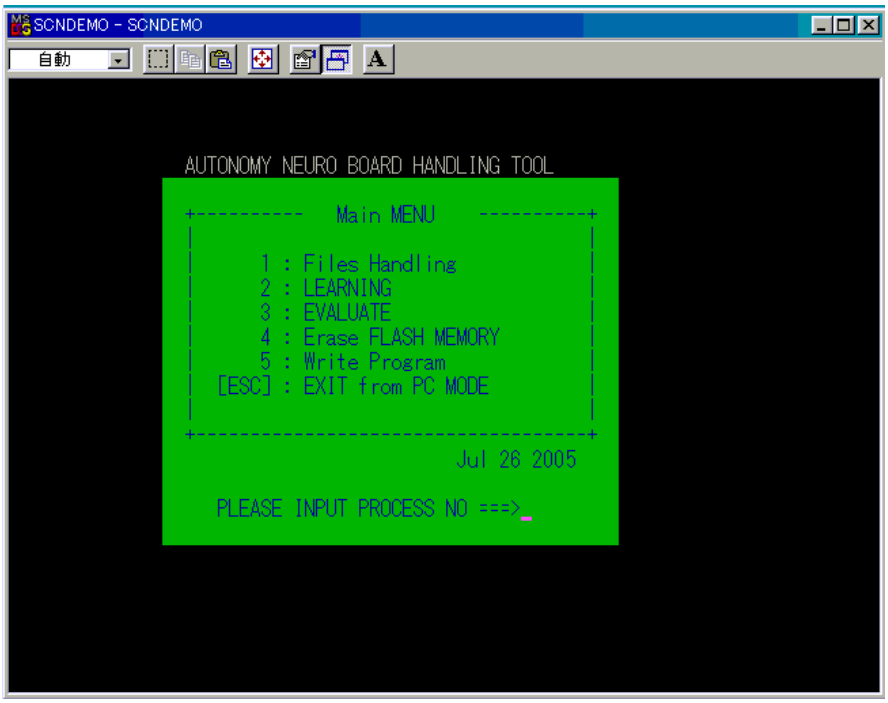


Fig.4.5 Main menu of the real-time system

All of the information is saved in the flash memory for learning and recognition. Fig.4.5 shows the main menu of the real-time system, by which the operating modes can be selected.

4.2.2 Learning mode

During the learning mode of this system, the information including the mask set, the initial values of the optimizing parameters, structure parameters of the network and images of training samples are indispensable. All of these data and the algorithm programs are transferred to the SRAM, which is the main memory of the DSP. Then, the learning procedure is begun to execute, and the parameters such as weights and biases are adjusted automatically with the proposed algorithms, in which the feedforward calculations of inputs and the back-propagation of the errors are all conducted. During this procedure, the relative parameters in each of iterations are also uploaded to the PC, which can display these parameters graphically on real-time. An example of this kind of

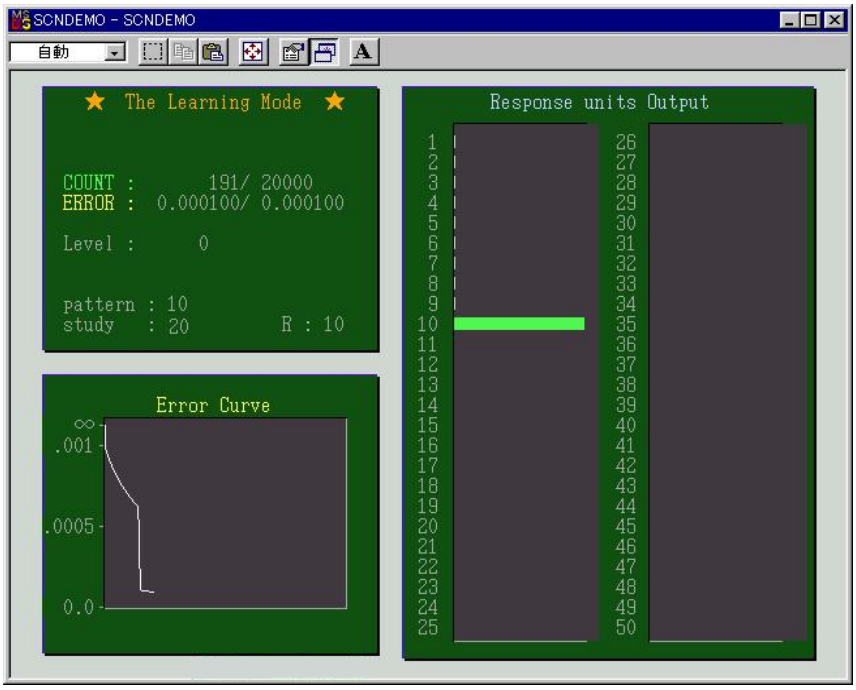


Fig.4.6 Interfaces of the learning mode

graphical display is shown in Fig.4.6, from which we can see the convergence procedure of the error curve, the steps of the iteration, the convergence criterion, and some other essential information. After learning, all of the converged network weights are then saved in the flash memory to the evaluation on the next step.

4.2.3 Evaluating mode

During the evaluation mode of the system, for any of input data that needs to be evaluated, the network parameters having been saved after the learning procedure are used to only realize the feedforward calculation for evaluating whether or not should be accepted, or which class should belong to. During this procedure, the real-time data is also uploaded to the PC and can be displayed on the screen graphically. Fig.4.7 illustrates this interface. Operators can be sure of the results of the evaluation clearly from it.

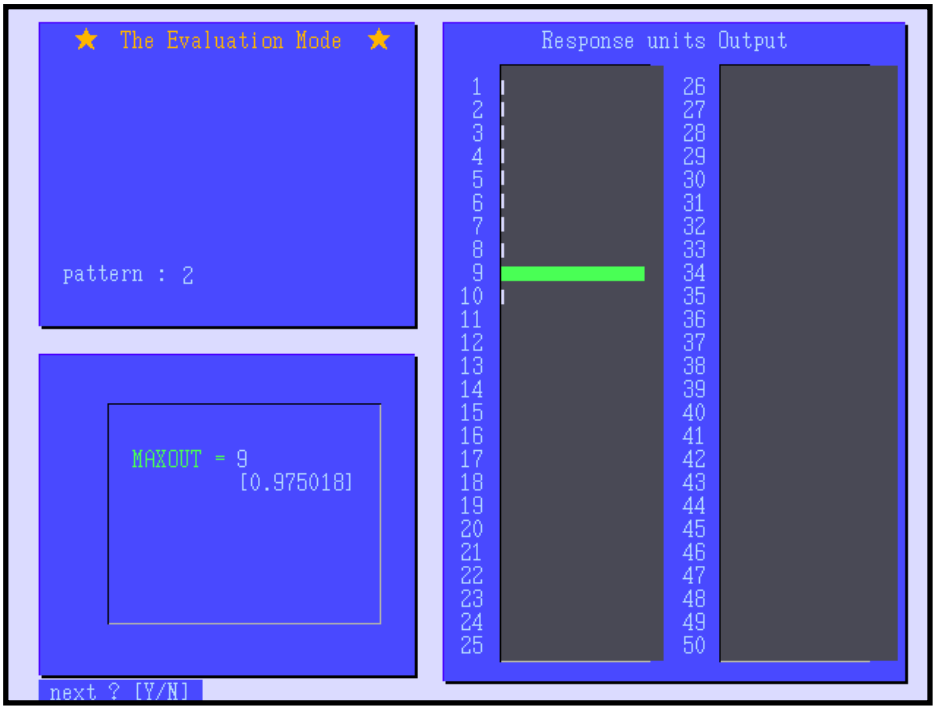


Fig.4.7 Interface of the evaluation mode

Chapter 5
EXPERIMEMTS AND ANALYSIS

In this chapter, many simulations and experiments will be introduced. There are two destinations of these simulations and experiments including: one is to verity the efficiency of the proposed network and algorithms, and further prove the proposed paper currency recognition system is effective not only on rejecting unknown currency patterns but also on guaranteeing the recognition capabilities for known currency patterns. On the other hand, it should be found the suitable parameters for this system having the good performances. In the experiments having been done, there are three countries' paper currencies being used mainly, including US dollars, which involve 8 par values (24 classes), Thai banknotes (10 classes) and Renminbi (Chinese paper currency, 10 classes). In each class there are 100 patterns (or pieces) of currency, respectively. In most of the experiments, Renminbi is used as known patterns for the system, which means the network is trained by 10 classes of Renminbi, then the US

dollars and Thai banknotes are the unknown patterns for this recognition system. In addition, other 16 counties' paper currencies including 252 patterns are also used as unknown currencies to evaluate the proposed recognition system. The constitution of the network follows the rules mentioned in last chapter.

Just as mentioned in chapter 2, a feature area should be selected for this mask set. The selection of this feature area should be in accordant with that its size will be smaller than that of the smallest paper currency. Furthermore, this feature area cannot be positioned adjacently to edges of paper currencies, because in practice the edge of the currency is easy to be damaged. It is possible to gather incorrect image features including some unwanted noises in this situation. In experiments, on the base of the mentioned rules, a feature area being composed of 50 blocks is determined. Although a mask pattern included in the mask set can be got by combining the blocks with different positions and different numbers as mentioned in section 2.4, there is only one block being included in each pattern in this research to verify universalities of the proposed network. All of these blocks are therefore utilized sequentially as the mask patterns in this mask set. Since the mask set with 50 mask patterns is used to extract 50 slab values for each piece of paper currency, there are 50 input units in input layer of the network. Some conditions of training the network is shown as in Table 5.1

Table 5.1 Conditions of the network learning

Number of input units	50
Number of hidden units	30
Number of output units	Decided by training samples
Convergence Criterion	Average of square error<0.0001
Maximum Iteration No.	20000

In this table the convergence criterion is $E(\cdot)/M < 0.0001$, in which $E(\cdot)$ is accordant with Eq.(3.8), M is the number of training samples in the training set. For the maximum Iteration No., it means that the iteration has to be stopped constrainedly if it cannot converge until to the 20000th step. The number of output units is equal to that of classes included in the training set. For example, if using 10 classes of paper currency to train the network, there are corresponding 10 output units in output layer, each of which is the representative of corresponding classes.

In practical paper currency systems, it has no any meaning to discuss improving

rejection capabilities for unknown currencies if the recognition capabilities for known currencies cannot be guaranteed. Therefore, before discussing the contents of improving rejection capabilities, we will address on the recognition capabilities of the proposed system for known currency patterns.

5.1 Recognition capabilities of the system

In this section, the recognition capabilities of the proposed recognition system are verified firstly, and comparative experiments with the conventional system are also executed. First, being as known currencies 10 classes of Chinese currency are used to evaluate the system. The results are given in Table.5.2, and in which the recognition results of the conventional system employing sigmoid function with the same restrictions are also shown. The same restrictions mean that the network structure, the input data of each samples, and the evaluations are identical. In this section, biases and widths of the proposed Gaussian function as in Eq.(3.5)are all constants, the values of biases and width are 0 and 0.54 respectively. The weights are updated according to Eq.(3.30).

Table.5.2 Recognition capabilities for Chinese currencies

Recognition Patterns	10 pieces of learning sample		20 pieces of learning sample	
	Sigmoidal	Gaussian	Sigmoidal	Gaussian
RMB100H	99%	99%	100%	100%
RMB100T	100%	98%	100%	100%
RMB50H	100%	100%	100%	100%
RMB50T	99%	97%	100%	100%
RMB20H	100%	100%	100%	100%
RMB20T	98%	99%	100%	100%
RMB10H	98%	98%	100%	100%
RMB10T	100%	100%	100%	100%
RMB5H	100%	99%	100%	100%
RMB5T	100%	100%	100%	100%
Average	99.4%	99%	100%	100%

It can be seen from this table that the average recognition capability of the proposed system is less 0.4 percentages than that of the conventional network as 10 pieces of Chinese currency of each class are used as the training samples. Moreover, as the Gaussian is used, the recognition ratios of all classes except for RMB20T are less than or equal to the corresponding ratios with sigmoid function. It reveals that the recognition capability of the proposed system is appreciably less than that of the sigmoid function in certain conditions, but it can be fetched up by increasing training samples. As the training samples in each class are 20 pieces, the recognition ratios of two methods are all 100 percent. To further verify its recognition capabilities, 8 classes of US dollar are also used to evaluate this system, the results are shown in Table 5.3.

Table 5.3 Recognition capabilities for US dollars

	Training samples	Evaluation data	Training samples	Evaluation data
	30	70	40	60
US20a	100%		100%	
US20b	98.57%		100%	
US20c	100%		100%	
US20d	100%		100%	
US50a	98.57%		98.33%	
US50b	98.57%		100%	
US50c	98.57%		100%	
US50d	97.14%		98.33%	
Average	98.93%		99.58%	

In this table the average rejection ratio for 8 classes of the US dollar is increased from 98.93% to 99.58% as the training samples in each class are increased from 30 to 40. The total number of misrecognized currencies is decreased from 6 to 2. It also can be concluded that the recognition capability of the system is increased as the increasing of the training samples. After accomplish the experiment of recognition capabilities for Chinese currencies and US dollars, it reveals that the recognition capabilities of the proposed system is not inferior to that of the conventional system, although its parameters are not optimized completely. It is an essential prerequisite of discussing the improvement of rejection capabilities of the system for unknown currencies in next sections. In following sections the recognition capability of the system will still be referred because of different leaning algorithms being used.

5.2 Influences of the network parameters

In this section the influences of the width and the bias of the proposed Gaussian function and positions of mask sets to rejection capabilities and convergence procedures are discussed respectively. One of these two parameters will be a constant as the other is changed for discussing. In the procedure of preprocessing, the sizes and the initial positions of all masks are same, and the mask set is same too.

5.2.1 Influences of the widths

In experiments of this subsection, in order to qualitatively study the influences of the width parameter to rejection capabilities and convergence procedures, all biases of hidden units and out put units are fixed to a constant 0. Then the width parameters of each unit in hidden layer and output layer are the same and are modified in a certain region manually. The average rejection ratios for 24 patterns of US dollar with the different widths (Sigma is 0.535, 0.54, and 0.70, respectively.) are illustrated in Fig.5.1. Moreover, the corresponding rejection capabilities for 10 patterns of Thai banknote are all listed in Table.5.4. Simultaneously, the rejection ratio of the network with sigmoid

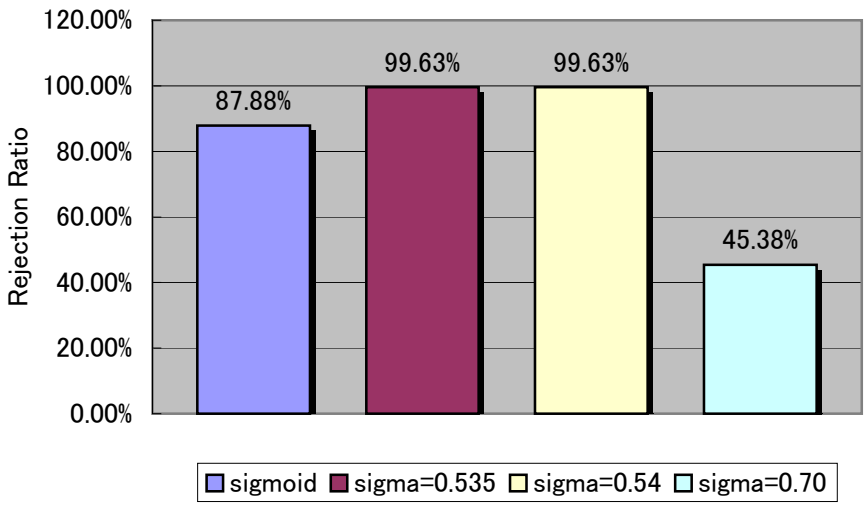


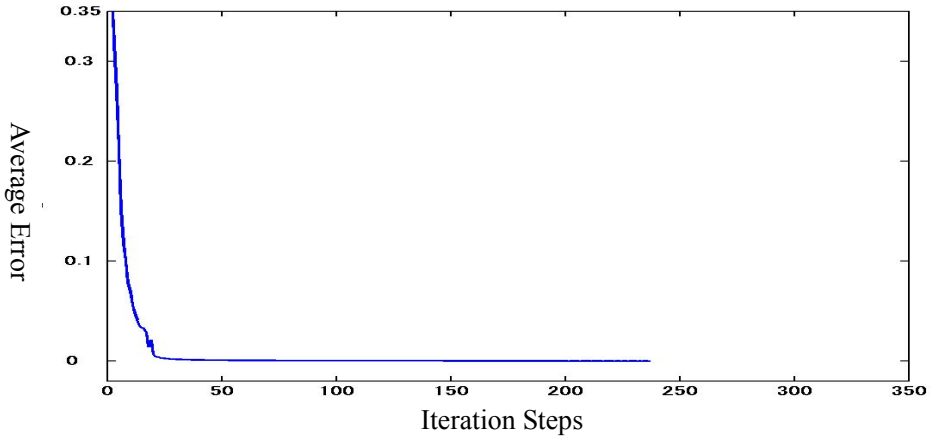
Fig.5.1 Rejection ratios for US dollars

function is combined being as a comparison with the Proposed Gaussian in the figure and the table. All the restrictions are same with that of Gaussian. As can be seen from Fig.5.1, the rejection ratios of the proposed network with Gaussian are 99.63% as the width parameters are 0.535 and 0.54 respectively. Although width parameters are not optimized with certain optimizing algorithms, the rejection ratios of the network employing Gaussian are still greater about 12 percentages than that of the network with sigmoid activation function, which is 87.88%. However, as the widths are 0.7, the rejection ratio decreases sharply, it is only about a half of that as sigma is 0.54. The similar situations are also can be found from the change of rejection ratios for Thai banknotes shown in table5.4. But it cannot shadow the potentials of the proposed system on aspect of improving rejection capabilities for unknown currency patterns. From Table.5.4 it can be found that the rejection ratios for all classes of Thai banknote, except for BG500T, are increased as the Gaussian, in which widths are 0.535 or 0.54, is employed. All of these just reveal the proposed system can possess good performances with the appropriate width parameters.

