Integration of Unascertained Method with Neural Networks and Its Application

Huawang Shi
Hebei University of Engineering, Handan, P. R. China
stone21st@163.com

Abstract—This paper presents the adoption of artificial neural network (ANN) model and Unascertained system to assist decision-makers in forecasting the early warning of financial in China. Artificial neural network (ANN) has outstanding characteristics in machine learning, fault, tolerant, parallel reasoning and processing nonlinear problem abilities. Unascertained system that imitates the human brain's thinking logical is a kind of mathematical tools used to deal with imprecise and uncertain knowledge. Integrating unascertained method with neural network technology, the reasoning process of network coding can be tracked, and the output of the network can be given a physical explanation. Application case shows that combines unascertained systems with feedforward artificial neural networks can obtain more reasonable and more advantage of nonlinear mapping that can handle more complete type of data.

Index Terms—artificial neural network, unascertained system, financial early warning

I. INTRODUCTION

Unascertained system that imitates the human brain's thinking logical is a kind of mathematical tools used to deal with imprecise and uncertain knowledge. Artificial neural network that imitates the function of human neurons may function as a general estimator, mapping the relationship between input and output. Combination of these two methods can take into account the effect of complementary effect of each other! Our theoretical analyses are the following aspects: First, the artificial neural network is a nonlinear mapping from input to output; it does not rely on any mathematical model. Unascertained system also as a nonlinear mapping is to convert input signals x in domain U into signal y in domain V as output. Second, artificial neural networks can only deal with explicit data classification, and not suitable for the expression of a rule-based knowledge. However unascertained systems can handle abnormal, incomplete and uncertain data. Third, the artificial neural network's knowledge representation and treatment are simple in form, and hard to the introduction of heuristic knowledge, and the lower efficiency of the network. Unascertained system can make use of expertise knowledge, thus be easy to introduce of heuristic knowledge that making the reasoning process more reasonable. Finally, artificial neural network's greatest strength are memory, learning and inductive functions; Unascertained system does not have the learning function.

So, in theory, combining unascertained systems with feed forward artificial neural networks can obtain more reasonable and more advantage of nonlinear mapping that can handle more complete and comprehensive type of data.

The rest of this paper is organized as follows: Unascertained Number and Algorithm are described in Section2. Section3 describes Unascertained BP Neural Networks in detail and gives Network Learning Process. The experimental results on Unascertained BP Neural Networks and some discussions are presented in Section4. Finally, Section5 provides the conclusion.

II. MATERIALS AND METHODS

A. Introduction to Unascertained Number:

1) Definition of Unascertained number:

Unascertained mathematics, proposed by Want [1], is a tool to describe subjective uncertainty quantitatively. It deals mainly with unascertained information, which differs from stochastic information, fuzzy information, and grey information. Unascertained information refers to the information demanded by decision-making over which the message itself has no uncertainty but, because of situation constraints, the decision-make cannot grasp the whole information needed. Hence, all systems containing the behavior factors, such as the problem of clustering have unascertained property.

Definition 1: Suppose a is arbitrary real number, 0 < α ≤ 1, then define \([a, a]_\alpha(x)\) is first-order unascertained number, where

\[
\phi(x)=\begin{cases} a, x = a \\ 0, x \neq a \cup x \in R \end{cases}
\]

(1)

Note that \([a, a]\) express the interval of value, and \(\phi(x)=\alpha\) express belief degree of a. When \(\alpha=1\), belief degree of a is 1. Where \(\alpha=0\), belief degree of a is zero.

Definition 1: Suppose \([a, b]\) is arbitrary closed interval, \(a = x_1 < x_2 < \cdots < x_n = b\), if

\[
\phi(x) = \begin{cases} a_i, x = x_i (i = 1, 2, \cdots, n) \\ 0, \text{other} \end{cases}
\]

(2)

and \(\sum a_i = a\), 0 < α ≤ 1, then \([a, b]\) and \(\phi(x)\) compose a n -order unascertained number, as follow \([a, b], \phi(x)\).
where $\alpha$ is total degree belief, $[a,b]$ is the interval of value, is $\varphi(x)$ the density function.

