Application of RBFNN Heuristic Training Method in Human Brain CT Images

Zhou Jing1, Shao Hongbo1, Wang Ruihong2

1. College of Science, Agriculture University of Hebei, Baoding, China.
2. College of Information Science & Technology, Agriculture University of Hebei, Baoding, China.

Abstract: With the rapid development of computer and network technology, the pattern classification technology is widely applied in the field of artificial intelligence, this paper introduces the radial basis function neural network (RBFNN) and local generalization error model. Since the expressions of the model's results contain integral operation, so the computation is of high complexity. And we can only apply it in the RBFNN's structure selection problem under the condition that the characteristics dimension of the sample is large. In this paper, we give a heuristic method to training RBFNN based on local generalization error model, we can apply the method no matter when the characteristics dimension of the sample is large or not too large. At last, we apply the heuristic method in the brain CT image data, the experiment shows that computation time and the training precision are very satisfying.

Keywords: localized generalization error model (LGEM); radial basis function neural network (RBFNN); a greedy algorithm; pattern classification; computation complexity

1. Introduction

Pattern classification technology is widely applied in the field of artificial intelligence, which can be seen as a learning process. It summarizes and extracts information from the known sample to get the classifier. And we could predict the category of the unknown sample.

Among the pattern classification techniques, artificial neural network is the most commonly used method which is composed of simple signal processing units, which can be viewed as a parallel distribution information processor. RBF neural network is a multilayer type in multilayer feed-forward neural network. It has the ability to approximate any smooth input/output mapping. RBF neural network has the advantages of the fast training speed. More and more people pay attention to it and recognize it.

Generalization ability is the right ability of classification which is resulted from the training of neural network in the unknown sample. For the pattern classification problems, it is one of the most important indicator to evaluate the performance of neural network. The general learning algorithm target is to minimize the training error on the known sample set, but in fact, the ultimate goal of our training network is to classify the unknown sample. So how to minimize the generalization error should be our learning goals.

Neural network sensitivity is one of the important parameters of the neural network. The sensitivity of the neural network shows that the degree of the neural network output changes with the input changes. Its processing method is usually to input a small amount of disturbance into the network, and then measure the changes of the network output, so we can determine the output sensitivity to the input changes, thus to guide the network design, enhance the anti-jamming capability of the network and measure network's performance.

Local generalization error model was put forward by Wu Yongxian (Wing W.Y. Ng) et al., the model shows the relationship between trained classifier's local generalization ability and experience error and random input sensitivity. It gives classifier the upper bound of the generalization ability in the local domain. So how to better use the upper bound to design the classifier is a question worth discussing.

2. Rbfnn

2.1 Topological structure of the RBFNN

Radial basis function (RBF) neural network is composed of input layer, hidden layer (radial basis layer), and the linear output layer, its topology structure is as shown in Figure 1.

![Figure 1: Radial basis function (RBF) neural network](image)

When the hidden layer neurons is gaussian function, the neuron output can be expressed by the following mathematical form:

\[ h_i = \exp\left(-\frac{||x-c_i||^2}{2\sigma_i^2}\right), i = 1, 2, ..., K \]  

(1)

And \( j \) neuron's output layer of the radial basis function (RBF) neural network can be expressed as:
\[ f_i(x) = w_i h = \sum_{j=1}^{K} w_{ij} \exp \left( -\frac{||x - c_i||^2}{2\sigma^2_i} \right), \quad i = 1, 2, \ldots, K \text{ and } h = (h_1, h_2, \ldots, h_K) \]  

(2)

In order to get a specific RBFNN, we need to identify the parameters, such as: (1) The number of radial basis function is \( K \); (2) Each center of radial basis function is \( c_i \); (3) The width of the radial basis function is \( \sigma_i \); (4) The weights \( w_{ij} \).

2.2 RBF neural network training method

The number of radial basis function in the radial basis function (RBF) neural network is usually for sure. When the number is determined, we can use gradient descent algorithm for training to calculate the weights of the radial basis function network's hidden layer to output layer, each center of the radial basis function and the width of the radial basis function.