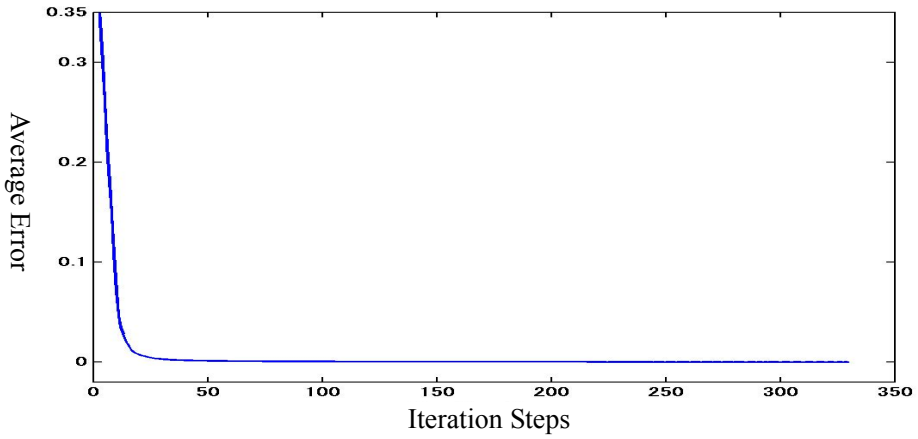
Table.5.4 Conditions of the network learning

	Sigmoid	**Gaussian**		
		σ =0.535	σ =0.54	σ =0.70
BG1000H	75%	91%	87%	49%
BG1000T	100%	97%	100%	90%
BG500H	98%	100%	100%	15%
BG500T	100%	83%	82%	52%
BG100H	82%	98%	96%	60%
BG100T	46%	100%	100%	50%
BG50H	100%	100%	100%	54%
BG50T	97%	99%	99%	12%
BG20H	97%	100%	100%	100%
BG20T	62%	100%	100%	86%
Average	85.7%	96.8%	96.4%	56.8%

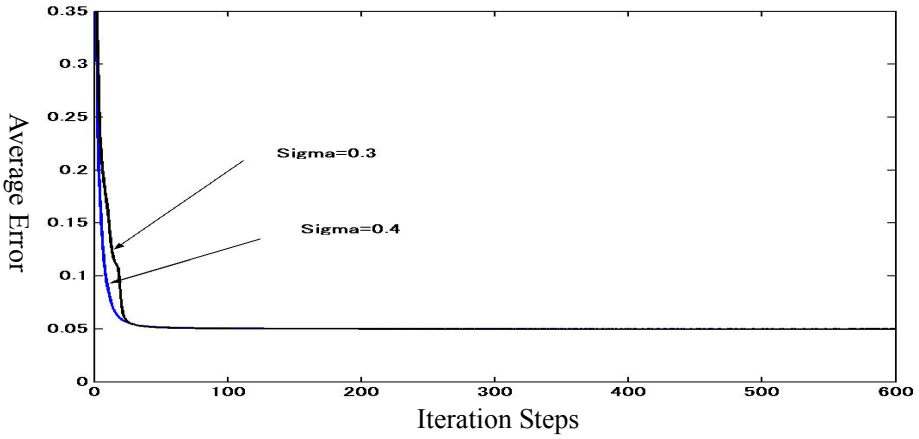
From the table and the figure it also can be found that as the sigma is changed from 0.535 to 0.54, the average rejection ratio for Thai banknotes is decreased 0.4 percentages. Furthermore, as sigma is 0.70, the average rejection ratios for US dollars and Thai banknotes decrease about a half. This phenomenon reveals the rejection ratio is descendent along with the increments of widths in a certain range. Now let us try to explain why the increment of sigma leads to descent of corresponding rejection ratios from the characteristics of the employed Gaussian activation function and the network.



(a) Sigma=0.70



(b). Sigma=0.535



(c). Sigma=0.3 and 0.4

Fig.5.2 Error curves with different widths

Just as mentioned in chapter 3, the proposed activation function is a ridge-like function and it produces many closed bump-like regions. The width parameters control not only active ranges of these ridges but also the size of these closed regions. If sigmas are increased, which means all the widths of ridges and the size of those closed regions are larger than before, some of the customarily rejected data being out of the active ranges of those ridges and closed regions will be re-delimited into the active ranges. They will be misrecognized as known patterns. Furthermore, if the width is too small, some training samples cannot be included in its active ranges, it is possible to lead to the network cannot converge to be accordant with the criterion. The convergence procedures illustrated with the error curves in Fig.5.2(c) just prove this hypothesis from another side. It shows that as the sigma is 0.3 and 0.4 respectively, the iterative searches are stuck in a bad local minimum and cannot escape form it. Comparing the error curves in Fig.5.2 (a) and (b), you can discover that the iteration procedure of widths are 0.7 is shorter than that of widths are 0.54. The first one is 227 steps and the other is 331. It can be explained as following. In this case, the biases and widths are constants. It means the positions and the shapes of those ridges produced by the proposed function are fixed, the weight vector, which controls the directions of the ridges, is unique vector need to be regulated. Because of being with the same training samples, if the widths are larger, it is easier to determine the directions of the ridges. The iteration procedure is therefore faster.

It is concluded form the contents mentioned above that in a certain range the proposed Gaussian activation function with lager width can possibly lead to not only the descent of rejection capabilities for unknown currency patterns, but also the cut of the iteration procedure.

5.2.2 Influences of the biases

In this section, influences of the biases of the proposed network to rejection capabilities for unknown currency patterns will be analyzed qualitatively. As mentioned in chapter 3, the positions of ridge-like functions in the multi-dimensional space are determined with these biases. On the base of training samples, being with the ridge-like functions having appropriate positions, it is possible to get accurate interfaces for each class of the training samples easier. On the contrary, if the positions of the ridges are not appropriate, the demands for training weights and widths will be more serious. Although the trained network can recognize known currency patterns effectively, the

possibility of that it cannot reject unknown currency patterns correctly is still higher. Because of inappropriate positions of the ridges, the interfaces, which are decided with directions and shapes by weights and biases, cannot be guaranteed to compactly encircle each classes of known currency pattern. The unknown patterns are not included in the training set, so we do not know their distributions clearly. If some data locates inside the interfaces mentioned above, it will be misrecognized as a certain known pattern. Just because of these incompact encircling caused by inappropriate biases, the probabilities of unknown patterns being inside the interfaces are apparently increased. It is just the reason leading to bad rejection capabilities for unknown currency patterns in this case.

Table.5.5 Rejection ratios with different biases for Thai banknotes

Bias	0	0.25	0.5	0.75	1.0	1.25	1.5	1.75	2.0	2.25
BG1000H	41%	66%	51%	78%	91%	94%	82%	45%	45%	50%
BG1000T	99%	100%	99%	99%	100%	100%	100%	100%	100%	64%
BG500H	14%	57%	78%	49%	50%	56%	80%	75%	81%	50%
BG500T	50%	78%	100%	100%	86%	99%	95%	100%	100%	83%
BG100H	48%	51%	64%	100%	100%	98%	99%	69%	87%	96%
BG100T	77%	95%	95%	72%	80%	92%	54%	84%	79%	74%
BG50H	47%	60%	100%	70%	64%	51%	59%	50%	51%	51%
BG50T	63%	100%	100%	99%	97%	100%	100%	95%	97%	100%
BG20H	100%	100%	100%	100%	97%	97%	100%	100%	68%	97%
BG20T	49%	100%	78%	100%	100%	75%	100%	97%	100%	99%
Average	58.8%	80.7%	86.5%	86.7%	86.5%	86.2%	86.9%	81.5%	80.8%	76.4%

In the experiments of this section, 10 classes of Chinese currency are used as training samples to train the proposed network, in which the widths of all hidden units and output units are 0.50. Being as unknown currencies, US dollars and Thai banknotes (3400 pieces totally) are evaluated with the trained network. The corresponding rejection capabilities with different biases are given in Table.5.5. It can be found for each class of Thai banknote the largest rejection ratio is 100%, which means all patterns involved in that class are rejected correctly and successfully; the smallest rejection ratio is only 14%, which presents 86 pieces in that class are misrecognized as some kinds of the Chinese currency. By surveying the changing tendency of each class of Thai banknote along with linear increasing of biases from 0.0 to 2.25 in Fig.5.3, the changes them are desultory and it is difficult to discover any regular rules. It seems there is no rules for it.

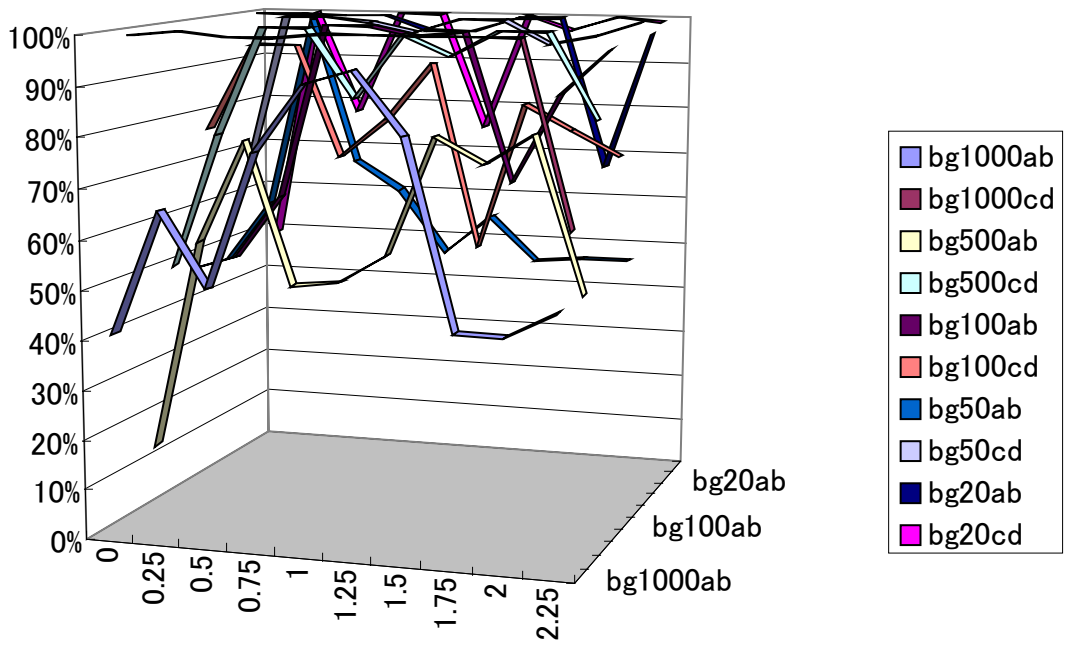


Fig5.3 Rejection ratios for each of Thai banknote classes

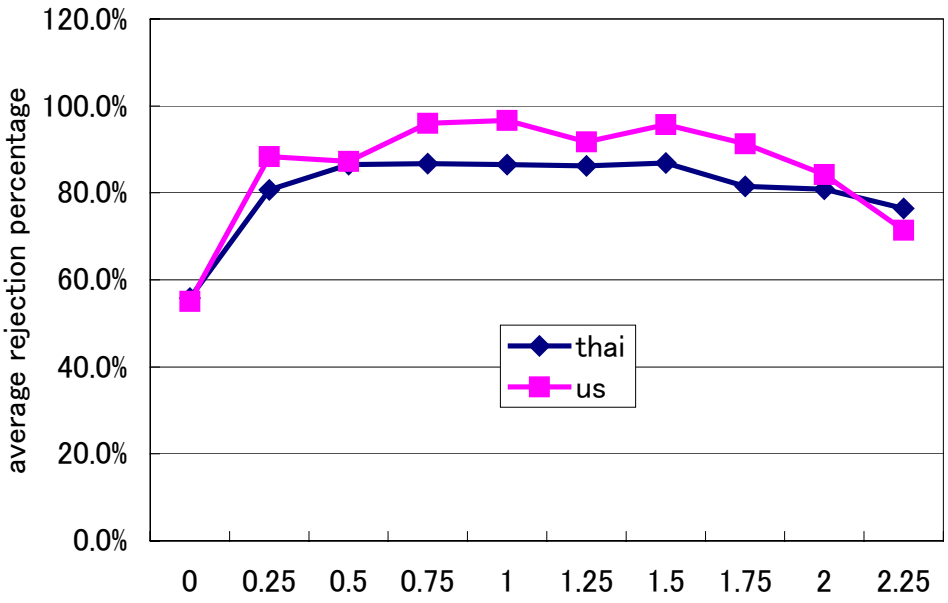


Fig.5.4 Average rejection ratios for US dollars and Thai banknotes

However, as we observe the lines in Fig.5.4, which shows the average rejection ratios for all classes of US dollar and Thai banknote, it can be found that in the middle part of the changing range of the biases the average rejection capabilities of the system are larger, and are smaller in both sides. The shapes of these two lines are similar with that of the normal distribution. This figure also reveals that in this case the rejection ratios of the system for US dollars are greater than that for Thai banknotes. These results are also a reference for determining the initial range of the biases in automatically regulating them with optimizing algorithms. In the following part, the relative contents will be discussed.

Table.5.6 shows the rejection capabilities of the proposed network for Thai banknotes with different initial values of biases. In this case the width parameter is 0.58, and the biases are updated using the algorithm shown in Eq.(3.31). It can be seen from this table as the range of initial values of biases is (0, 1.5), the average rejection ratio is the best to 99.3%, there are only 7 pieces of Thai banknote being misrecognized as Chinese currencies. But as the range of initial values is (0, 0.5), the average rejection ratio is only 54.9%. As the initial range is enlarged to (0, 3.0), the rejection ratio is also descending about 5 percentages. It presents that it is beneficial for improving rejection capabilities of the system by enlarging the initial range of biases properly. This initial

Table.5.6 Rejection ratios for Thai banknotes with different initial biases

	(0, 0.5)	(0, 1.0)	(0, 1.5)	(0, 2.0)	(0, 3.0)
BG1000H	78%	60%	99%	93%	100%
BG1000T	24%	100%	95%	99%	97%
BG500H	49%	62%	100%	74%	100%
BG500T	30%	93%	100%	99%	100%
BG100H	51%	96%	100%	100%	99%
BG100T	56%	100%	100%	100%	67%
BG50H	65%	99%	100%	100%	94%
BG50T	42%	99%	99%	98%	93%
BG20H	99%	100%	100%	99%	98%
BG20T	55%	100%	100%	100%	96%
Average	54.9%	90.9%	99.3%	96.2%	94.4%

range also cannot be so large that it influences the convergence of the network certainly. Fig.5.5 shows the bias distributions after training as the initial range is (0, 1.5) and (0, 3.0) respectively, in which asterisks denote those of initial range is (0, 3.0), and circles denote those of initial range is (0, 1.5). It can be found from this figure there is not any evidences to present that biases have a certain tendency to converge into a special region, in which the biases can make the system possess considerable rejection

capabilities for unknown currency patterns. On the contrary, most of biases are still staying inside the respective initial ranges. It can be explained from two aspects. One of them is that the optimizing algorithm employed here is bottom on local gradient methods whose searching range is limited. It is difficult for it to explore more distantly. The other is that the biases are not very sensitive for convergence procedures.

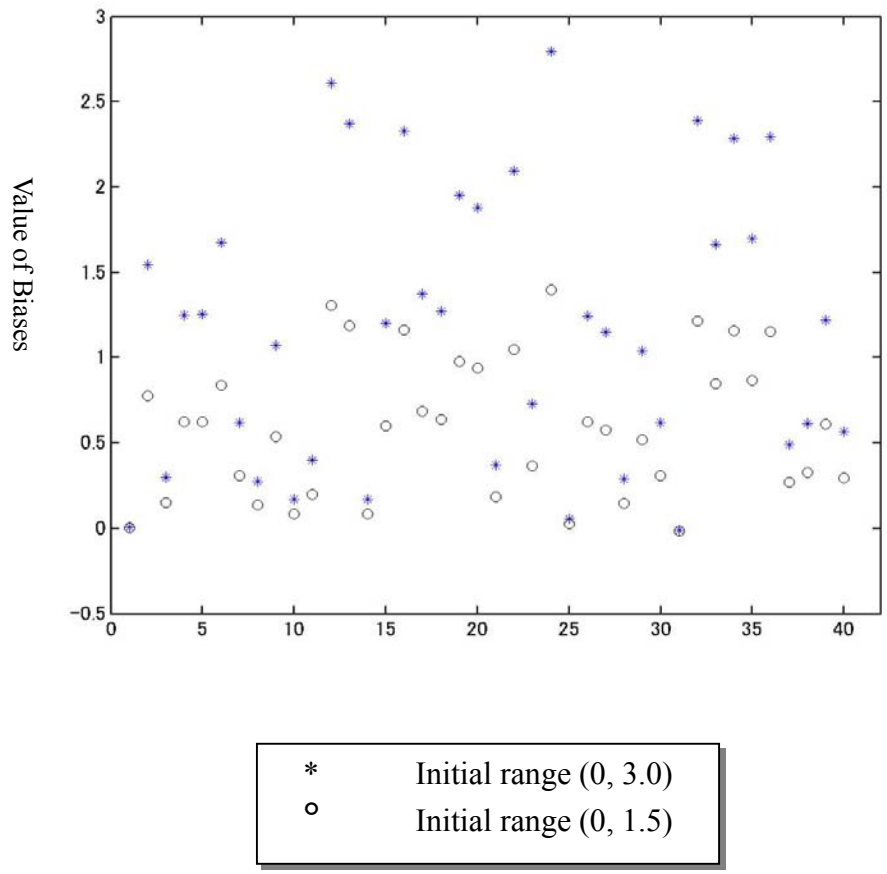
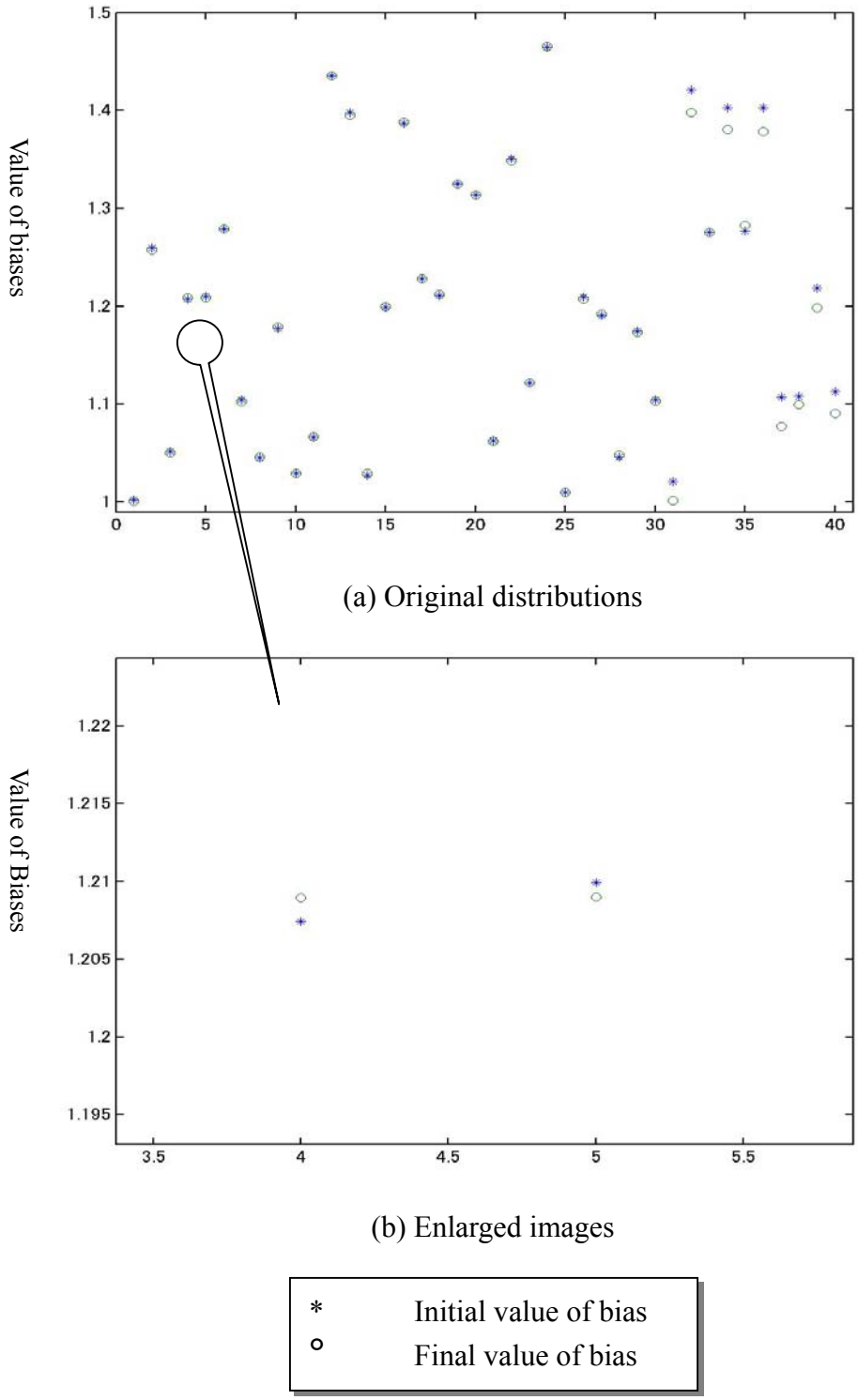