Definition 1: Suppose unascertained number is $A = \{x_1, x_2\} \varphi(x)$, where

$$\varphi(x) = \begin{cases} \alpha, x = x_i (i = 1, 2, \ldots, k) \\ 0, \text{other} \end{cases}$$

(3)

$$0 < \alpha_i < 1, i = 1, 2, \ldots, k$$

Then first-order unascertained number:

$$E(A) = \left[ \frac{1}{\alpha} \sum_{i=1}^{k} x_{i} \alpha_i, \frac{1}{\alpha} \sum_{i=1}^{k} x_{i} \alpha_i \right], \varphi(x)$$

$$\varphi(x) = \begin{cases} \alpha, x = x_i (i = 1, 2, \ldots, k) \\ 0, \text{other} \end{cases}$$

(4)

It is expected value of unascertained number $A$. When $\alpha = 1$, as $E(A)$, unascertained number $A$ is discrete type random variable. When $\alpha < 1$, $E(A)$ is first-order unascertained number. Where $$\frac{1}{\alpha} \sum_{i=1}^{k} x_{i} \alpha_i$$ is expected value of $A$ that belief degree is $\alpha$.

2) Algorithm of unascertained number:

Each unascertained number includes two parts of probable value and belief degree. So, unascertained number algorithm also includes two parts. Suppose unascertained numbers are $A$ and $B$. Where

$$A = f(x) = \begin{cases} \alpha, x = x_i (i = 1, 2, \ldots, m) \\ 0, \text{other} \end{cases}$$

$$B = g(x) = \begin{cases} \beta, y = y_j (j = 1, 2, \ldots, n) \\ 0, \text{other} \end{cases}$$

(5)

$$C = A \times B$$ also is unascertained number. Probable value and belief degree of $C$ is calculated as follows.

(1) Constitute multiply matrix of probable value of unascertained number $A$ and $B$, where individual is probable value number series $x_1, x_2, \ldots, x_k$ and $y_1, y_2, \ldots, y_m$ as $A$ and $B$, permute from little to big.

(2) Constitute multiply matrix of belief degree of unascertained number $A$ and $B$, where individual is belief degree number series $\alpha_1, \alpha_2, \ldots, \alpha_n$ and $\beta_1, \beta_2, \ldots, \beta_n$ are $A$ and $B$. Suppose $a_j$ and $b_j$ individual is element of multiply matrix of probable value of $A$ and $B$, here $i$ is line of matrix, $j$ is array of matrix. We called $a_j$ and $b_j$ as relevant position element.

(3) $\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_k$ result from multiply matrix of probable value of unascertained number $A$ and $B$, which permute from little to big. And an equal element is one element of relevant position element in multiply matrix of belief degree. Suppose $\bar{r}_1, \bar{r}_2, \ldots, \bar{r}_k$ is relevant position element permutation. Where

$$C = \varphi(x) = \begin{cases} \bar{r}_i, x = \bar{r}_i (i = 1, 2, \ldots, k) \\ 0, \text{other} \end{cases}$$

(6)

Suppose $C = \varphi(x)$ is arithmetic product of unascertained number $A$ and $B$. Where

$$C = A \times B = f(x) \times g(x) = \begin{cases} \bar{r}_i, x = \bar{r}_i (i = 1, 2, \ldots, k) \\ 0, \text{other} \end{cases}$$

(7)

3) Unascertained membership:

Using Unascertained to describe "uncertain" or "unclear boundary" phenomenon, the key problem is that a reasonable Unascertained membership function. Despite the clear definition rules of the construction unascertained measure, the definition is non-structural in nature, and did not give a specific construction method. It still needs to be in accordance with the background knowledge in specific areas, known to the measured data and personal experience of decision-makers, etc.

Under normal circumstances, decision-makers do not know exactly state of membership function. At this point, the simplest and most reasonable method is by fitting line shape of membership function. A standard membership functions is in Figure 1.