In the first stage, we get the parameter through training by gradient descent algorithm, but the gradient descent algorithm is easy to fall into local minimum points, and the training speed is slow. In 1989, the scholars Moody and Darken have given the two-phase training method. The method uses heuristic \( k \)-means clustering method to determine the center and the width of each radial basis function, and through the global optimal method, to determine the weight.

Based on the network resulted from the first two stages, we use gradient descent algorithm of training in the first phase and adjust all parameters of network at the same time, including center, breadth, and weight. We use the gradient descent algorithm at last, because in the first stage, various parameters are determined in a heuristic way, so the last step played a role of fine-tuning optimization.

3. A heuristic algorithm based on localized generalization error model to construct RBFNN

3.1 Localized generalization error model theory

The ultimate goal of pattern classification problems is to find \( f_\theta \), and to make it classify the unknown samples as much as possible, generalization ability can be presented through this equation:

\[ R_{gen} = \int_{T \setminus D} (f_\theta(x) - F(x))^2 p(x)dx \]  

(3)

In this equation, \( T \) stands for the whole space, \( x \) represents the vector quantity of the samples in the whole space \( T \setminus D \), \( p(x) \) stands for the probability density function of the sample \( x \) in the whole space.

Because the output of the objective function of unknown sample and its probability distribution in whole space is unknown, so the expression \( R_{gen} \) cannot be calculated directly. W.W.Y. Ng used the local generalization error model to solve the problem, he ignored the error caused by the unknown sample which is far away from the training sample, and then get the upper bound of local generalization error, and then use the classifier's upper bound of the generalization ability in the local domain as a criteria to evaluate classifier. For this, W.W.Y. Ng Q has given the concept of neighborhood.

[ definition: neighborhood \( Q_i \)]

Each sample point \( x_i \in D \) of training set \( D \) in the \( Q_i \) neighborhood \( S_Q(x_i) \) can be expressed as:

\[ S_Q(x_i) = \{ x \mid x = x_i + \Delta x; |\Delta x_i| \leq Q, k = 1 \ldots n \} \]  

(4)

\( Q \) is a given real Number, \( \Delta x = (\Delta x_1, ..., \Delta x_n)^T \) and \( H \) is the dimension of the sample space.

In the pattern classification problems, we usually do not have the prior knowledge of the samples distributing in the whole space. We can recognize that the unknown sample have the same chance to appear, so the appearing chances of \( \Delta x \) in the neighborhood \( S_Q(x_i) \) may be regarded as equally distributed random variable obeying to mean value of 0.

Then all the neighborhood \( Q_i \) and sets of the sample \( S_Q(x_i) \) consist the local domain, which can be represented as:

\[ S_Q = \bigcup_{i=1}^{N} S_Q(x_i) \]  

(5)

According to the concept of neighborhood \( Q_i \), we can get a local generalization error expressions:

\[ R_{SM}(Q) = \int_{S_Q} (f_\theta(x) - F(x))^2 p(x)dx \]  

(6)

Formula (6) expresses the error caused by the unknown sample which is near to the training sample in the whole space, which is shown in Figure 2. We only consider the error caused by the unknown sample in the shadow, without considering the error outside of the shadow, which is caused by the unknown sample far away from the shadow, so we can try to find the bound of local generalization error rather than looking for bound of the global generalization error.
The basic idea of the Local generalization error model is to define a $Q_i$ neighborhood $S_Q(x_i)$ for each sample $x_i$, we put all the sample of the neighborhood $Q$ and set $S_Q = \bigcup_{i=1}^{N} S_Q(x_i)$ as the local domain to discuss the classifier's generalization ability. W.W.Y. Ng has given the upper bound $R_{emp}(Q)$ of the $R_{emp}(Q)$, and use the upper bound as criteria to evaluate the classifiers. We think that the smaller the $R_{emp}(Q)$ is, correspondingly the classifier is better. Based on some assumptions, according to the Hoeffding inequality, we could get the following conclusions:

(7) \[
R_{emp}(Q) = \sqrt{R_{emp} + E_{emp}\left((\Delta y)^2\right)} + \epsilon
\]

\[
= \left(\frac{1}{N} \sum_{i=1}^{N} (f_{\theta}(x_i) - F(x_i))^2\right) + \epsilon
\]

\[
= \max \left[\frac{1}{N} \sum_{i=1}^{N} (f_{\theta}(x_i) - F(x_i))^2\right] + \epsilon
\]

\[
\Delta y = f_{\theta}(x) - f_{\theta}(x_{b})
\]

\[
R_{emp} = \frac{1}{N} \sum_{b=1}^{N} (f_{\theta}(x_b) - F(x_b))^2
\]

\[
E_{emp}(\Delta y)^2 = \frac{1}{N} \sum_{i=1}^{N} (f_{\theta}(x_i) - f_{\theta}(x_{b}))^2 \frac{1}{(2Q)^d} d\mathbf{x} = B \sqrt{\frac{\ln \eta}{-2N}}
\]

$A$ is the difference between maximum and minimum values outputed by the Objective function. We assume that the objective function’s output range is known or have been specified in advance.

$B = \max \left(\frac{f_{\theta}(x)}{F(x)}\right)^2, i = 1, \ldots, N$, $N$ is the number of training samples.

Under the conditions that the ranges output by objective function $F$ is for sure, and for a given set of training, $A$ and $\epsilon$ are constant.

In 2007, Wang and Wu put forward the unreasonable part of W.W.Y. Ng’s model, improved the proof of the model and revised the result of the model, thus making sure the conclusions of "local generalization error of the training function is related to empirical error and stochastic sensitivity".

Then we get:

\[
\int_{S_Q} \left\| f_{\theta}(x) - F(x) \right\| p(x) d\mathbf{x}
\]
In this one, $F(x)$ is approximating function, $f_\theta(x)$ is training function, $p(x)$ is probability density function in the whole pace, $P_{S_{\theta_i}}(x) = \int_{S_{\theta_i}} p(x) \, dx$ can be seen as probability density function in neighborhood $S_{\theta_i}(x_i)$, $w(x_i)$ stands for the real category of sample $x_i$, $\xi$ is controllable factor that controls the neighborhood $S_{\theta_i}(x_i)$ of each sample $x_i$. 

3.2. The analysis of the stochastic sensitivity in the model

Since local generalization error model has given upper bound of classifier's local generalization ability, thus it can be used to evaluate the generalization ability of classifier in the local domain, we think that the smaller the $R^{*}_{SM}(S_{Q})$ is, the classifier is better. There are three terms in the results of local generalization error, but because that the third terms is related to the approximating function, therefore in the application we can only calculate the sum of the first two terms, which are the experience error and random sensitivity. And therefore we can use the sum $\xi\left(R_{emp} + E_{\xi}(\|\Delta y\|)\right)$ of the first two terms to estimate the classifier, we think that the smaller the $\xi\left(R_{emp} + E_{\xi}(\|\Delta y\|)\right)$ is, the corresponding classifier is better. Since the neighborhood of the sample is fixed, $\xi$ is a fixed value, so we can take out and use $R_{emp} + E_{\xi}(\|\Delta y\|)$ to estimate the stand or fall of classifier.

We noticed that in the model the second terms is input stochastic sensitivity, and its calculation involved integral operation for each sample in neighborhood $Q$, and it is complex to calculate. In order to simplify the computational complexity, based on some assumptions, Dr Wu Yongxian made a reduction for the input random sensitivity in local generalization error model. One of the hypothesis is that: suppose the dimensions of the input features of RBFNN is large, then according to the central limit theorem, the output of the single RBFNN neuron obeys to the logarithmic normal distribution. So reduction method given by Dr Wu Yongxian is a problem under the assumption that the input characteristic dimension is large, however, in some practical problems, its dimensions of input characteristics is not large, such as the Iris database in UCI database, so we put forward heuristic algorithm based on local generalization error model to construct RBFNN, first of all, we analysis the second terms of the model.