Fig.5.5 final distributions of biases with different initial ranges

Furthermore, the initial distribution of the biases in the range (1, 1.5) and its final distributions after training are illustrated in Fig.5.6, in which (a) is the original image and (b) is the local enlarged image. Asterisks denote initial values of biases, and circles denote the final values of biases. From it you can see that after learning most of the biases are still staying nearby its initial points, the differences between them are not very much, the magnitude of some of them is only about 10^{-3}.



(a) Original distributions

(b) Enlarged images

*	Initial value of bias
o	Final value of bias

Fig.5.6 Distributions of biases before and after training

5.2.3 Influences of mask sets

The influences of mask sets applied in preprocessing to the rejection capabilities of the network for unknown currency patterns will be involved in this section. The mask set mentioned here is used to produce the slab values representing digital features of currency patterns. It is used to process not only training samples, but also all currencies evaluated by the network including known and unknown patterns. Table.5.7 and 5.8 respectively show rejection ratios for US dollars and Thai banknotes with different initial positions of mask sets.

Table 5.7 Rejection ratios for US dollars with different mask sets

Initial positions of mask sets	-59, -6		-60, -6	
Sigma	0.535	0.54	0.535	0.54
US100a	100%	100%	100%	100%
US100b	100%	100%	100%	100%
US100c	97%	97%	83%	87%
US100d	100%	100%	100%	100%
US50a	100%	100%	100%	100%
US50b	100%	100%	100%	100%
US50c	100%	100%	100%	100%
US50d	100%	100%	100%	100%
US20a	100%	100%	100%	100%
US20b	100%	100%	89%	95%
US20c	100%	100%	100%	100%
US20d	99%	99%	98%	99%
US10a	100%	100%	100%	100%
US10b	100%	100%	100%	100%
US10c	99%	99%	84%	88%
US10d	100%	100%	98%	98%
US5a	100%	100%	100%	100%
US5b	98%	98%	96%	98%
US5c	100%	100%	100%	100%
US5d	100%	100%	100%	100%
US1a	99%	99%	91%	94%
US1b	100%	100%	100%	100%
US1c	100%	100%	100%	100%
US1d	99%	99%	99%	99%
Average	**99.63%**	**99.63%**	**97.42%**	**98.25%**

If the initial positions are different, the position of each mask pattern included in this mask set is then different, so it is possible to lead to corresponding slab value is different, and lead to different input vectors of the network for the same paper currency. If the input vector of training samples is different, the error surfaces of the network and the initial points of training on these surfaces are not same. Because of complexities of the error surface just as mentioned in chapter 3, it is too difficult to converge to a same minimum with good performance. The analysises mentioned above can be presented from the results of Table.5.7 and 5.8.

Table.5.8 Rejection ratios for Thai banknotes with different mask sets

Initial positions of mask sets	-59, -6		-60, -6	
Sigma	0.535	0.54	0.535	0.54
BG1000H	91%	87%	100%	99%
BG1000T	97%	100%	99%	99%
BG500H	100%	100%	98%	99%
BG500T	83%	82%	100%	98%
BG100H	98%	96%	100%	100%
BG100T	100%	100%	100%	100%
BG50H	100%	100%	100%	100%
BG50T	99%	99%	100%	99%
BG20H	100%	100%	100%	100%
BG20T	100%	100%	100%	100%
Average	**96.8%**	**96.4%**	**99.7%**	**99.4%**

As can be seen from Table.5.7, if the widths are 0.535, the average rejection ratio for US dollars decreases about 2 percentages as the initial position of the mask set is moved from (-59,-6) to (-60, -6), and this kind of descent is about 1.5 percentages as the widths are 0.54. But this kind of difference of initial positions leads to that the corresponding rejection capabilities for Thai banknotes increase about 3 percentages. Especially, this movement of the initial position results in that the changes of the rejection ratios for US100c and BG500T are all greater than 10 percentages. It is dangerous in practical applications. It is essential to carefully determine the positions of mask sets. Furthermore, even if with the same initial positions, the mask patterns in the mask set are possibly different. There so many selections of positions, size and combination of masks, all of these are factors influencing performance of the network. Moreover, the two thresholds Th1and Th2 mentioned in evaluation procedure will also influences the

performance of the proposed system. The contents of these parts are not involved here no longer. In the following parts some contents about optimizing algorithms will be introduced in detail.

5.3 Analysis for optimizing algorithms

In order to discover effecive algorithms of regulating width parameters aumatically to get good perfomances of the system, many experiments of algorithms have been done. In this section, we will observe influences with the differnet algorithms for system performances, especially for rejection capabilities for unknown currency patterns, and analyze the reasons leading to thses results and influences.

Table.5. 9 Rejection ratios for Thai banknotes with different initial width ranges

Sigma	(0.4, 0.7)	(0.5, 0.6)	(0.5, 0.8)	(07, 0.8)
BG1000H	100%	99%	100%	100%
BG1000T	100%	100%	97%	100%
BG500H	89%	100%	76%	100%
BG500T	75%	93%	100%	100%
BG100H	100%	100%	100%	100%
BG100T	100%	100%	94%	100%
BG50H	96%	100%	100%	100%
BG50T	100%	99%	100%	85%
BG20H	100%	100%	99%	100%
BG20T	100%	100%	100%	100%
Average	**96.0%**	**99.1%**	**96.6%**	**98.5%**

5.3.1 Sequential gradient method for BP algorithm

It has mentioned in section 3.4, this approach is a stochastic algorithm, in which parameters are updated with every training sample. Therefore, the total error function is not always descendent in each step of iterations during the procedure of exploring

optimal solutions. It has possible to escape from local minima. In this research, this method is always used to optimize weights and biases. Here it also used to explore optimal widths firstly. The experiments of widths with different initial ranges, which are (0.4, 0.7), (0.5, 0.6), (0.5, 0.8), and (0.7, 0.8) respectively, are executed. The rejection ratios for Thai banknotes and US dollars are shown in Table.5.9 and 5.10, correspondingly. Furthermore, distributions of the widths before and after training are also illustrated in Fig.5.8 to Fig.5.11. The error curves presenting the iteration procedures of these four situations are given in Fig.5.12. The analysis of them will be presented in the following parts.

Table.5.10 Rejection ratios for US dollars with different initial width ranges

	(0.4, 0.7)	(0.5, 0.6)	(0.5, 0.8)	(07, 0.8)
us1a	100%	98%	97%	100%
us1b	100%	100%	100%	100%
us1c	100%	100%	99%	99%
us1d	98%	99%	99%	100%
us5a	100%	100%	100%	100%
us5b	100%	98%	99%	96%
us5c	100%	100%	100%	100%
us5d	100%	100%	100%	100%
us10a	100%	100%	100%	100%
us10b	100%	100%	100%	100%
us10c	97%	99%	100%	100%
us10d	100%	100%	100%	100%
us20a	100%	100%	100%	100%
us20b	100%	100%	100%	100%
us20c	100%	100%	100%	100%
us20d	91%	99%	99%	97%
us50a	100%	100%	100%	100%
us50b	100%	99%	100%	99%
us50c	100%	100%	100%	100%
us50d	100%	100%	100%	100%
us100a	100%	100%	100%	100%
us100b	100%	100%	100%	99%
us100c	97%	99%	98%	100%
us100d	99%	100%	100%	100%
Average	**99.25%**	**99.63%**	**99.63%**	**99.58%**

As can be seen from Table.5.9 and 5.10, the average rejection ratios for US dollars do not changed very much as the initial widths being changed in these four ranges. All of

them are greater than 99%, the differences between them are all less than 0.4 percentages. Even in the worst case with initial range (0.4, 0.7), there are only 18 pieces of US dollar in all 2400 pieces being misrecognized as some kinds of Chinese currencies. For Thai bankenotes, the best case is in initial range (0.5, 0.6), the average rejection ratio is 99.1%. there are only 9 pieces being misrecognized in all 1000 piecese. The worst case is still the initial range (0.4, 0.7), the average ratio is 96%, which seems not so bad. But the rejection ratios for the class BG500ab and BG500cd are only 89% and 75%. It reveals that the section of initial width ranges are still impertant for improving rejection capabilities of the proposed system even if the training procedures with different initial rangs can coverge smoothly. It because the error surface of the proposed network is too complex, there are many local minima that can be satisfied

with the convergence criterion even in a small region. If all structure parameters except for activation function are same, the error surface of the proposed network should be more complex than that of the network with sigmoid activation function. It results from that the proposed activation function is a ridge-like function, it has a more flexure than sigmoidal function. Therefore, it should be more careful in selecting the initial parameters of the network.

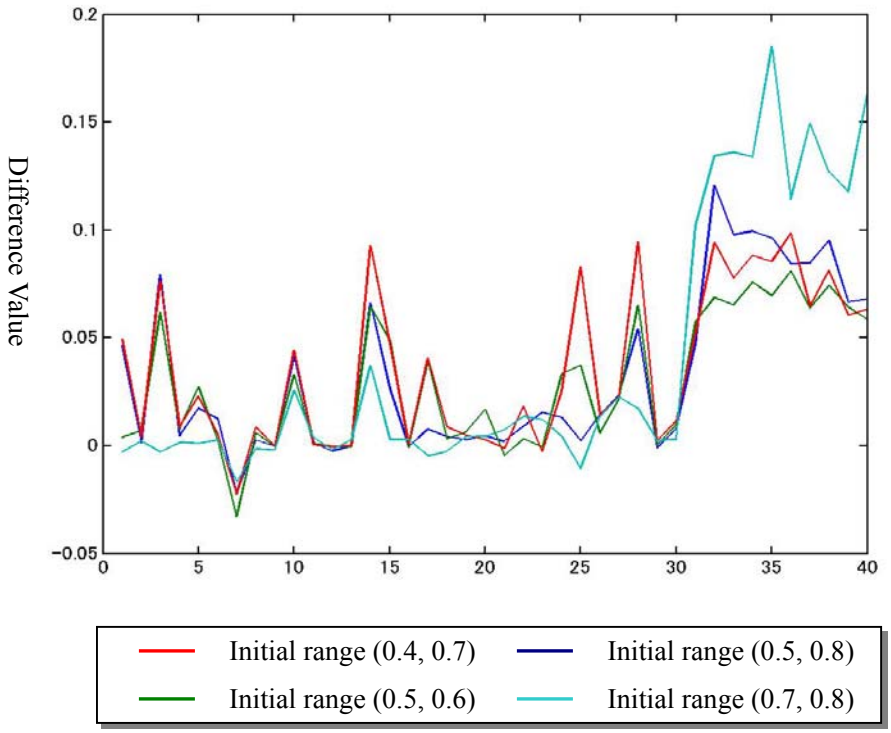


Fig.5.7 Differences of the widths before and after learning

From Fig.5.8 to Fig.5.11, they illustrate the distributions of the widths before and after learning with the initial range (0.4, 0.7), (0.5, 0.8), (0.5, 0.6), and (0.7, 0.8) respectively. In which the rectangles denote the initial values of widths, and the asterisks denote the final values of them after learning. You can see from these figures that most of the first 30 widths, which are the widths of hidden units, move down to smaller values after leaning, and several of them stay nearby their initial values or move towards upside a little. For all the last 10 widths, which are the widths of output units, the final values are move down after learning. The distances of these moving are larger. This phenomenon is presented much apparently as the intial range is (0.7, 0.8). All of these moving are illstrated quantitatively in Fig.5.7. as can be seen from it the largest differnece of before and after learning is near to 0.2 (decreasing). But the largest incrment after lerning is only about 0.04

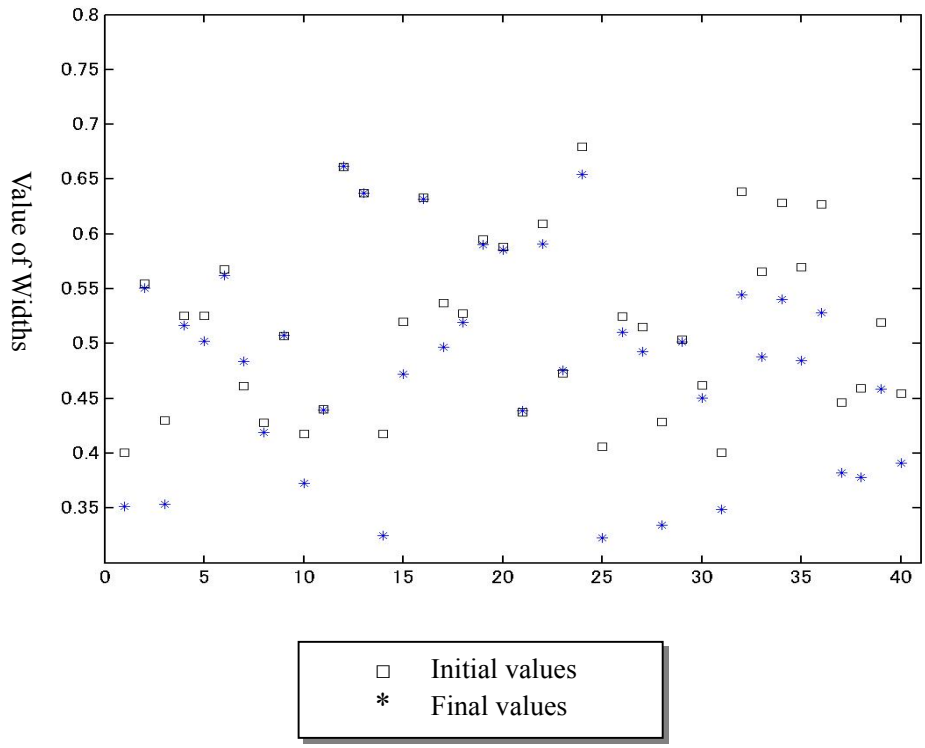


Fig.5.8 Distributions of the widths before and after learning with the initial range (0.4, 0.7)

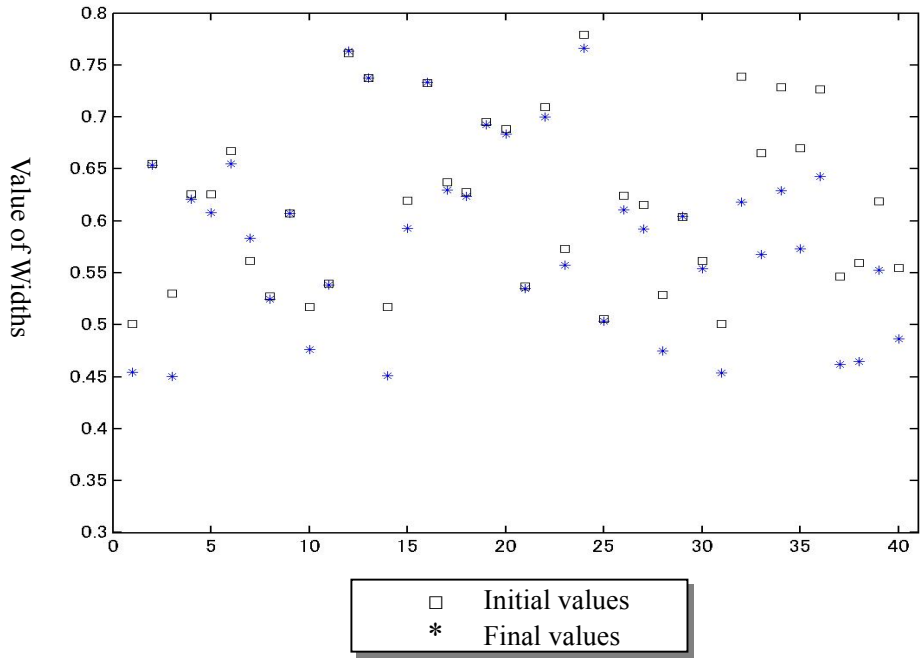


Fig.5.9 Distributions of the widths before and after learning with the initial range (0.5, 0.8)