![Figure 1. A standard membership functions curve](image)

The class $I_1$ membership function $\varphi_1(x)$ was expressed by the broken line $AC_1DI$; The class $I_2$ membership function $\varphi_2(x)$ expressed by the broken line $OCC_2EI$; The class $I_3$ membership function $\varphi_3(x)$ was expressed by the broken line $ODC_3FI$; The class $I_4$ of membership function $\varphi_4(x)$ was expressed by the broken line $OEC_4GI$; The class $I_5$ of membership function $\varphi_5(x)$ was expressed by the broken line $OFC_5B$

B. Introduction to ANN

Artificial Neural Networks (ANNs) are composed of simple elements that imitate the biological nervous systems. In the last few decades, significant research has been reported in the field of ANNs and the proposed ANN architectures have proven the inefficiency in various applications in the field of engineering. The structure of a neural network of most commonly used type is schematically shown in Fig.1. It consists of several layers of processing units (also termed neurons, nodes). The input values are processed within the individual neurons of the input layer and then the output values of these neurons are forwarded to the neurons in the hidden layer. Each connection has an associated
parameter indicating the strength of this connection, these called weight.

![Diagram of feedforward networks](image)

Figure 1. The single layer of feedforward networks.

The NN model frequently used is multilayer perceptron learning with error back-propagation. In the present research work, the sequence with which the input vectors occur for the ANN training is not taken into account, thus they are static networks that propagate the values to the layers in a feed-forward way. The training of the neural networks is performed through a back-propagation algorithm. In General, the back-propagation algorithm is a gradient-descent algorithm in which the network weights are moved along the negative of the gradient of the performance function.

Artificial Neural Network (ANN) is basically as implied model of the biological neuron and uses an approach similar to human brain to make decisions and to arrive at conclusions[7]. Every neuron model consists of a processing element with synaptic input connections and a single output. The structure of a neural network of most commonly used type is schematically shown in figure 1.

![Neural model](image)

Figure 2. Neural model.

The neuron can be defined as

\[ y = f(W \times X + \theta) = f(\sum_{j=1}^{n} w_{ij} x_{i} - \theta_{j}) \]

where, \( X \) is input signals, \( \omega_{ij} \) is synaptic weights of neuron, \( f \) is the activation function and \( y \) is the output signal of neuron. The architecture of multi-layered feedforward neural network is shown in Fig. 2.

It consists of one input layer, one output layer and hidden layer. It may have one or more hidden layers. All layers are fully connected and of the feedforward type. The outputs are nonlinear function of inputs, and are controlled by weights that are computed during learning process.

At present, the BP neural network is one of the most matures, wide spread artificial neural network. Its basic network is three-layer feed-forward neural network such as input layer, hidden layer, and output layer. The input signals must firstly disseminate for- ward into the hidden node. The output information of the concealment node transmits into output node Via- function action. Finally the output variable result is obtained. The BP network can realize complex non-linear mapping relations will fully from input to output and has good exuding ability, which can complete the duty of complex pattern recognition.

ANN has outstanding characteristics in machine learning, fault, tolerant, parallel reasoning and processing nonlinear problem abilities. It offers significant support in terms of organizing, classifying, and summarizing data. It also helps to discern patterns among input data, requires few ones, and achieves a high degree of prediction accuracy. These characteristics make neural network technology a potentially promising alternative tool for recognition, classification, and forecasting in the area of construction, in terms of accuracy, adaptability, robustness, effectiveness, and efficiency. Therefore, cost application areas that require prediction could be implemented by ANN.

### C. Unascertained BP Neural Network

1) **Description of unascertained BP network:**

Assuming there is \( N \) known samples, divided into \( K \) categories, \( X^{k} \) represents the \( k \) th sample space with the sample size for \( N_{k} \), apparently: \( \sum_{k=1}^{K} N_{k} = N \).

\( x_{i}^{k} \) represents \( i \) th sample \(( 1 \leq i \leq N_{k} )\), so \( X^{k} = \{ x_{1}^{(k)}, \cdots, x_{i}^{(k)}, \cdots, x_{N_{k}}^{(k)} \}^{T} \). Suppose that each sample \( x_{i}^{k} \) has \( J \) characteristics (or indicators), the \( j \) th feature (or indicators) is \( I_{j}^{k} \), \( 1 \leq j \leq J \).
\( x_{ij} \) represents the observation value of sample \( x_i \) with reference to the \( j \) th characteristic (or indicator).

2) **Unascertained BP neural network structure:**
Unascertained BP neural network structure and the structure of BP neural network is basically the same seen in Fig.1.