Now we analyze the first item $R_{emp}$ and second item $E_{\xi}(\|\Delta y\|)$ of the model respectively, we first assume that $f_\theta(x)$ is instruction function (step function), as follows:

1) The first item $R_{emp}$ of the model $R_{emp}$ reflects the experience error, we put the number of the sample whose classifier output does not agree with the real output $N_{err}$, set $\|\cdot\|$ to norm2-, so when the output of the classifier is not agree with the real output, we get

$$\left|f_\theta(x_i) - w(x_i)\right| = \sqrt{2} \text{ and } \frac{1}{N} \sum_{i=1}^{N} \left|f_\theta(x_i) - w(x_i)\right| = \frac{1}{N} \sum_{i=1}^{N} \left|f_\theta(x_i) - w(x_i)\right| = \frac{\sqrt{2} N_{err}}{N}$$

Then we know that the size of the $N_{err}$ trained by classifier is related to all the samples, so $R_{emp}$ uses all the information of the samples to adjust the overall properties of the classifier.

2) The second terms $E_{\xi}(\|\Delta y\|)$ of the model to analyze the second terms $E_{\xi}(\|\Delta y\|)$ of the model, we can divide the samples into two parts to analyze.

We put the sample which is far from the boundary of the classifier as:

$$\Theta_{f_\theta} = \left\{ x_i | S_{\theta_i}(x_i) \in \Omega_{f_\theta}^+ \cap S_{\theta_i}(x_i) \in \Omega_{f_\theta}^- \land i \in \{1, \ldots, N\} \right\}$$

We put the sample which is near the boundary of the classifier as:
\[ \Phi_i = \left\{ x_i \mid S_{Q_i} (x_i) \in S_{\Omega} \& S_{Q_i} (x_i) \not\in \Omega_{Q_i} \& S_{Q_i} (x_i) \not\in \Omega_{Q_i} \& i \in \{1, \ldots, N\} \right\} \]  

(11)

In the following analysis process, we assume that the problems are two kinds of problems, which are positive and negative. Trained classifier \( f_{\theta}(x) \) makes \( \Omega_{Q_{\theta}} \)’s output is \( (1, 0) \) in space, and \( \Omega_{Q_{\theta}^-} = S_{Q_{\theta}} - \Omega_{Q_{\theta}^+} \)’s output in space is \( (0, 1) \), and among it, the \( S_{Q_{\theta}} \) is local domain.

As \( E_{S_{\theta}} (\|\Delta y\|) \) can be expressed in this way

\[ E_{S_{\theta}} (\|\Delta y\|) = \frac{1}{N} \sum_{i=1}^{N} \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

(12)

Now we calculate the result of a sample points \( x_i \)’s expression \( \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \). We set the neighborhood of sample points \( x_i \) as \( S_{Q_{\theta}} (x_i) \). If the \( x_i \)’s neighborhood \( S_{Q_{\theta}} (x_i) \) totally fall in positive class \( \Omega_{Q_{\theta}^+} \) or completely fall into negative class \( \Omega_{Q_{\theta}^-} \), then the classifier’s output for all unknown sample points in neighborhood \( S_{Q_{\theta}} (x_i) \) is same, so for any point in \( S_{Q_{\theta}} (x_i) \), there is \( \| f_{\theta}(x) - f_{\theta}(x) \| = 0 \) , so we get \( \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx = 0 \). If the \( x_i \)’s neighborhood \( S_{Q_{\theta}} (x_i) \) partly falls into positive class \( \Omega_{Q_{\theta}^+} \) or partly falls into negative class \( \Omega_{Q_{\theta}^-} \), which means that the Classification border going through the neighborhood \( S_{Q_{\theta}} (x_i) \), so we might as well suppose \( x_i \) that falls into the positive class \( \Omega_{Q_{\theta}^+} \), the corresponding output is \( f_{\theta}(x_i) = (1, 0) \) for the hypothesis of \( x_i \), that falls into the negative class, its conclusion is similar. So we get

\[ \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

\[ = \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx + \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

\[ = 0 + \sqrt{2} \int_{S_{\theta}(x_i)} \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