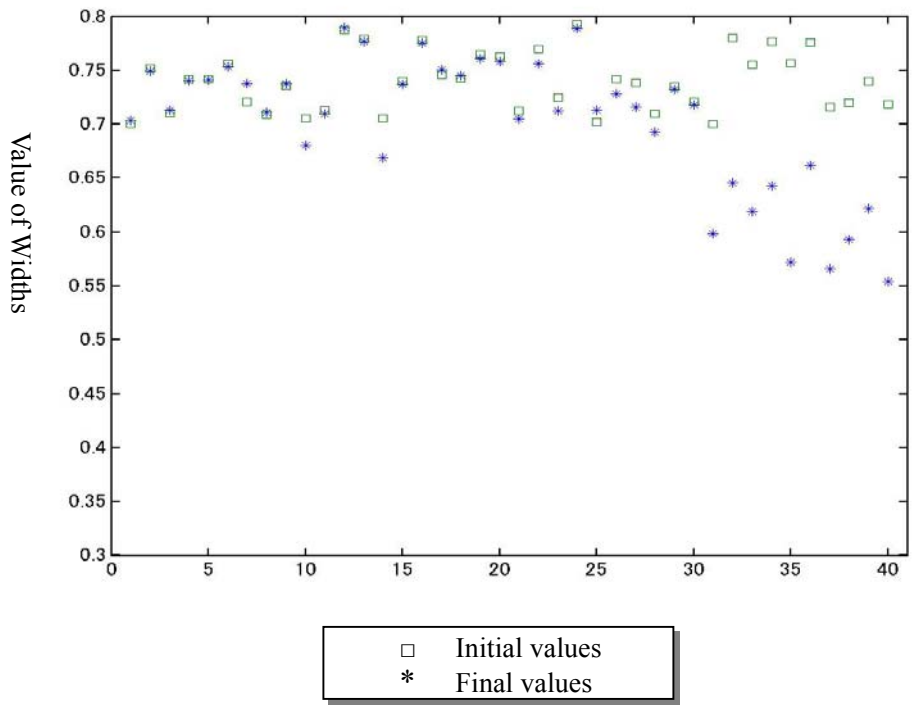


Fig.5.10 Distributions of the widths before and after learning with the initial range (0.7, 0.8)

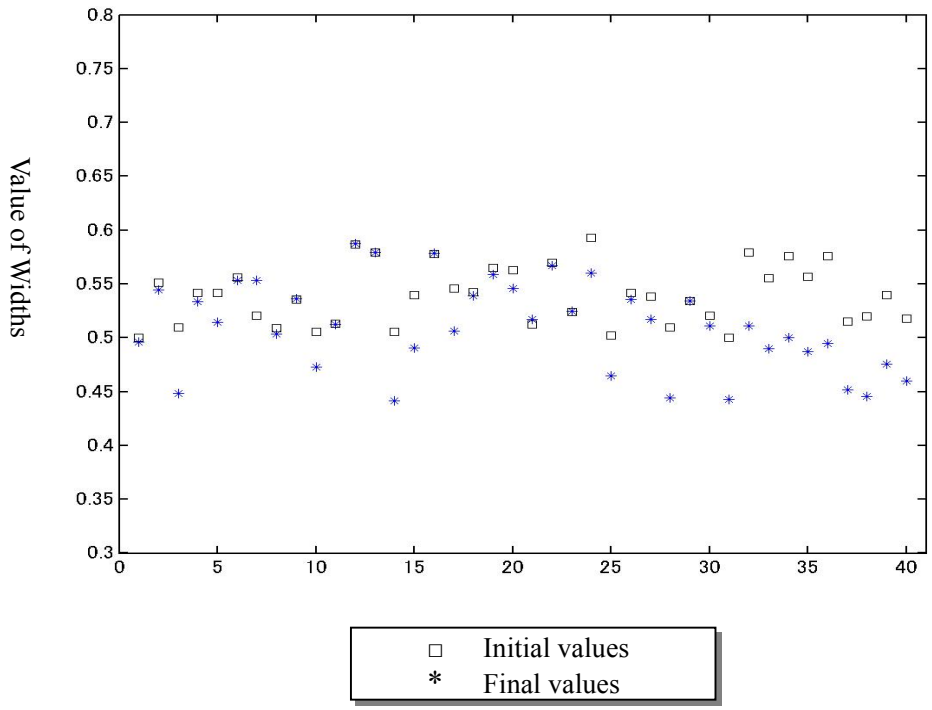
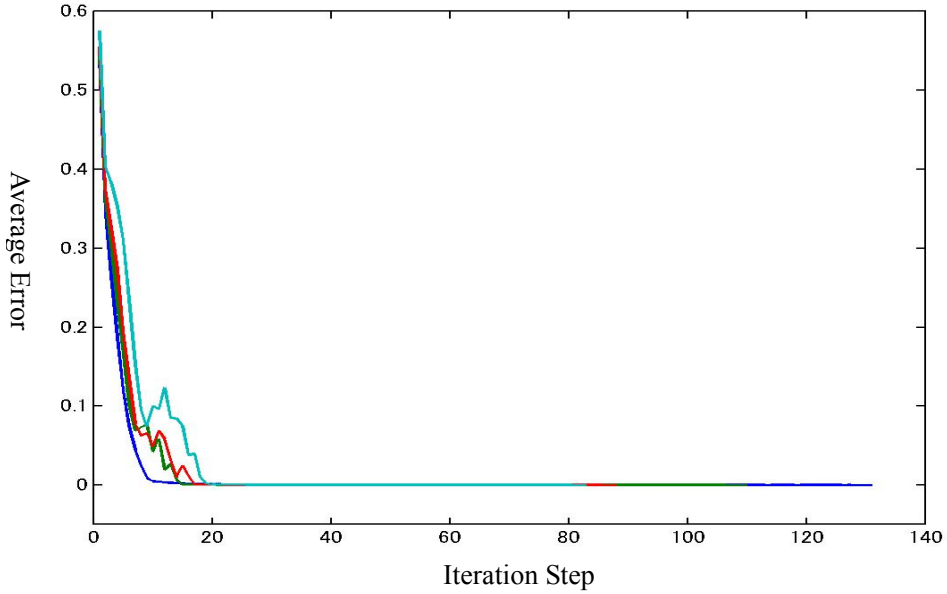


Fig.5.11 Distributions of the widths before and after learning with the initial range (0.5, 0.6)

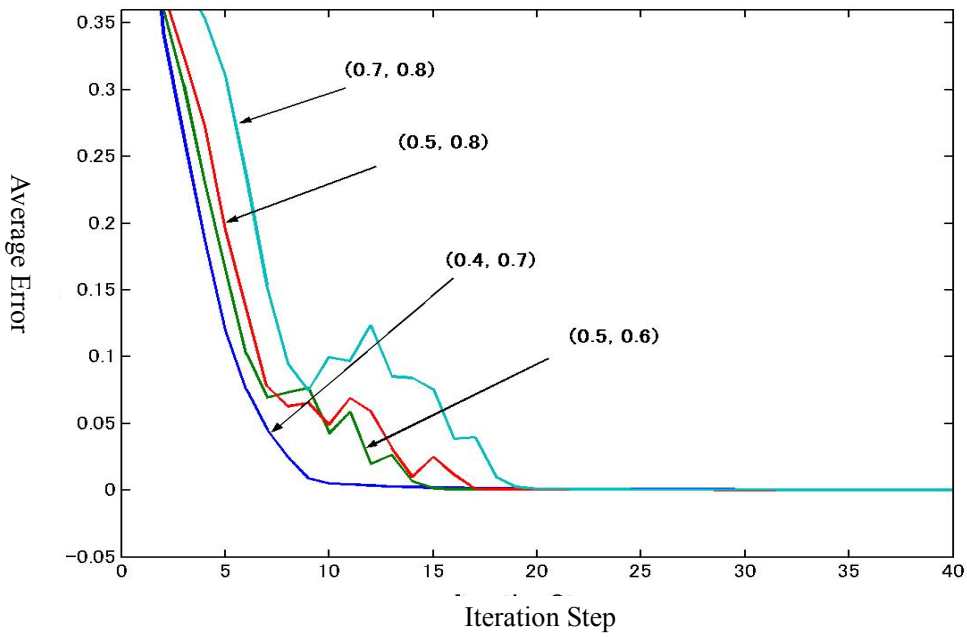
Fig.5.12 shows convergence procedures of the error function with differnt initial width ranges, in which (a) is the whole convergence procedures, and (b) shows the enlarged image of the convergence from the 1st step to the 40th step. The corresponding iteration steps are shown in Table.5.11, in which the one step means all training smaples included in the training set are updated one time. Because of using the sequential version of gradient descent, the practical iteration steps of the parameters in one step mention above are equal to the number of training samples included in the training set. iteration steps. For example, if there are 50 training samples in the training set, the practical iterations steps are 50 times of the steps shown in the following table.

Table.5.11 Itration steps with different initial width ranges.

Intial Range	(0.4, 0,7)	(0.5, 0.8)	(0.5,0.6)	(0.7, 0.8)
Itration Steps	131	110	88	83



(a) Original convergence procedures



(b) Local enlarged convergence procedures

Fig.5.12 Iteration procedures of different initial width values

It can be seen clearly from Fig.5.12 that except for the initial range (0.4, 0.7), the iteration procedures are not very smooth in a certain period for the other three initial width ranges. During this period, in some iteration steps values the average errors are not decreasing, but increasing, although the iterations of these four situations move along with negative direction of error function. However, this is just a an advantage of employing the sequntial gradient method. Because the gradient of each training samples included in the training set is calculated with respect to the error caused by themselves in the sequential version, the gradient inthis case is only a gradient component at that point on the error surface. The iteration moves along with the direction of this negative gradient can only guaratee decreacing the error caused by this training sample. For the other training samples, this moving direction probably leads to increment or descent of errors caused by themselves. The total error caused by the training set is therefore possible to be increased or decreased moving along with this direction. Just because of this property of the sequential gradient method, it is benefit for escaping the iteration search from local minima and searching optimal parameters in a lager region.

Table.5.12 Itration steps and rejection ratios with the different initial learning rate

Intial Range	Itration Steps	Average Rejection Capabilities	
		Thai Banknotes	US Dollars
(0.4, 0,7)	321	94.5%	98.92%
(0.5, 0.8)	117	94.7%	99.71%
(0.5,0.6)	107	98.9%	99.83%
(0.7, 0.8)	64	98.7%	99.86%

Then the influences of the learning rate for performances of the proposed network will be discussed in the following. The initial value of learning rates of widths in the completed experiments is 0.05. in order to observe the influences of the learning rate, its initial value is enlarged one time to 0.1. the contrastive experiments with the four initial width ranges are excuted. Table.5.12 shows the corresponding iteration steps and rejection capabilities for US dollars and Thai banknotes. Comparing it with Table.5.11, it can be found that the iteration steps with the range (0.4, 0,7) chang most obviously as enlarging the initial learning rate of widths. It takes 321 steps for the iteration to accord with the convergence criterion. Why does the iteration take so many steps? Let us look for answers from Fig.5.13, which shows the iteration procedures with the two different initial learning rates as the initial width range is (0.4, 0.7). it can be seen that the changing curves of the error for these two initial learning rate are almost superposed in

the first about 100 steps. It reveals that the error surface near the initial point is smooth and there are not big flxures which cannot be stridden with the two learning rates. Because of the step size with initial value 0.05 is so appropriate that the iteration is easier to converge to the minimum locating near the starting ponit. But as the intial learning rate is 0.1, the succesive iteration steps are oscillatory on the two sides of that minimum because of the lager step size, and it cannot converge to this minimum. Duirng the procedure of this oscillation the practical step size is regulating automatically as mentioned in section 3.4, as the iteration is executed to more than about 250 steps, it is beginning to climb up from this valley-bottom and then coverge to another minimum. The iteration steps in this case are therefore much more than that with the initail step size 0.05.

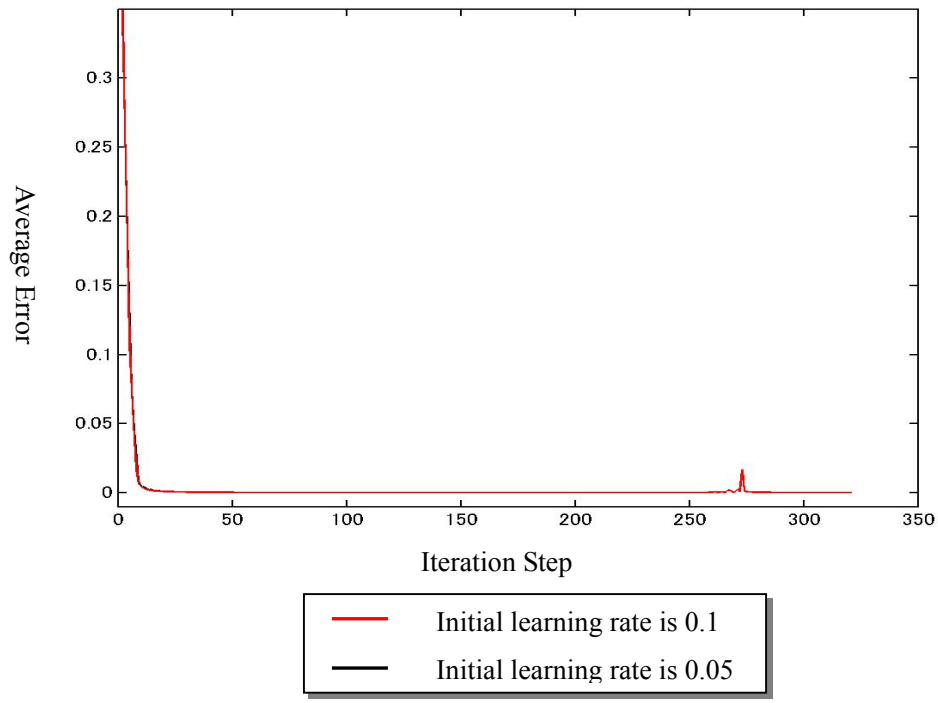


Fig.5.13 Error curves of different initial learning rates with initial width range (0.4, 0.7)

Fig.5.14 shows the changing procedure of the learning rate from initial value 0.1 with width range (0.4, 0.7). From it we can see that it is a oscillatory increasing procedure about during the 100th step and 200th step. It results from the iteration is oscillating near the minimum and the error is also oscillating again and again. It can prove the analysis mentioned above from another side.

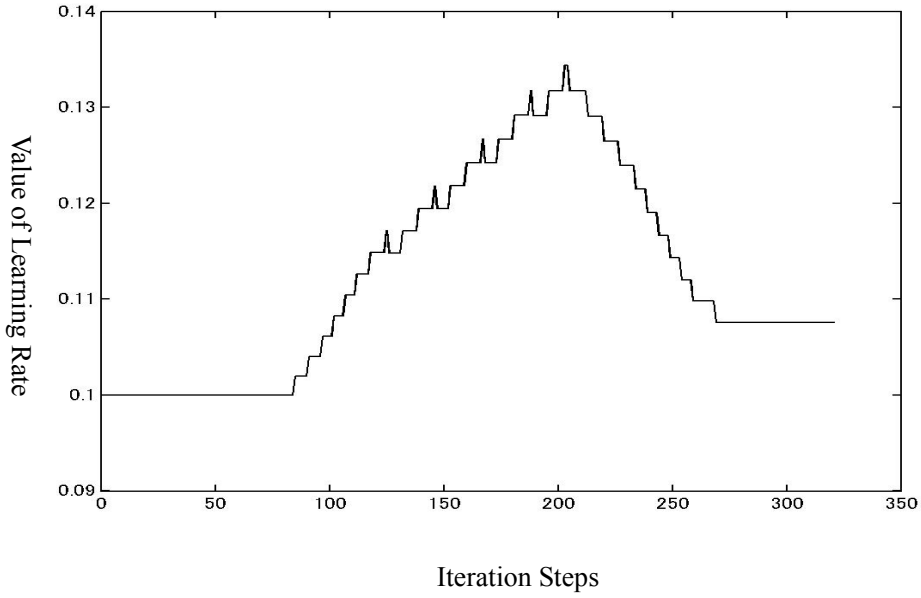


Fig.5.14 The changing procedure of the learning rate with width range (0.4, 0.7)

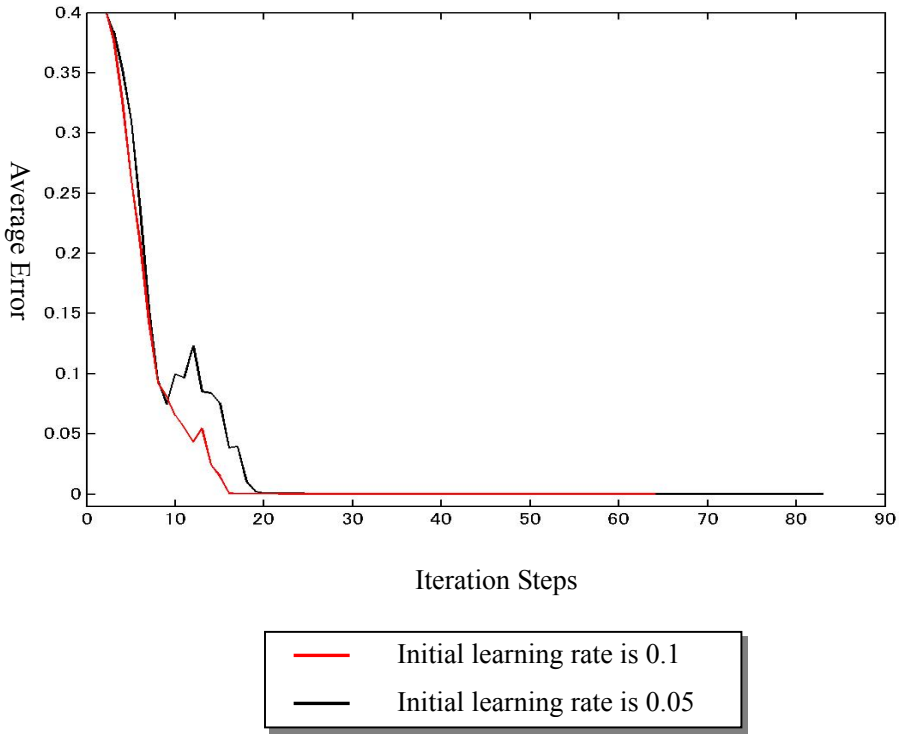


Fig.5.15 Error curves of different initial learning rates with initial width range (0.7, 0.8)

Let us consider the situation that the initial range is (0.7, 08) again. From Table 5.12, it can be found that as the initial learning rate is 0.1 the iteration steps are only 64, which is less about 20 steps than the initial learning rate is 0.05. The reason of this phenonenon can be get from analying Fig.5.15, in which the red line denots the situation that the initial learning rate is 0.1, the black one denotes the initial learning rate is 0.05.