![Unascertained BP network structure](image)

The first layer is input layer, the number of nodes is the same of feature space dimension. The second layer is hidden layer. The third layer is output layer; the layer number of nodes is equal to the classification number.

3) **The desired membership calculating method:**
In the usual artificial neural network, training samples were divided into specific categories, that is, a sample is determined belonging to a category. Therefore, training in the network, its corresponding output node of the desired output as "1", and the rest of the output node of the desired output for the "0". However, in practice, data are often sick, and its classification border is not very specific, and samples are belonging to categories in certain degree of membership. Therefore, the desired output is not simply a two-valued logic, need to calculate exactly, which leads to uncertainty in the network.

As the input data may be numerical value, also possible be the degree of membership, the corresponding desired output, there are differences in the calculation. The following discussion is made under numerical value input:

Supposing there are \( N_k \left( \sum_{k=1}^{K} N_k = N \right) \) samples in the \( k \) th category and category center is \( O_k \):

\[
O_k = (O_{1k}^1, \cdots, O_{Jk}^j, \cdots, O_{yk}^J)^T \quad (1 \leq j \leq J, 1 \leq k \leq K)
\]

(8)

Where, \( O_j^k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_{ij} \)

\[
\overline{O}_j = \frac{1}{K} \sum_{k=1}^{K} O_j^k = (\overline{O}_1^1, \overline{O}_2^j, \cdots, \overline{O}_J^j)^T
\]

(9)

\[
\sigma_j^2 = \frac{1}{K} \sum_{k=1}^{K} (O_j^k - \overline{O}_j)^2
\]

(10)

\[
w_j = \frac{\sigma_j^2}{\sum_{j=1}^{J} \sigma_j^2}
\]

(11)

Obviously, \( 0 \leq w_j \leq 1 \),and \( \sum_{j=1}^{J} w_j = 1 \). Therefore, \( w_j \) is the indicator \( j \) ’classification weight of given classification.

Set \( x_i = (x_{i1}, \cdots, x_{ij}, \cdots, x_{iy}) \) \((1 \leq i \leq N)\) as any training samples.

When the larger \( \| x_i - O_k \| \), the farther sample \( x_i \) away from the center of \( k \) th category and its membership belonging to the \( k \) th category be smaller. On the other hand, when \( \| x_i - O_k \| \) the smaller, the nearer sample \( x_i \) away from the center of \( k \) th category and its membership belonging to the \( k \) th category be larger.

When the larger \( w_j \), the greater the contribution to classification of indicator \( I_j \), that is, the more important to classification of indicator \( I_j \) On the other hand, when the smaller \( w_j \), it shows that the smaller the contribution to classification of indicator \( I_j \), that is, the less important for classification of indicators \( I_j \). From the above, we can define the weighted distance of sample \( x_i \) to the \( k \) th class center \( O_k \):

\[
\gamma_{ik} = \sum_{j=1}^{J} w_j (x_{ij} - O_j^k)^2
\]

(12)

\[
\mu_k(x_i) = \frac{1}{\gamma_{ik} + \varepsilon} \left( \sum_{k=1}^{K} \frac{1}{\gamma_{ik} + \varepsilon} \right)
\]

(13)

Obviously, \( 0 \leq \mu_k(x_i) \leq 1 \), \( \sum_{k=1}^{K} \mu_k(x_i) = 1 \).

Therefore, as \( \mu_k(x_i) \) is unascertained membership of sample belonging to the \( k \) th category, that is, it is the expectations output of membership degree that we have to calculate: \( d_{ik} = \mu_k(x_i) \)

4) **Mathematical derivation of amendment \( w_j \):**
Supposing that \( O_j^i \) represent output of the \( j \) th node, \( O_i \) express the output of \( i \) th node of the relative former layer and \( O_k \) express output of the \( k \) th node of the relative behind layer \( w_j \) express connection weights of the upper layer node \( i \) to this node \( j \):
\[
\text{net}_j = \sum_i w_{ij} \cdot O_i
\]

\[
O_j = f(\text{net}_j) = \frac{1}{1 + e^{-\text{net}_j}} \quad (14)
\]

Where, \(\text{net}_j\) express the net input of nodes \(j\).