When we assume the \( p_{S_{\theta}(x_i)} (x) \) is in uniform distribution, we get

\[ = \sqrt{2} \left| S_{Q_{\theta}} (x_i) \right| \cdot p_{S_{\theta}(x_i)} (x) \]

(13)

\[ \left| S_{Q_{\theta}} (x_i) \right| \]

stands for the measure of a collection of points of which neighborhood \( S_{Q_{\theta}} (x_i) \) falls in the space \( \Omega_{Q_{\theta}^-} \), from the view of Euclidean space, it means volume, now we calculate

\[ E_{S_{\theta}} (\|\Delta y\|) = \frac{1}{N} \sum_{i=1}^{N} \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

We set the part that is different from the output result of sample \( x_i \) in \( S_{Q_{\theta}} (x_i) \) as \( S_{Q_{\theta}^*} (x_i) \) , \( p_{S_{\theta}(x_i)} (x) \) is uniform distribution, we get

\[ E_{S_{\theta}} (\|\Delta y\|) = \frac{1}{N} \sum_{i=1}^{N} \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

\[ = \frac{1}{N} \sum_{x_{i},x_{\not\in\Omega_{\theta}^+}} \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx + \frac{1}{N} \sum_{x_{i},x_{\not\in\Omega_{\theta}^-}} \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

\[ = \frac{1}{N} \sum_{x_{i},x_{\not\in\Omega_{\theta}^+}} \int_{S_{\theta}(x_i)} \| f_{\theta}(x) - f_{\theta}(x) \| \cdot p_{S_{\theta}(x_i)} (x) \, dx \]

\[ = \sqrt{2} \sum_{x_{i},x_{\not\in\Omega_{\theta}^+}} \left| S_{Q_{\theta}^*} (x_i) \right| \cdot p_{S_{\theta}(x_i)} (x) \]

(14)
From the above results, we can see that the $E_{\Delta y}$ is only related with the boundary sample, so we can say that when we commutating $E_{\Delta y}$, we use the information of the classification boundary sample, which can be used to adjust $E_{\Delta y}$. We can show it in this figure, as is shown in Figure 3, the classifier in the figure is the result by using all the information of the sample through first terms $R_{emp}$ of the model; As is shown in Figure 4, the classifier in the figure is the result by adjusting the second terms $E_{\Delta y}$ of the model, which is the result adjusted based on the classifier in the Figure 3.

![Figure 3 Using overall information](image1)

![Figure 4 Using marginal information](image2)

3.3 Heuristic algorithm to construct RBFNN

According to the above analysis, we can see the experience error could be used to adjust the overall performance, and stochastic sensitivity could be used to adjust the edge performance of the classifier. So we could train the classifier by heuristic method, the algorithm framework is as follows:

(1) Data preprocessing;
(2) Train classifier $f_\theta(x)$ according to $R_{emp}$;
(3) Adjust $E_{\Delta y}$ by $f_\theta(x)$ to find a boundary fault locations.

In order to better understand the boundary fault locations in step 3, we could get this definition from the view of mathematics. Now we first give the definition of sample boundary point, the definition is as follows:

**[Definition: the sample boundary point]**

$$S = \left\{ x \mid \min_{j=1,2,...,K} \left| f_{\theta}(x) - Label_j \right| > threshold, x \in TrainSet \right\} \quad (15)$$

This definition first give a *threshold* in advance, if the output difference between the sample and the recent categories is bigger than the threshold, then we think the sample is the sample boundary point.