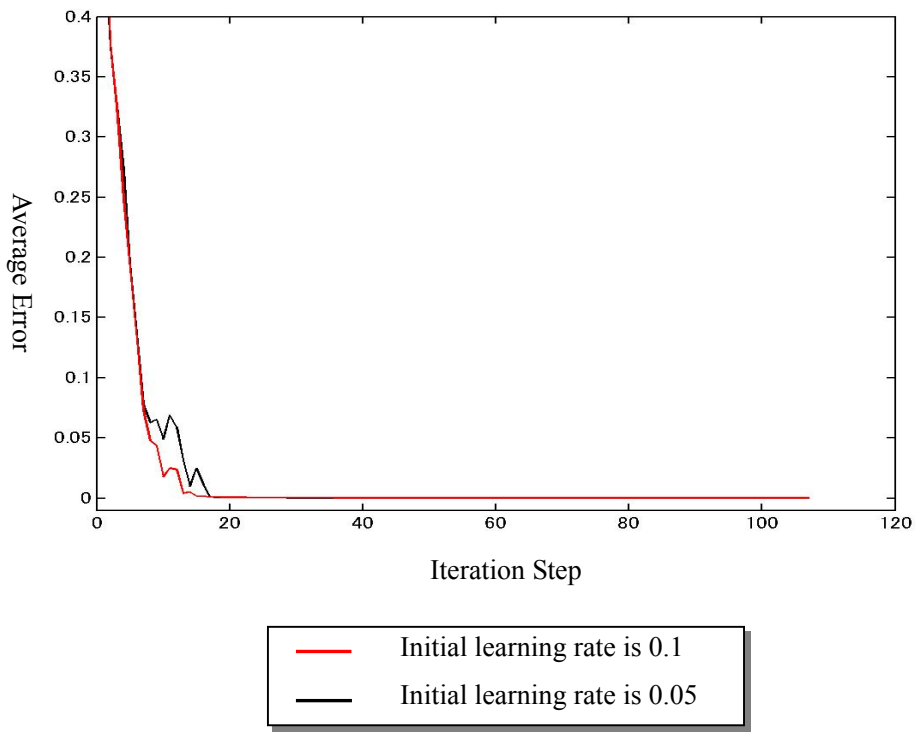


Fig.5.16 Error curves of different initial learning rates with initial width range (0.5, 0.6)

it can be seen that the red line descends faster and has less oscillations than black line. It results from there are some small crinkles locating at its descending channel, some of which can be stridden as the initial step size is 0.1. Moreover, after striding these crinkles, instead of arriving at a valley-bottom, the iteration stays at a point on the slope. It is the reason that the iteration step is less as the initial step size is larger. For the situation of initial widht range is (0.5, 0.6), its convergence procedure is shown as Fig.5.16, in which its steps with larger step size are about 20 more than that with smaller step size, the reason of descending quikly is similar with that of initial range is (0.7, 0.8). the reason of converging more slowly is the search strides the valley-bottom caused by the larger step size and oscillates damply to the just stridden valley-bottom. It is possible to lead to the iteration converge to differnet points on the error surface as changing the initial learning rate of width. Therefore, this kind of changing also

influence the performance of the proposed network. It is be proved from the comparisons among the results of Table.5.9, Table.5.10, and Table.5.12, although these kinds of difference are perhaps not very much.

There is an interesting phenomenon in the experiment that the rejection capabilities are different as the sequences of training samples are changed. The relative experiment results are given in Table.5.13 and 5.14, which show the corresponding rejection ratios for US dollars and Thai banknotes with the different sequences of training samples respectively. In the training set there are 10 par values of Chinese currency being as training samples. The sequence from small to large of their par values is called original order. The sequence from large to small is called reverse order.

Table.5.13 Rejection ratios for Thai banknotes with different training sequences

	(07, 0.8)		(0.4, 0.7)		(0.5, 0.8)	
	Original order	Reverse order	Original order	Reverse order	Original order	Reverse order
bg1000ab	100%	70%	100%	85%	100%	53%
bg1000cd	100%	100%	100%	73%	97%	100%
bg500ab	100%	41%	89%	90%	76%	50%
bg500cd	100%	100%	75%	63%	100%	83%
bg100ab	100%	53%	100%	94%	100%	76%
bg100cd	100%	50%	100%	84%	94%	80%
bg50ab	100%	100%	96%	88%	100%	50%
ba50cd	85%	95%	100%	100%	100%	100%
bg20ab	100%	75%	100%	65%	99%	67%
bg20cd	100%	57%	100%	89%	100%	94%
Average	**98.5%**	**74.1%**	**96.0%**	**83.1%**	**96.6%**	**75.3%**

As can be seen from these two tables, the average rejection ratios of being trained in original order are more stable than that of being trained reverse order. As being trained in reverse order, the best average rejection ratio is 91.83% for US dollar with initial width (0.7, 0.8), and the worst one is 67.67% all for US dollar with initial width (0.4, 0.7). The difference of them is about 24 percentages. But in the other situation, the largest difference is only about percentages. Moreover, the rejection ratios for some classes of currency are very bad, are even less than 20%. In these contrastive experiments, all of the restrictions such as initial weights, biases and width, network structures, training parameters, and so on, are same except sequence of training samples. What does this phenomenon result from? Please consider the error function governed by Eq.(3.10), which includes the input vector. If the sequences of input samples are different, the error function is then different, and the staring point on the error surface is

also different. Meanwhile, in each step of iteration the moving directions are also different since using the sequential version of gradient descent method, which calculates gradient with respect to the error produced by each samples in each step. If the sequence of samples is changed, the gradient in each step is different. Therefore, even if the other conditions are same the network cannot move to a same minimum with different data sequences.

Table.5.14 Rejection ratios for US dollars with different training sequences

	(07, 0.8)		(0.4, 0.7)		(0.5, 0.8)	
	Original order	Reverse order	Original order	Reverse order	Original order	Reverse order
us1a	100%	97%	100%	74%	97%	94%
us1b	100%	100%	100%	55%	100%	2%
us1c	99%	79%	100%	37%	99%	90%
us1d	100%	97%	98%	29%	99%	71%
us5a	100%	92%	100%	97%	100%	88%
us5b	96%	100%	100%	94%	99%	69%
us5c	100%	100%	100%	15%	100%	81%
us5d	100%	95%	100%	40%	100%	100%
us10a	100%	95%	100%	85%	100%	95%
us10b	100%	100%	100%	77%	100%	67%
us10c	100%	100%	97%	42%	100%	81%
us10d	100%	24%	100%	29%	100%	76%
us20a	100%	99%	100%	92%	100%	34%
us20b	100%	100%	100%	76%	100%	46%
us20c	100%	100%	100%	83%	100%	78%
us20d	97%	100%	91%	90%	99%	90%
us50a	100%	87%	100%	88%	100%	94%
us50b	99%	97%	100%	80%	100%	8%
us50c	100%	100%	100%	94%	100%	65%
us50d	100%	90%	100%	14%	100%	88%
us100a	100%	91%	100%	88%	100%	90%
us100b	99%	100%	100%	81%	100%	12%
us100c	100%	87%	97%	67%	98%	45%
us100d	100%	74%	99%	97%	100%	89%
Average	**99.58%**	**91.83%**	**99.25%**	**67.67%**	**99.63%**	**68.88%**

In order to improve the situation of influences to the performances of the proposed network caused by different sample sequences, some essential measures are needed. On the base of the reasons leading to this phenomenon analyzed above, it is difficult to change the certainty that different sample sequences lead to the different starting point

on the error surface. Therefore, it is attempted to compensate the influences caused by the stochastic learning method, sequential version of gradient descent. But the original batch version of the gradient descent is prone to make the iteration getting stuck in bad local minima and cannot converge completely. In this experiment, a combination method of the sequential and batch versions as mentioned in section 3.4 are applied. First, a sample data is taken randomly from each class of the training set, and then they are grouped together into a block. For example, there are 10 classes of Chinese currency in the training set, so there are 10 pieces of different Chinese currency coming from the different classes in a block. The number of the block is equal to that of the training sample in each class of the training set. After that, all of these blocks are presented sequentially as if each of them is representative of the whole training set. For each block, the batch version is employed. This kind of processing is just applied to the regulating of width in the following experiments, and the corresponding rejection ratios are shown in Table.5.15, and 5.16.

Table.5.15 Rejection ratios for Thai banknotes of different training sequences using combined method

	(07, 0.8)		(0.4, 0.7)		(0.5, 0.8)	
	Original order	Reverse order	Original order	Reverse order	Original order	Reverse order
bg1000ab	94%	90%	89%	55%	80%	66%
bg1000cd	100%	100%	100%	96%	71%	64%
bg500ab	96%	56%	59%	69%	97%	82%
bg500cd	100%	100%	99%	63%	51%	73%
bg100ab	99%	100%	54%	64%	99%	70%
bg100cd	100%	98%	63%	93%	76%	95%
bg50ab	100%	100%	64%	100%	52%	100%
ba50cd	90%	100%	99%	100%	63%	100%
bg20ab	100%	100%	91%	100%	99%	86%
bg20cd	96%	100%	76%	98%	69%	81%
Average	**97.5%**	**94.4%**	**79.4%**	**83.8%**	**75.7%**	**81.7%**

As can be seen from these two tables, the differences of the rejection ratios with the different sample sequences become smaller than using the simple sequential gradient method. The largest difference is about 6 percentages, which is 24 percentages as using sequential version. It reveals that the influences of sample sequences of using the combined algorithm are smaller than that of using the simple sequential gradient method. But as comparing these two Tables with Table.5.14 and 5.15, it can be found that as using the combined algorithm, although the average rejection ratios are increased

for the other cases except for the two cases US dollars with initial rang (0.7, 0.8) and Thai banknotes with initial range (0.4, 07), in which the average rejection ratios for them are kept stable with the reverse sample sequence, but the combined algorithm leads to the average rejection ratios are decreased with the original sample sequence. It is the cost of decreasing differences of this combined method, with which the probabilities of getting better minima are smaller than that with the simple sequential version of the gradient descent.

Table.5.16 Rejection ratios for US dollars of different training sequences using combined method

	(07, 0.8)		(0.4, 0.7)		(0.5, 0.8)	
	Original order	Reverse order	Original order	Reverse order	Original order	Reverse order
us1a	100%	98%	82%	94%	42%	89%
us1b	100%	97%	99%	100%	94%	33%
us1c	57%	68%	96%	95%	54%	97%
us1d	53%	77%	91%	59%	48%	63%
us5a	98%	88%	90%	91%	90%	75%
us5b	100%	77%	99%	100%	90%	50%
us5c	100%	99%	85%	100%	69%	100%
us5d	100%	96%	87%	98%	85%	100%
us10a	100%	96%	97%	82%	85%	93%
us10b	100%	74%	100%	98%	82%	54%
us10c	100%	100%	61%	100%	90%	99%
us10d	100%	100%	100%	99%	85%	93%
us20a	100%	93%	100%	96%	98%	63%
us20b	100%	90%	95%	99%	81%	54%
us20c	100%	100%	94%	48%	86%	99%
us20d	100%	100%	79%	100%	96%	79%
us50a	100%	100%	70%	89%	86%	98%
us50b	100%	96%	92%	92%	66%	17%
us50c	100%	84%	66%	100%	67%	97%
us50d	100%	99%	85%	100%	99%	100%
us100a	100%	100%	96%	97%	91%	90%
us100b	100%	96%	100%	100%	92%	41%
us100c	100%	78%	97%	100%	84%	54%
us100d	100%	88%	100%	100%	91%	88%
Average	**96.17%**	**91.42%**	**90.04%**	**93.21%**	**81.29%**	**76.08%**

Furthermore, in order to decrease influences of the fixed sample sequences for network performances, we can reselect a sequence of training samples in the training set

randomly after finishing the each training of this whole training set. This method also can decrease the influences of sample sequence for network properties.

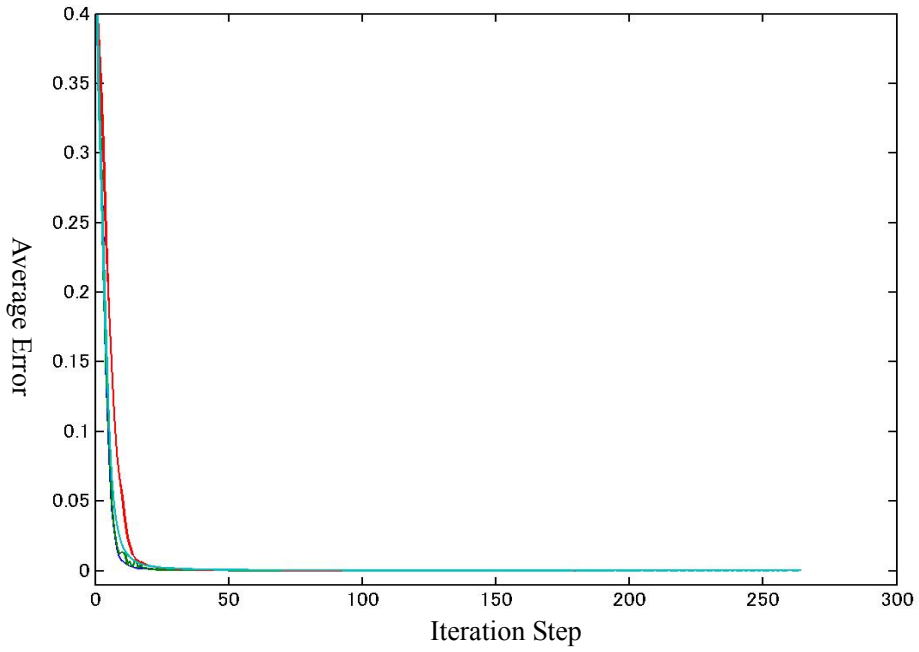
In addition in order to further verify the rejection capabilities of the proposed network, there are 252 patterns of other 16 counties' currency, which also are evaluated by the proposed system. Furthermore, the contrastive experiments using the sigmoid activation function are done with the same restrictions. The results are presented in Table.5.17. It further proves the effectiveness of our proposed system on rejection capabilities for unknown patterns.

Table.5.17 Rejection Ratios for other Unknown Currencies

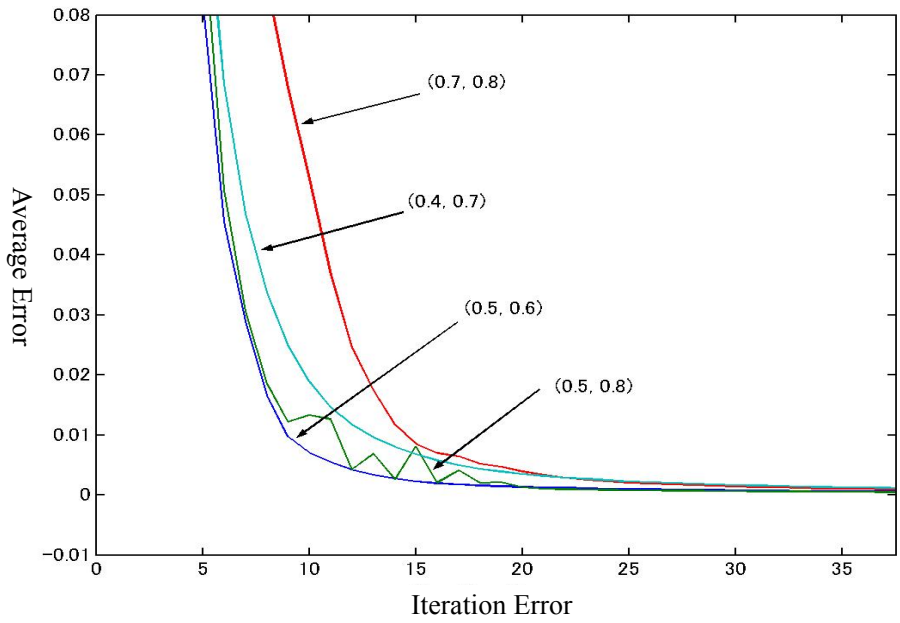
Nation	Gaussian Function		Sigmoid Function	
	Rejection Ratio	Rejected / Total Piece	Rejection Ratio	Rejected / Total Piece
AUS	95.8%	23 / 24	70.8%	17 / 24
BER	100%	16 / 16	62.5%	10 / 16
CAN	100%	28 / 28	85.7%	24 / 28
CHN (T)	100%	12 / 12	41.7%	5 / 12
DAN	100%	8 / 8	50.0%	4 / 8
DEU	100%	19 / 20	75.0%	15 / 20
ENG	100%	16 / 16	93.8%	15 / 16
FRA	87.5%	7 / 8	62.5%	5 / 8
ITA	95%	19 / 20	65.0%	13 / 20
KOR	91.7%	11 / 12	58.3%	7 / 12
NED	100%	16 / 16	50.0%	8 / 16
NOR	100%	8 / 8	62.5%	5 / 8
SAU	100%	20 / 20	85.0%	17 / 20
SPA	93.8%	15 / 16	25.0%	4 / 16
SWE	100%	16 / 16	62.5%	10 / 16
SWH	91.7%	11 / 12	25.0%	3 / 12
Average	**97.62%**	**246/252**	**65.42%**	**157/252**

5.3.2 Conjugate gradient algorithm

In this section, the widths parameters of the proposed network are regulated automatically using the conjugate gradient method mentioned in section 3.4. The experiment conditions here are same with that of using the sequential version of gradient descent. The initial ranges of weights, biases, and widths are also identical, and



(a) Original convergence procedures



(b) Local enlarged convergence procedures

Fig.5.16 Iteration procedures of different initial width values with the conjugate gradient

the sequence of training samples is original order. It is ensured that the iterations with the different algorithms can start from the same initial points, so that these two of algorithms can be compared together. The influences of the conjugate gradient algorithm to the performances of the proposed network also are analyzed. Let us observe the error curves of convergence procedures of the network with the 4 different initial width ranges. They are illustrated in fig.5.16, in which (a) shows the whole convergence procedures of 4 error curves, and (b) shows an enlarged special part of them.