Now we give the definition of the sample boundary fault locations:

**[Definition: the sample boundary fault point]**
\[ S = \left\{ x: \min_{i=1,2,\ldots,k} \left( |f_{\theta}(x) - \text{Label}_j| \right) > \text{threshold}, \right\} \]

On the basis of sample boundary point, with the condition \( |f_{\theta}(x) - y_i| \geq 0.5 \), that is to say, when the difference between sample output \( f_{\theta}(x) \) and real category \( y_i \) is big, we think the sample is classified wrongly. Figure 5 directly illustrates the sample boundary fault point. As is shown in the circle.

Now we illustrate the above algorithm by showing this figure, first of all, we get a classifier based on training \( R_{\text{emp}} \), as is shown in Figure 6, the boldest curve indicates the classifier, circle point stands for the sample boundary fault point.

Then we use these sample boundary fault point to adjust \( E_{x_i}(\Delta y) \) as shown in Figure 7,
We already know that we use $R_{\text{emp}} + E_{\theta_y}(\|\Delta y\|)$ to estimate the classifier, generally speaking, we believe that the smaller the $R_{\text{emp}} + E_{\theta_y}(\|\Delta y\|)$ is, the classifier is better. In our algorithm framework, we get a classifier with the standard $R_{\text{emp}}$ after the calculation in step 2, but this classifier may be bigger than $E_{\theta_y}(\|\Delta y\|)$, therefore in step 3, we adjust $E_{\theta_y}(\|\Delta y\|)$ based on the step 2. According to the previous analysis, the main method we adjust $E_{\theta_y}(\|\Delta y\|)$ is to adjust $f_\theta$, to make the sample be as far from $f_\theta$ as possible and fall on one side of $f_\theta$.

We applied the idea into RBFNN, which is to put the boundary fault point found by the $f_\theta(x)$ into the center to retrain the network. For training RBFNN is a fitting process in the space for space distribution based on RBF as the center. We specifically apply the above algorithm framework to RBFNN, algorithm process is as follows:

(0) Data preprocessing;

(1) Determine the clustering number for the center as the number of the known samples.

(2) According to $R_{\text{emp}}$ clustering algorithm, we can determine the category center and radius, we could get an initial RBFNN network based on $f_\theta$;

(3) We give threshold value and find a boundary fault locations according to the threshold values, then we add these boundary point as the center, and select the radius of the corresponding center by using the ways mentioned before, and determine the weight of RBFNN according to $R_{\text{emp}}$.

The step (3) in the above algorithm is a kind of way to adjust $E_{\theta_y}(\|\Delta y\|)$ by using the boundary fault locations.

### 3.4 The result and analysis of the experiment

First, we test on two simple database Toy1 and Toy2, we artificially produced two database whose feature number is 2 and category number is 2, the information of the two databases is given in Table 1. The training data are randomly chosen from the database Toy1 and Toy2, and the rest 50% of the data is test data, the threshold value is 0.35.

<table>
<thead>
<tr>
<th>Database</th>
<th>Number of the feature</th>
<th>Number of the sample</th>
<th>Number of category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy1</td>
<td>2</td>
<td>316</td>
<td>2</td>
</tr>
<tr>
<td>Toy2</td>
<td>2</td>
<td>754</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 8 gives the illustration of the experiment of database Toy1, the figure on the left shows the results of algorithm step (2), the figure on the right shows the results of algorithm step (3).

We can see from Figure 8 that the the result in database Toy1, the training accuracy is 1, its test precision in the experiment is also 1.
Figure 9 The Experiment of database toy2

We can see from Figure 9 that the result in database Toy2, the training accuracy is 1, its test precision in the experiment is 97.49%.

Secondly, we test on database UCI [16] with our algorithm. Information of the experiment selecting several UCI database is as shown in Table 2.