Table.5.18 Itration steps of different initial width ranges with two methods

Intial Range	(0.4, 0,7)	(0.5, 0.8)	(0.5,0.6)	(0.7, 0.8)
Sequential gradient	131	110	88	83
Conjugate gradient	265	154	170	182

As can be seen form Fig.5.16 that all error curves except that with initial rang (0.5, 0.8) are considerable smooth, in which there are not any oscillations. It reveals that the iterations moving along the direction given by the conjugate gradient can find out the

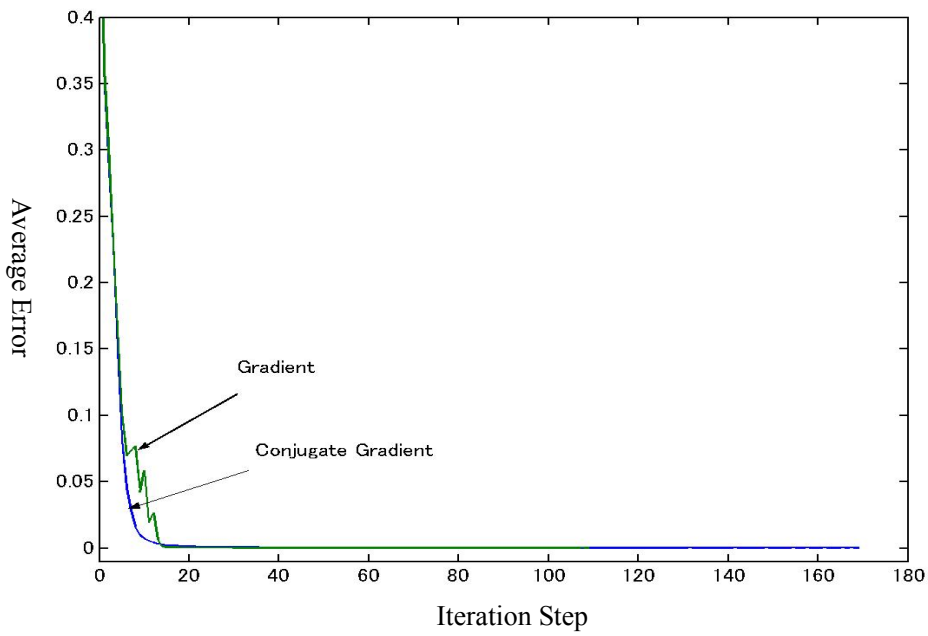


Fig.5.17 Error curves of initial width range (0.5, 0.6) with different leaning algorithms

comparative smooth channels on the error surface and converge to a minimum. But from Table 5.18 we can see that the iteration steps of the conjugate gradient method are all greater than that of the sequential gradient method in this experiment. Why does it happen? Perhaps the answer can be found from Fig. 5.17 and 5.18, which show the error curves of initial width ranges (0.5, 0.6) and (0.5, 0.8) with these two different leaning algorithms respectively. It can be found that in the initial stages the iterations along the conjugate gradient direction descend more quikly than that along the negative gradient direction. But near the minimum the speed of convergence is slow than that of using Sequential gradient method. It is the reason that the iteration steps of it are greater than that of the gradient descent method.

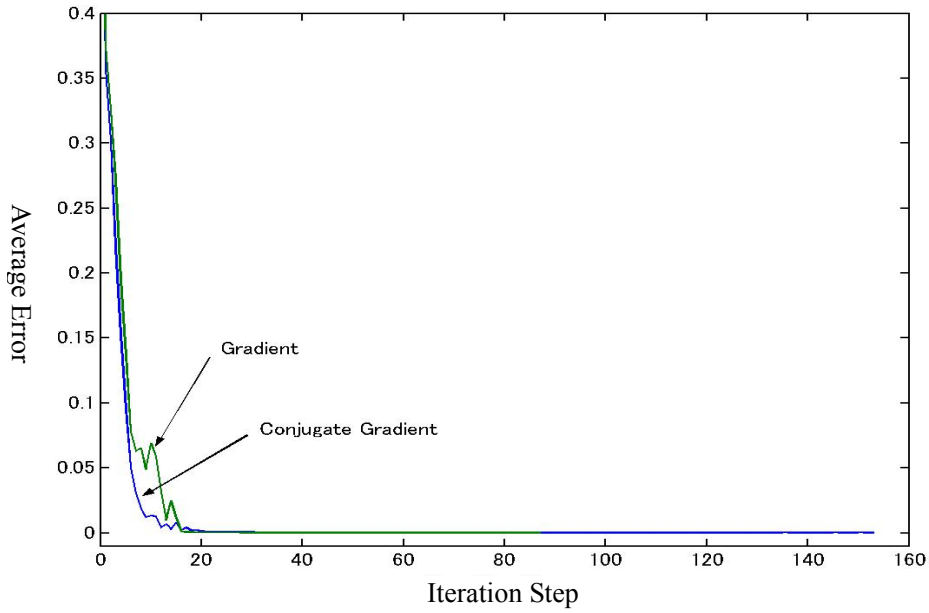


Fig.5.18 Error curves of initial width range (0.5, 0.8) with different leaning algorithms

As the conjugate gradient method is employed, the corresponding rejection capabilities of the proposed network for US dollars and Thai banknotes with different initial width ranges are shown in table 5.19 and 5.20. Comparing these results with the corresponding results in table 5.9 and 5.10, it cannot find any advantages of the conjugate gradient algorithm on improving the performances of the proposed network.

Table.5. 19 Rejection ratios for Thai banknotes with the conjugate gradient

Initial range of width	(0.4, 0.7)	(0.5, 0.6)	(0.5, 0.8)	(07, 0.8)
bg1000ab	100%	99%	100%	100%
bg1000cd	100%	100%	97%	100%
bg500ab	89%	100%	76%	100%
bg500cd	75%	93%	100%	100%
bg100ab	100%	100%	100%	100%
bg100cd	100%	100%	94%	100%
bg50ab	96%	100%	100%	100%
ba50cd	100%	99%	100%	85%
bg20ab	100%	100%	99%	100%
bg20cd	100%	100%	100%	100%
Average	**96.0%**	**99.1%**	**96.6%**	**98.5%**

Table 5.20 Rejection ratios for US dollars with the conjugate gradient

Initial range of width	(0.4,0.7)	(0.5,0.6)	(0.5,0.8)	(0.7,0.8)
us1a	99%	97%	100%	100%
us1b	100%	100%	100%	98%
us1c	98%	100%	99%	94%
us1d	95%	100%	93%	38%
us5a	100%	100%	100%	97%
us5b	100%	100%	99%	63%
us5c	100%	100%	100%	100%
us5d	100%	100%	100%	100%
us10a	100%	100%	100%	100%
us10b	100%	100%	100%	79%
us10c	94%	97%	90%	100%
us10d	100%	100%	100%	100%
us20a	100%	100%	100%	98%
us20b	100%	100%	100%	88%
us20c	100%	100%	100%	100%
us20d	79%	99%	100%	95%
us50a	100%	100%	100%	100%
us50b	100%	100%	100%	87%
us50c	100%	100%	100%	94%
us50d	100%	100%	100%	100%
us100a	100%	100%	100%	100%
us100b	100%	100%	100%	86%
us100c	88%	99%	94%	100%
us100d	100%	99%	98%	99%
Average	**98.04%**	**99.63%**	**98.88%**	**92.33%**

5.3.3 Experiments of hybrid learning algorithm

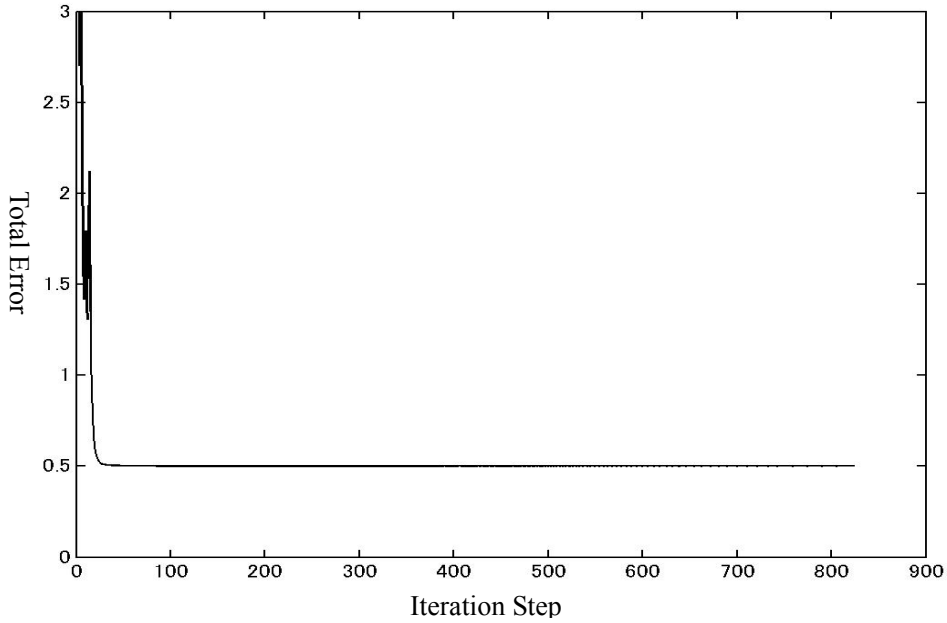
In this section, the relative experiments of the proposed hybrid-learning algorithm are involved. Let us introduce the conditions of the experiments and initial values of some parameters. The parameters ε_T and λ in Eq.(3.38) and Eq.(3.39) are 10^{-4} and 0.03, respectively. The other conditions of learning are shown in Table 5.21. The initial range of weights and biases are fixed respectively in (-0.3, 0.3) and (0.0, 1.0). Then the initial values of the widths are generated in the ranges of (0.5, 0.6), (0.45, 07), (0.2, 0.9) and (0.2,1.2) respectively.

Table 5.21 Conditions of Learning

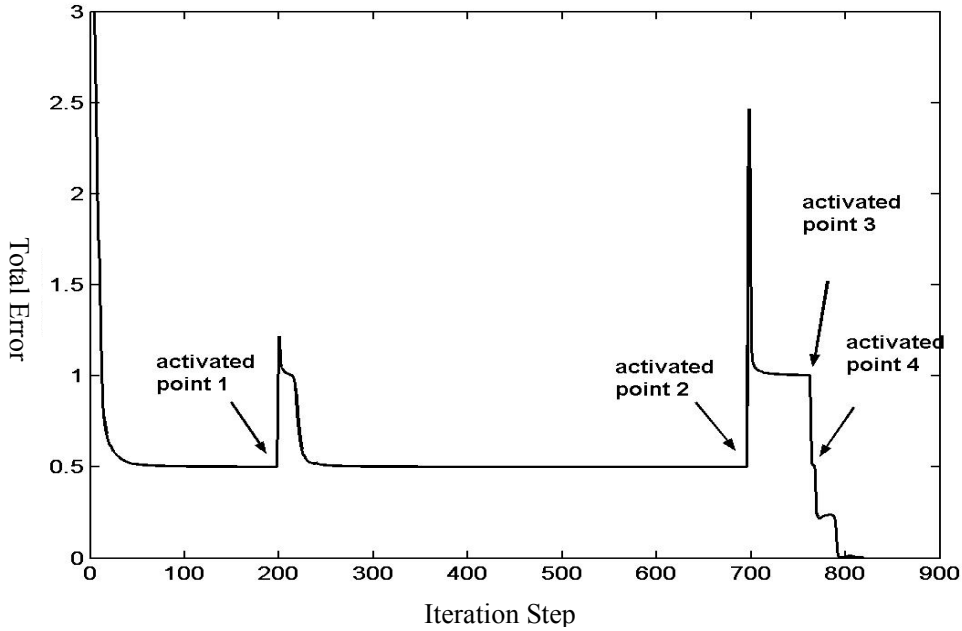
Initial Learning Rate α	0.05
Momentum Coefficients β	0.80
Oscillation Coefficient γ	-0.0001
Reflection Coefficient C_r	1.0
Expansion Coefficient C_e	1.2
Contraction Coefficient C_c	0.7
Convergence Criterion	Average of square error<0.0001
Maximum Iteration No.	20000

First, 10 classes of Chinese currency were used to train the network by the proposed mentioned in section 3.5. 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 according 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.

In order to observe and analyze the influences of the proposed algorithm for the convergence of error function, it is needed to draw the response curve of the error firstly. Fig.5.19 just shows this response curve of the training procedure for the error of all 10 classes of the training samples with different algorithms as the initial width range is



(a) Error response with the sequential gradient descent method



(b) Error response with the proposed hybrid-learning algorithm

Fig.5.19 Response curves of total error with different algorithms

(0.2, 1.2), in which (a) shows the error response with the sequential version of gradient descent method, (b) shows the error response with the proposed hybrid-learning algorithm.

As can be seen from Fig.5.19, as the sequential gradient algorithm is employed, there are several oscillations occurring on its error response curve until it converge to the bad local minimum, in which the error is approximately equal to 0.5. And then the iteration is getting stuck in this minimum and remains there indefinitely. It reveals that if there are some small crinkles on the way to a local minimum, it can be stridden relying on the essentialities of the sequential algorithm and appropriately regulating the step sizes. But if the iteration falls into a deep minimum, it is difficult for this algorithm to jump out from the minimum by itself. Afterwards, let us see the other situation, in which the proposed algorithm is employed. You can see after the iteration remains 0.5, there are several times of big oscillation occurring, which are just caused by activations of the proposed method. Because the first stage of the algorithm just is the sequential gradient method with a momentum, these oscillations are caused by the second stage of the proposed algorithm. We can see that the error finally approaches to zero experiencing these several times of oscillation. It reveals that the proposed algorithm can pull the iteration out of the bad local minimum and find better minima. In the figure the tags indicate the four activation points, in which the second stage acts. The parts on this error curve behind these activation points are different apparently. What is the reason? It will be benefit for understanding the proposed algorithm by observing and analyzing the error responses of each class of the training samples, Chinese currencies. From Fig.5.20 to 5.29 show the respective error response curves of each class of Chinese currencies in this whole convergence procedure.

As comparing each of these figures from Fig.5.20 to 5.29, it can be found that the four times of the activation of the second stage in the proposed algorithm are respectively active on the class RMB100H, RMB50T, RMB20T, and RMB10T. The corresponding activated situations are given in table 5.22. It can be seen from the figure of RMB100H that the error gets trapped into a local minimum, in which the error caused by class RMB100H is about 0.5, soon after starting the iteration. After the criterion formularized In Eq.(3.38) is satisfied, the proposed algorithm begins to execute its second stage. The error caused by RMB100H is 0.500069 as can be seen in table 5.22. Depending on the downhill simplex method, a new point on the error surface is discovered, in which the error is 0.481053 and the iteration is extricated from this bad local minimum successfully. And then the first stage of the algorithm starts again, the iteration still move along the direction of the negative gradient descent. You can see the influences of this leap to errors caused by each class of training samples from these

figures. The values of all of the errors at this point are increased, in which the increment of RMB10T is the largest. However, after several steps of iteration, except for RMB10T, the search of all class of training samples can move on the way to convergence along the negative gradient directions. The iterations for RMB10T unfortunately got trapped into a new local minimum, in which the error is still about 0.5. As the error caused by RMB10T is 0.500004 at *activated point 2*, the second stage of the algorithm is activated again. When the iteration search escapes from this local minimum by using simplex, its value decreases to 0.362579. At the same time, this leap also leads to apparent increments of errors caused by other classes of training sample. This transition also resulted in that the total error function was increased to about 2.5. From this new point, using the first stage of the algorithm, the errors of some classes of training sample converged very soon, but unfortunately the errors of RMB50T and RMB20T are stuck in local minima again. Hence the second stage of the algorithm is activated successively 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. The influences of these two leaps for the errors caused by other training samples are not very much. Then the first stage is reemployed and their errors converged to the point according with the criterion. The new point found by the fourth activation lead to the increase of the error of RMB10T, but it converged by inactivating the second stage of the algorithm.

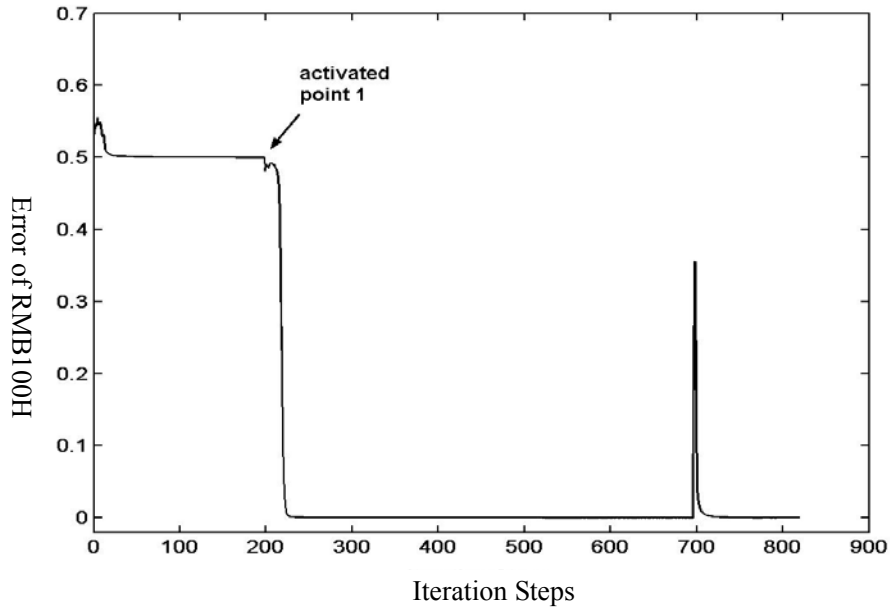


Fig.5.20 Error curve of RMB100H

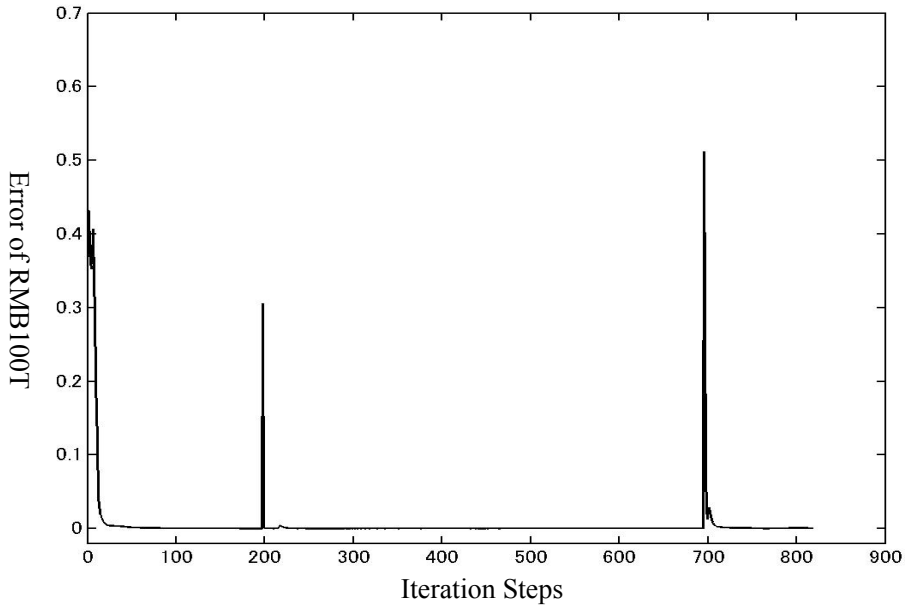


Fig.5.21 Error curve of RMB100T

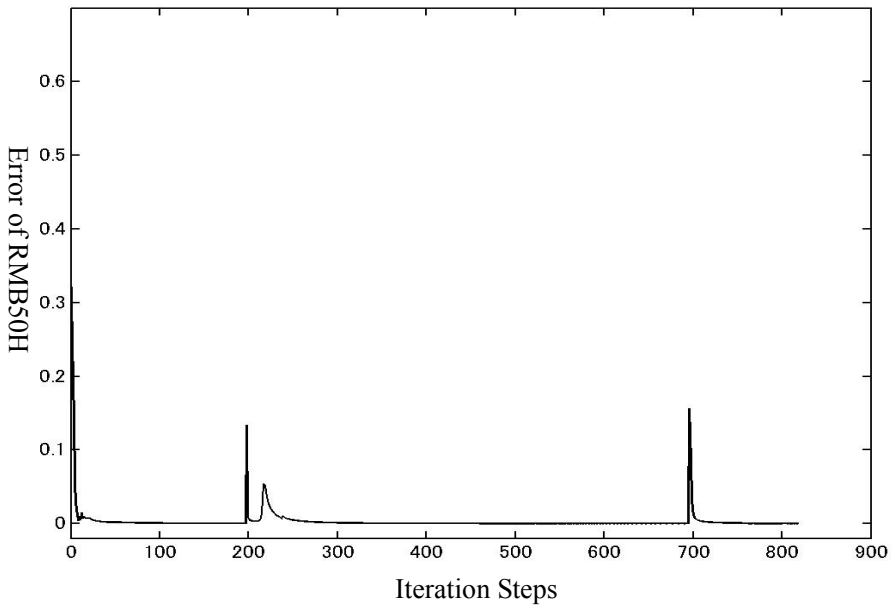


Fig.5.22 Error curve of RMB50H

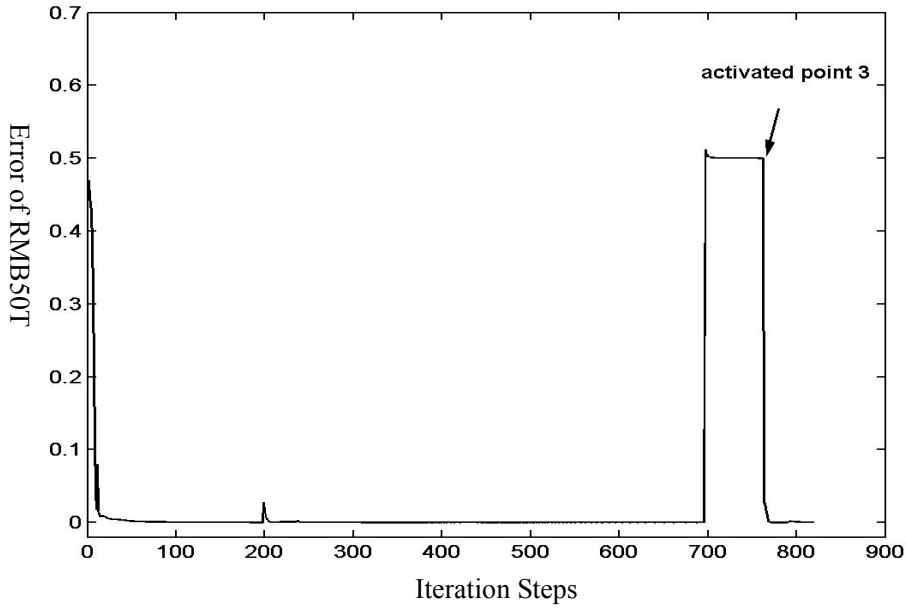


Fig.5.23 Error curve of RMB50T

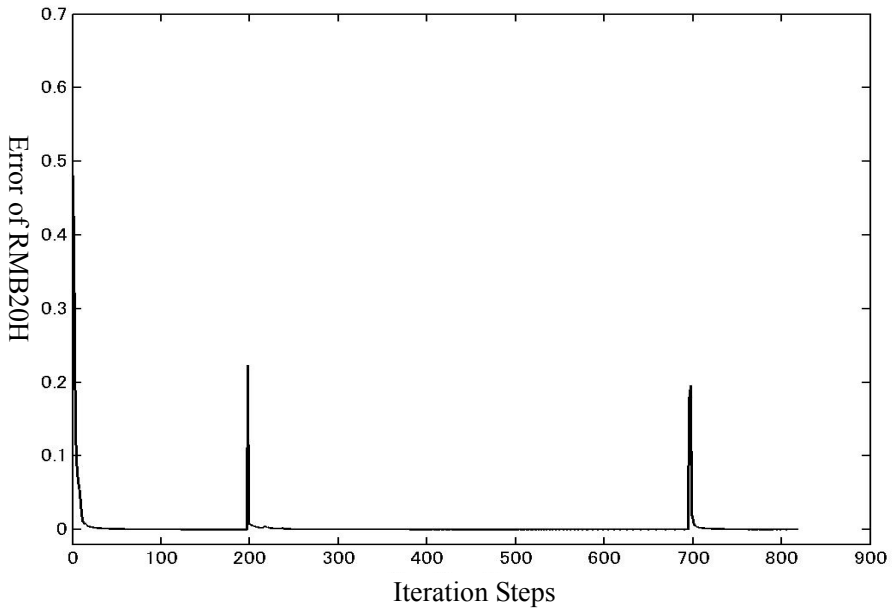


Fig.5.24 Error curve of RMB20H

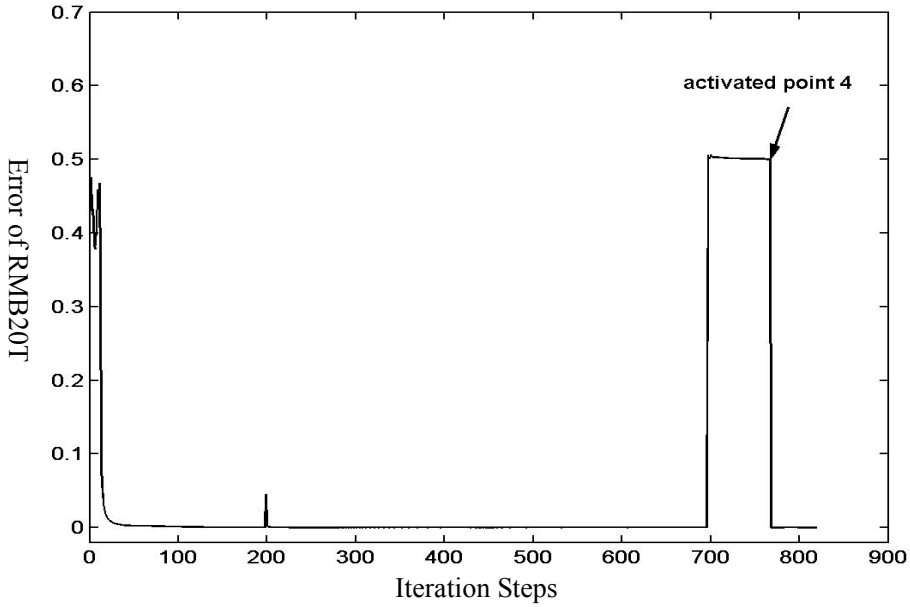


Fig.5.25 Error curve of RMB20T

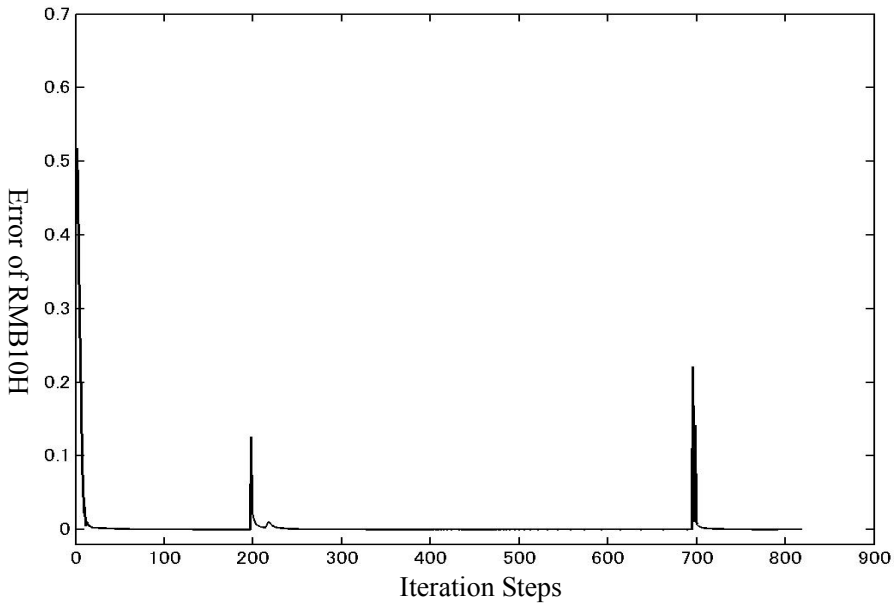


Fig.5.26 Error curve of RMB10H

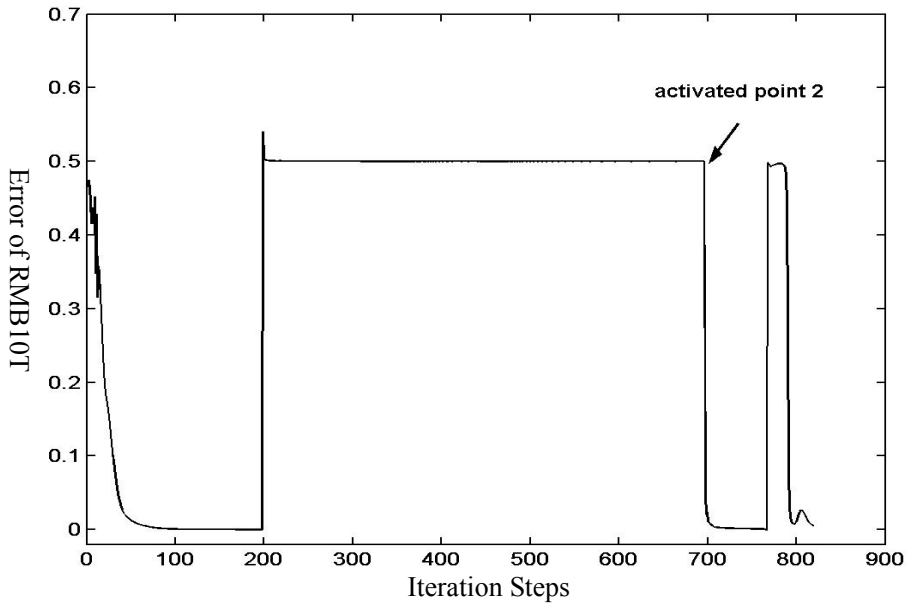


Fig.5.27 Error curve of RMB10T

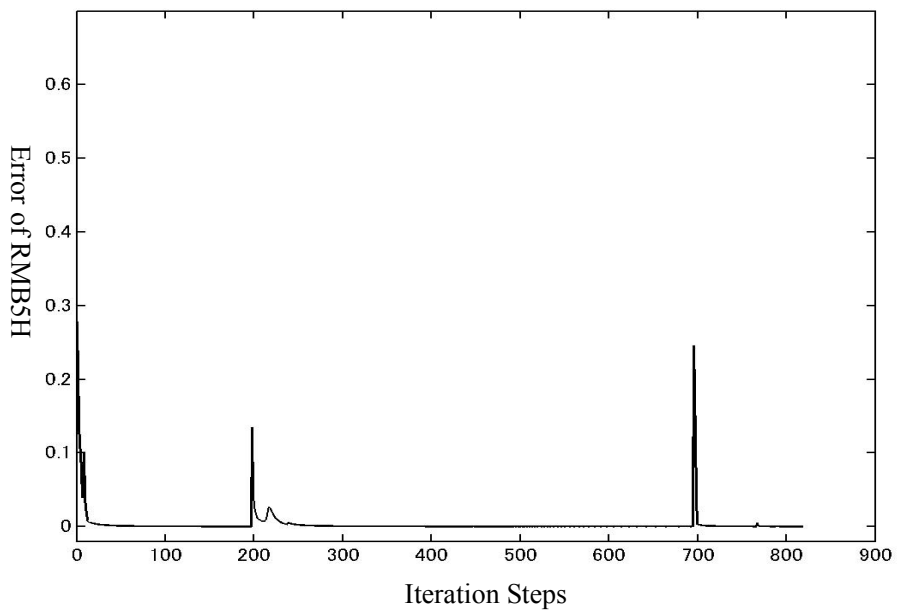


Fig.5.28 Error curve of RMB5H

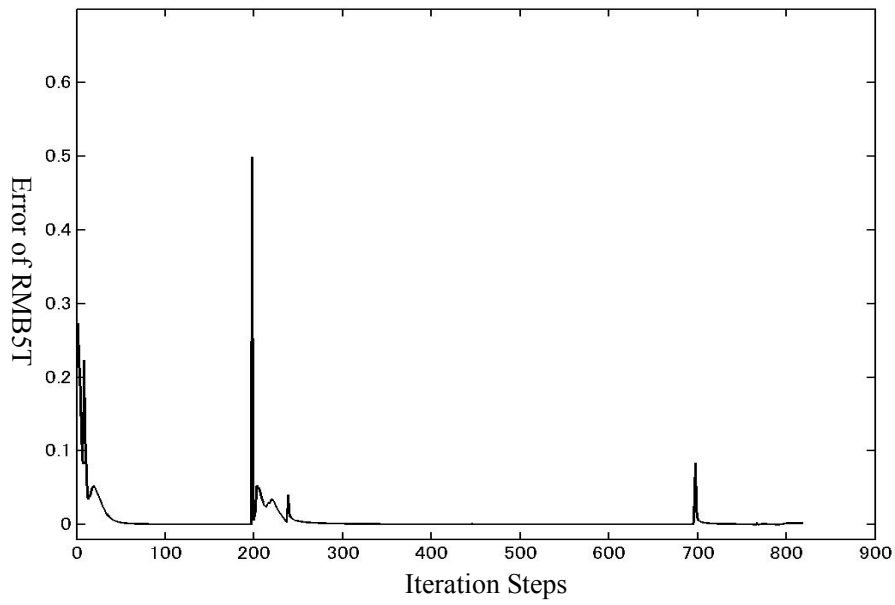


Fig.5.29 Error curve of RMB5T

After training the network using the proposed hybrid-learning algorithm, in order to verify the recognition capabilities of the network for known patterns again, 10 classes of Chinese currency being as the known data, each of which involves 100 pieces including some worn, smutched and incomplete currencies, are inputted the trained network. The corresponding recognition results are shown in Table.5.23.

Table 5.22 The active situations of the proposed algorithm

	Activated Class	Staring Point	Ending Point
Activated point 1	RMB100H	0.500069	0.481053
Activated point 2	RMB10T	0.500004	0.362579
Activated point 3	RMB50T	0.500259	0.299979
Activated point 4	RMB20T	0.499998	0.457219

As can be seen from Table.5.23, for each of initial range of the width parameters, although the recognition ratios for each class have a little bit differences among them, the average recognition ratios for all classes of Chinese currency are almost identical. They are greater than 98%. It reveals that given an 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 rang is not ideal, the system also can obtain the satisfied recognition effects by alternately utilizing the two stage of the algorithm. It shows that the system using this hybrid-learning algorithm is satisfied on aspect of recognition capabilities for known currency patterns.

Table.5.23 Recognition Ratios for Chinese currencies

	Initial range of width parameters			
	(0.5, 0.6)	(0.45, 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%**

Table.5.24 Rejection ratios for unknown Thai banknotes

Thai banknots	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%**

Therefore, to verify the rejection capabilities of the system for unknown currencies, 24 classes of US dollar and 10 classes of Thai banknotes are also recognized by the proposed system, there are 100 pieces in each class of US dollars and Thai banknotes, respectively. Table.5.24 and Table.5.25 show the corresponding rejection ratios of them. In this case, all classes of US dollar and Thai banknote are unknown patterns for the proposed network. They all should be rejected ideally.