<table>
<thead>
<tr>
<th>Database</th>
<th>Number of the feature</th>
<th>Number of the sample</th>
<th>Number of category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>4</td>
<td>150</td>
<td>3</td>
</tr>
<tr>
<td>Pima Diabetes</td>
<td>8</td>
<td>768</td>
<td>2</td>
</tr>
<tr>
<td>Wine</td>
<td>13</td>
<td>178</td>
<td>3</td>
</tr>
</tbody>
</table>

The training data used in the test are randomly chosen from the database Toy1 and Toy2, and the rest 50% of the data is test data, the threshold value is 0.35. We repeated the test for certain times, the Table 3 shows the experimental results of the data set Iris, Wine, Pima, including the average training accuracy, the average test precision, the average number of center, the average time and the times that the experiment repeated on the database.

<table>
<thead>
<tr>
<th>Database</th>
<th>Average accuracy of training</th>
<th>Average accuracy of testing</th>
<th>Number of the average center</th>
<th>Average time</th>
<th>Test repeated times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>0.9771</td>
<td>0.9361</td>
<td>9.2800</td>
<td>0.0426s</td>
<td>100</td>
</tr>
<tr>
<td>Wine</td>
<td>0.9544</td>
<td>0.9064</td>
<td>10.1100</td>
<td>0.0679s</td>
<td>100</td>
</tr>
<tr>
<td>Pima</td>
<td>0.8612</td>
<td>0.7299</td>
<td>115.8000</td>
<td>3.4662s</td>
<td>10</td>
</tr>
</tbody>
</table>

4. Application of rbfnn heuristic training method in human brain ct images

We apply the RBFNN heuristic training method in the medical image database[17], the medical image database is the data extracted from analyzing the brain CT image, Table 4 shows the information of the the medical image database.

<table>
<thead>
<tr>
<th>Database</th>
<th>Number of the feature</th>
<th>Number of the sample</th>
<th>Number of the category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Toy1</td>
<td>11</td>
<td>212</td>
<td>2</td>
</tr>
</tbody>
</table>

The training data used in the experiment are randomly chosen from the medical image database, and the rest 50% of the data is test data, the threshold value is 0.35. We repeated the test for certain times, the Table 5 shows the experimental results on the medical image database, including the average training accuracy, the average test precision, the average number of center, the average time and the times that the experiment repeated on the database.

<table>
<thead>
<tr>
<th>Database</th>
<th>Average accuracy of training</th>
<th>Average accuracy of testing</th>
<th>Number of the average center</th>
<th>Average time</th>
<th>Test repeated times</th>
</tr>
</thead>
<tbody>
<tr>
<td>lwx</td>
<td>0.9651</td>
<td>0.9286</td>
<td>46.6400</td>
<td>0.3093s</td>
<td>100</td>
</tr>
</tbody>
</table>

In the 100 experiments, we find out the best result and worst result, as is shown in Table 6.

<table>
<thead>
<tr>
<th>Result</th>
<th>Training accuracy</th>
<th>Testing accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worst</td>
<td>0.9349</td>
<td>0.8333</td>
</tr>
<tr>
<td>Best</td>
<td>0.9763</td>
<td>1</td>
</tr>
</tbody>
</table>
We can see that there is a big gap between the best result and the worst result, this problem exists in the current algorithms method, which is that the result is not stable. By analyzing, the problem may be caused by the following reasons: training data for each experiment is randomly selected, chosen data in some cases distribute evenly and distribute unevenly in individual cases.

5. Conclusions and outlook

In this paper, based on the hypothesis of that classifier is indicating function, we analyze the first two local generalization error model, which are experience error term and stochastic sensitivity. This paper also gives the method of heuristic training RBFNN classifier, this method does not directly calculate a stochastic sensitivity, but try to reduce the sensitivity of the RBFNN in another way. Because this method avoids to calculate the integral operation of multifarious, the computing speed is fast.

From the experiment, we can see the result of applying this method on human brain CT image is also satisfying.

Besides, the results of the experiment show that our algorithm remains to be improved, which is that the number of the center of the network trained on the database is large. If we could find more representative sample points and add them to the new center, then this method can effectively avoid the large number of center. So how to find more representative sample points from the boundary fault points is a problem worth discussing.

References