Table.5.25 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)
US100a	100%	100%	99%	99%
US100b	100%	100%	99%	100%
US100c	100%	98%	100%	99%
US100d	97%	98%	97%	80%
US50a	100%	100%	100%	99%
US50b	100%	100%	98%	97%
US50c	100%	100%	100%	98%
US50d	100%	100%	100%	98%
US20a	100%	100%	100%	99%
US20b	100%	100%	100%	100%
US20c	99%	93%	90%	95%
US20d	99%	97%	97%	85%
US10a	100%	100%	100%	100%
US10b	100%	100%	100%	100%
US10c	99%	99%	100%	100%
US10d	98%	98%	100%	100%
US5a	99%	100%	99%	100%
US5b	100%	100%	99%	99%
US5c	100%	100%	100%	100%
US5a	100%	100%	100%	100%
US1a	99%	100%	97%	96%
US1b	100%	100%	92%	100%
US1c	99%	98%	75%	87%
US1d	100%	96%	87%	90%
Average	**99.54%**	**99.04%**	**97.04%**	**97.13%**

Form Table.5.24 and Table.5.25, it can be found 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, but it decreases about 2 percentages for US dollars. However, the rejection

ratio for the pattern US1c is only 75% that means 25 pieces of US1c are misrecognized as some class of Chinese currencies when the range is (0.2, 0.9). 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.5.21, 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.5.26 and 5.27, in which for US dollars 'H' means the head side including two directions 'a' and 'b', the other 'T' denotes 'c' and 'd'. As can be seen from these tables that although the cost function accords with the convergence criterion, the performance of the system is unsatisfied relying on converge to a bad local minimum.

Table.5.26 Rejection ratios for a specified momentum

US Dollars	Initial range of width parameters
	(0.2, 0.9)
US100H	93%
US100T	95.5%
US50H	71.5%
US50T	100%
US20H	82%
US20T	98.5%
US10H	61.5%
US10T	100%
US5H	80%
US5T	100%
US1H	94.5%
US1T	87.5%
Average	88.67%

Table.5.27 Rejection ratios for a specified momentum

Thai Banknotes	Initial range of width parameters
	(0.2, 0.9)
BG1000H	100%
BG1000T	66%
BG500H	71%
BG500T	49%
BG100H	100%
BG100T	97%
BG50H	100%
BG50T	100%
BG20H	93%
BG20T	100%
Average	87.6%

As comparing Table.5.26 and 5.27 with Table.5.24 and 5.25, it can be found that as the proposed hybrid-learning algorithm is employed to search suitable widths of the proposed Gaussian functions, it is achievable to explore better solutions, with which the proposed system can recognize known currency patterns and reject unknown currency patterns more effectively.

Furthermore, the rejection capabilities of the proposed system are compared with that of a combined recognition system, which includes linear templates formed by mean and variance of slab values of learning samples besides of the perceptron with sigmoid activation function. The results are shown in Table.5.28. 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.5., it can be found that rejection capabilities for US dollars and Thai banknotes 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.5.28 that 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 simple.

Table.5.28 Comparison of rejection capabilities with different methods

Nation	Total Pieces	Rejection ratio	
		NN + linear template	Proposed method
US	2400	96.67%	99.54%
Thai	1000	93.1%	97.7%
Can	28	87.5%	100%
Aus	24	89.29%	95.83%

5.4 Experiments on the real-time system

In the experiment on the real-time system, the learning samples, the recognizing objects and the NN structures are the same with that of the experiments on PC, some results of the experiment with sequential version of gradient algorithm are shown in Table 5.29.

Table 5.29. Average recognition and rejection capabilities on real-time system

	Chinese Currencies	US Dollars	Thai Banknotes
Recognition Capabilities	99.15%	99.52%	97.55%
Rejection Capabilities	0.85%	0.48%	2.45%

It can be found from this table that because of being with the same restriction as that of on PC, the recognition capabilities for known data and rejection capabilities for unknown data of the proposed network executed on this real-time system are equivalent to those of the experiments on PC.

Table.5.30 Recognition Time for Each Piece of Currency

NN Structures	$50 \times 30 \times 10$			
Activation Function	Sigmoid		Gaussian	
Program Language	CLK	Time (μs)	CLK	Time (μs)
C	16014.1	960.8	14092.4	845.5
Assemble	5598.1	335.9		

In addition, the time of recognition for a piece of currency using the Gaussian function and the sigmoid function was respectively calculated. The results were shown in Table 5.30. A clock cycle of DSP(TMS320C31) is 60ns, from the table it can been seen that the structure of the NN was 50×30×10, and as C language is used, 16014.1 clock cycles are needed for recognizing each piece of currency when the activation function is sigmoid function, so it takes 960.8μs for recognizing each currency using the traditional sigmoid NN. Its speed is already enough for real-time recognition system. In contrast, while the proposed Gaussian function is used as the activation functionof the network, 14092.4 clock cycles, which is equal to 845.5μs, are needed. Moreover, once the assemble language is used to program, the recognition speed is 3 times of that using C language. While business machines such as readers and sorters need less than 10ms per piece as calculation time for currency recognition, all mentioned above reveals the potentials of high-speed recognition of the proposed system on business machines.

Chapter 6
CONCLUSIONS

In this paper, a kind of currency recognition system, in which a three-layer feedforward neural network is employed as the classifier, is proposed in order to improve the performances of currency recognition systems on rejecting unknown currency patters with the premise that the recognition capabilities for known currency patterns should be guaranteed. In the feedforward neural network a kind of Gaussian function is proposed as the activation function, which is employed in all units on the hidden layer and the output layer. The characteristics of this activation function are analyzed. It is a kind of localized function, its activation approaches zero as the distance to its center approaches infinity. Its active region is governed by its width parameter. Just relying on this property of the proposed Gaussian, the rejection capabilities of the system can be improved.

However, the proposed Gausian function is different with not only the simple radial

basis function but also the Gaussian bars function. It is a ridge-like function in multi-dimensional space. The relative analysis and discussions of the differences of them have been done in section 3.3. The neural network employing the proposed Gaussian activation function shows its advantages on improving the rejection capabilities for unknown currencies just because of its distributions features in multi-dimensional space.

In the part of preprocessing, the techniques of mask set and slab values are applied. The values of two thresholds being used to detect the corresponding edges of Chinese currencies are determined. The dimensionality of currency images is compressed adequately and the digital features of them are extracted effectively. The problems that the different currencies have the same slab values are resolved by using masks. The application of the mask set produces many slab values for one currency image and make the network can obtain adequate features of recognition objects. The influences of positions of the mask set to performances of the proposed system are also analyzed in experiments. It is benefit for improving the performance of the system by finding the appropriate positions of the mask set.

In the part of the recognition, because of only using the three-layer feedforward neural network with the proposed Gaussian function as the classifier, the structures of this currency recognition system are simpler than that of conventional recognition systems. The experiments about the influences of the parameters of the Gaussian function to performances of the system are designed and executed, and the corresponding results are analyzed. It can be found from these results that the performances of the system are very sensitive to variations of the width parameter in the Gaussian function. The improved back propagation algorithm is used to optimize the weights and biases of the network. The sequential gradient algorithm fitting for the proposed network are derived, and applied to optimize the width parameters. At the same time, it is found in this experiment that the different sequences of training currency samples influence the rejection capabilities of the system remarkably as using the sequential version of the gradient descent algorithm, and the reasons leading to this phenomenon are analyzed. The influence of the sample sequence is weakened by using a combined method of the sequential and batch version of the gradient descent.

In order to weaken influences of initial values of the width parameter to performances of the system and explore appropriate width parameters in larger regions, a hybrid-learning algorithm is proposed in this research. This hybrid-learning algorithm is composed of two steps: The first step is the sequential gradient descent method, which is used to search local minima near starting points quickly. The second step is a random search method is employed to extricate the search from local minima with the downhill

simplex. Once the search gets trapped in a local minimum this step will be activated. In the second stage of the algorithm, a random vector is mixed in increment terms of parameters to be optimized, and the downhill simplex method is used to explore appropriate coefficients of it to pull the search escaping from local minima and explore a better point in the search space. Because the downhill simplex method does not need to calculate derivatives and it intends to enclose the minimum inside an irregular volume defined by a simplex, the second step of the proposed algorithm is more efficent on pull the search out of the local minimum.

Just because of mixing a random vector in increment terms of width parameters and using the downhill simplex method to optimize coefficients of them, the span and directions of iteration search have more combinations, and hence it can extricate the iteration search from local minima more quickly and easily and explore optimal width parameters in a larger range to improve the recognition and rejection capabilities of the system.

Results of the experiment show the considerable potentials of the network with the proposed activation function on rejecting unknown currencies. Furthermore, it reveals that the proposed hybrid-algorithm is insensitive for the initial range of the width parameters since it can utilize the sequential gradient descent stage directly as the initial range of parameters is proper, and alternately use the two stages of the algorithm once the initial range is improper. Therefore, the currency recognition system with the proposed the Gaussian activation function and the algorithm can correctly classify the currency patterns having been learned and effectively reject unknown currency patterns.

Furthermore, in order to prove the practicability of the proposed system on business developments and applications, the verified and contrastive experiments of the proposed system are also executed on the real-time system. The corresponding experiment results show the recognition process is fairly rapid due to the relatively simple system structures. The potentiality of the proposed system on business developments and applications is also revealed by the experiments.

REFERENCES

[1] Takeda, F. and Omatu, S.: High Speed Paper Currency Recognition by Neural Networks, *IEEE Trans. on Neural Networks,* Vol.6, No.1, pp.73-77 (1995).

[2] Takeda. F., Omatu, S. and Onami. S.: 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] Takeda, F. and Omatu, S.: 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] Takeda F., Omatu S. and Matsumoto Y.: 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] Takeda F., Nishikage T. and Matsumoto Y.: 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] Takeda F., Nishikage T. and Omatu S.: Banknote Recognition by Means of Optimized Masks, Neural Network, and Genetic Algorithm, *Engineering Applications of Artificial Intelligence 12*, pp. 175-184 (1999).

[7] Takeda F., Nishikage T.: Development of a neuro-templates matching recognition method for banknotes, *Trans IEE of Japan,* Vol.121-C, No.1, pp. 196-205 (2001)

[8] Takeda F., Tanaka M.: 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).

[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, pp1234-1244 (2000)

[10] C. M. Bishop: "Neural Network for Pattern Recognition", Oxford University Press, New York, U.S.A.(1995).

[11] Widrow, B., Winter, R.G. and Baxter, R.A.: Layered Neural Nets for Pattern Recognition, *IEEE Transaction Acoustic, Speech & signal Preprocessing,* Vol.36, No.7, pp. 1109 –1118 (1988).

[12] Baba 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] Xugang Wang, Zheng 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] Zaiyong Tang, Gary J. K.: 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] Sheng Ma, Chuanyi Ji: Performance and Efficiency: Recent Advances in Supervised Learning, *Proceedings of the IEEE*, Vol. 87, No.9, pp. 1519-1535 (1999)

[18] Nelder J. A. and Mead R., A simplex method for function minimization, *Computer Journal*, Vol. 7, pp. 308-313 (1965)

[19] Patrick P.: Minimization methods for training feed forward neural networks, *Neural networks,* Vol. 7, No. 1, pp.1-11 (1994)

[20] 竹田史章，西蔭紀洋，藤田靖：「自己学習型ニューロ紙幣識別ボードの開発とその汎用展開」，電気学会論文誌(C)，121 巻 2 号，357~365 (平 13)

[21] 竹田史章，大松繁：「ニューロ紙幣識別ボードの開発」，電気学会論文誌(C)，116 巻 3 号，336~340 (平 8),

[22] Baiqing Sun, Fumiaki Takeda: Research on the Neural Network with RBF for

Currency Recognition, *Proceeding of IASTED International Conference on Neural Networks and Computation Intelligence*, Switzerland, pp201-205, 2004;

[23] Fumiaki Takeda, Baiqing Sun: Discussion for Rejection Ability of Unknown Foreign Currencies by Gaussian Neural Recognition System, *Taiwan-Japan Symposium on Fuzzy systems and Innovational Computing*, Japan, 2004;

[24] Baiqing Sun, Fumiaki Takeda: Proposal of Neural Recognition with Gaussian Function and Discussion for Rejection Capabilities to Unknown Currencies, *Proceeding of International Conference on Knowledge-based Intelligent Information & Engineering Systems*, New Zealand, pp859-865 (2004)

[25] Baiqing Sun, Fumiaki Takeda: Chinese Currency Recognition Using the Neural Network with RBF, *Proceeding of SICE System Integration Division Annual Conference*, Japan, pp476-477 (2003)

[26] Baiqing Sun, Fumiaki Takeda: Application of a Neural Network with Gaussian Function on Currency Recognition System, *Proceedings of SICE annual conference 2005,* Japan, pp601-605 (2005)

[27] Pierre Baldi: Gradient Descent Learning Algorithm Overview: A General Dynamical Systems Perspective, *IEEE Trans. on Neural networks*, Vol. 6, NO. 1, pp182-195 (1995).

[28] Waymaere N., et al.: On the Initialization and Optimization of Multilayer Perceptrons, *IEEE Trans. on Neural Networks,* Vol.5, No.5, pp.738-751 (1994)

[29] Terrence L. Fine: Feedforward Neural Network Methodology, *Springer Press*, New York, 1999

[30] Gregory A. Baxes: Digital Image Processing Principles and Application, *John Wiley & Sons, Inc.* 1994

[31] Cornelins T. Leondes: Image Processing and Pattern recognition, *Academic Press*, San Diego, 1998

[32] Andrew R. Webb: Statistical Pattern Recognition (Second Edition), *John Wiley & Sons, LTI*, 2002

[33] Zhi-Quan Luo: On the Convergence of the LMS Algorithm with Adaptive Learning Rate for Linear Feedforward Networks, *Neural Computation*, Vol. 3, NO. 2, pp.226-245 (1991)

[34] E. J. Hartman and J. D. Keeler: Predicting the Future. Advantages of Semilocal Units, *Neural Computation*, Vol. 3, NO. 4, pp566-578 (1991)

[35] Park J., et al.: Approximations and RBF Networks. *Neural Computation*, Vol. 5, pp305-316 (1993)

[36] Park J., et al.: Universal Approximations Using RBF Networks. *Neural Computation*, Vol. 3, pp246-257 (1991)

[37] Eccles, N. J.: Neural Network for Banknote Recognition and authentication, *United States Patent*, EP0660276 (1995)

[38] Miroslaw K., Wlodzislaw D.: A Survey of Factors Influencing MLP Error Surface, *Control and Cybernetics*

[39] Park J., et al.: Approximation and Radial Basis function Networks. *Neural Computation*, Vol. 5, pp305-316 (1993)

[40] Bors A. G., et al.: Median Radial Basis Function neural network, *IEEE Trans. on Neural Networks,* Vol.7, No.6, pp.1351-1365 (1996)

[41] Elanayar S., et al.: Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic Systems. *IEEE Trans. on Neural Networks,* Vol.4, No.5, pp.594-604 (1994)

[42] Barron A. R.: Universal Approximation Bounds for Superpositions of a Sigmoidal Function, *IEEE Trans. on Information Theory,* Vol.39, No.3, pp.930-945 (1993)

[43] Sanger T. D.: Optimal Unsupervised Learning in a Sigle-layer Feedforward Neural Network, Neural Networks, Vol.1, pp.459-473 (1989)

[44] Bianchini, M., et al.: On the problem of local minima in recurrent neural networks, *IEEE Trans. on Neural Networks,* Vol.5, No.2, pp.738-751 (1994)

[45] Corwin E.M., et.al.: An Iterative Method for Training multiplayer networks with threshold functions, *IEEE Trans. on Neural Networks,* Vol.5, No.3, pp.507-508 (1994)

ACKNOWLEDGEMENTS

I would like to express my sincere appreciations to my supervisor Prof. Fumiaki Takeda for his careful guidance, continuous encouragements and serious supervising in the last three years. I profit significantly from his profound special knowledge and abundant practical experiences.

I also would like to express my sincere appreciations to leaderships and colleagues of Shenyang University of Technology who give so many support to me in last three years.

I will appreciate research assistants Nishikage Toshihiro, and Shirashi Yuhki, doctor candidates Lina Mi and Satoh Hironobu for their beneficial discussions and helps on my research. I also appreciate Saeki Yoshihiro for his kindness and many kinds of help.

I would like to thank secretaries Yoshida Naka and Yamanaka Miyuki for their helps and warm heart. I will further appreciate families of Yoshida Naka for their kindness and warm heart. I also appreciate all members of Takeda Laboratory. Just because of them I spend three unforgettable and happy years.

Especially, I will appreciate to my wife and my parents sincerely for their encouragements and sacrifice. I am so regret that I cannot take care of my parents in last three years.

I will appreciate Kochi University of Technology for giving us the special scholarship to complete my research smoothly.

PUBLICATIONS

International Conference Papers

1. Baiqing Sun, Fumiaki Takeda, Research on the Neural Network with RBF for Currency Recognition, *Proceeding of IASTED International Conference on Neural Networks and Computation Intelligence*, Switzerland, pp201-205, 2004;

2. Baiqing Sun, Fumiaki Takeda, Proposal of Neural Recognition with Gaussian Function and Discussion for Rejection Capabilities to Unknown Currencies, *Proceeding of International Conference on Knowledge-based Intelligent Information & Engineering Systems*, New Zealand, pp859-865, 2004;

3. Baiqing Sun, Fumiaki Takeda: Application of a Neural Network with Gaussian Function on Currency Recognition System, *Proceedings of SICE annual conference 2005,* Japan, pp601-605 (2005)

4. Fumiaki Takeda, Baiqing Sun, Discussion for Rejection Ability of Unknown Foreign Currencies by Gaussian Neural Recognition System, *Taiwan-Japan Symposium on Fuzzy systems and Innovational Computing*, Japan, 2004;

Journal Papers

1. Baiqing Sun, Fumiaki Takeda, A Paper Currency Recognition System based on Neural Networks with Gaussian Functions and an Optimizing Method for Its Parameters on Way to Learning, Trans. ISCIE Japan, is accepted.

2. Baiqing Sun, Fumiaki Takeda, Toshihiro Nishikage, Neural Recognition with Gaussian Function for Rejection of Unknown Foreign Currencies, Trans. IEE of Japan, was rejected.

3. Baiqing Sun, Fumiaki Takeda, Reaserch on Currency Recognition using Neural Networks with Gaussian Activation Function, journal of SICE Japan, was rejected after second review.

Domestic Conference Papers

1. Baiqing Sun, Fumiaki Takeda, Chinese Currency Recognition Using the Neural Network with RBF, *Proceeding of SICE System Integration Division Annual Conference (SI2003)*, Japan, pp476-477, 2003;