- When \(O_j\) is the output of output layer nodes, the actual output \(y_j = O_j\). Set \(d_j\) is the desired output of node \(j\): \(d_j = \mu^k(x_j)\), then squares sum error of output are as follows:

\[
E_j = \frac{1}{2} \sum_j (y_j - d_j)^2 = \frac{1}{2} \sum_j (O_j - d_j)^2 \quad (15)
\]

The total output error is:

\[
E = \sum_i E_i \quad (16)
\]

Considering amendments to the weights:

\[
\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial w_{ij}} = (O_j - d_j) \cdot [O_j \cdot (1 - O_j)] \cdot O_i \quad (17)
\]

\[
\delta_j = (O_j - d_j) \cdot O_j \cdot (1 - O_j) \quad (18)
\]

- When \(O_j\) is the output of hidden layer nodes, \(O_j\) affects each node of the lower classes.

Output square error:

\[
E = \frac{1}{2} \sum_k (y_k - d_k)^2
\]

The actual output of \(k\) th node of the output layer:

\[
y_k = f(\text{net}_k) = \frac{1}{1 + e^{-\text{net}_k}} \quad (19)
\]

\[
\text{net}_k = \sum_j w_{jk} \cdot O_j \quad (20)
\]

\[
O_j = f(\text{net}_j) = \frac{1}{1 + e^{-\text{net}_j}} \quad (21)
\]

\[
\text{net}_j = \sum_i w_{ij} \cdot O_i
\]

Considering amendments to the weights:

\[
\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial \text{net}_k} \cdot \frac{\partial \text{net}_k}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial w_{ij}} = \sum_k (y_k - d_k) \cdot y_k \cdot (1 - y_k) \cdot w_{jk} \cdot O_j \cdot (1 - O_j) \cdot O_i
\]

Set \(\delta_k = (y_k - d_k) \cdot y_k \cdot (1 - y_k)\)

Then

\[
\frac{\partial E}{\partial w_{ij}} = \sum_k \delta_k \cdot w_{jk} \cdot O_j \cdot (1 - O_j) \cdot O_i
\]

### 5) Network learning process:

Set counter \(t\), and \(t = 0\), randomly generated initial values of weights \(w_{ij}(t)\), set learning rate \(\eta\), the system error \(\varepsilon\) as well as the impulse factor \(\alpha\), set the maximum number of iterations \(T\).

Enter the study samples \(X\), and calculate the desired output membership \(d_k\) of sample \(X\), then squares sum error of output are as follows:

\[
E_j = \frac{1}{2} \sum_j (y_j - d_j)^2 = \frac{1}{2} \sum_j (O_j - d_j)^2 \quad (15)
\]

The total output error is:

\[
E = \sum_i E_i \quad (16)
\]

Considering amendments to the weights:

\[
\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial O_j} \cdot \frac{\partial O_j}{\partial \text{net}_j} \cdot \frac{\partial \text{net}_j}{\partial w_{ij}} = (O_j - d_j) \cdot [O_j \cdot (1 - O_j)] \cdot O_i \quad (17)
\]

\[
\delta_j = (O_j - d_j) \cdot O_j \cdot (1 - O_j) \quad (18)
\]

Output layer:

\[
\delta_j(t) = (O_j(t) - d_j) \cdot O_j(t) \cdot (1 - O_j(t))
\]

Hidden layer:

\[
\delta_j(t) = \sum_k \delta_k(t) \cdot w_{jk}(t) \cdot O_j(t) \cdot (1 - O_j(t))
\]

Where, \(k\) express the lower node number to node \(j\).

Calculate the Adjustment value of weights:

\[
\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) + \eta \delta_j(t) \cdot O_i(t)
\]

Revisit weights:

\[
w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t)
\]

\(t = t+1\), turn to (3).

### 6) Network identification:

Suppose \(x\) is the sample to be recognized. Input \(x\) into the trained network. Suppose the greatest output is of the \(k_0\) output node, \(x\) belongs to the \(k_0\) th category is determined.

\[
k_0 = \max \{ \mu^k(x) \mid k = 1, 2, \ldots, K \} \quad (19)
\]

### D. Unascertained RBF Neural Networks

1) Structure of unascertained RBF network:

Unascertained RBF network consists of three layers such as input layer, hidden layer and output layer, which neurons in same layers has no connection, and between the adjacent two-layers has fully connected. Number of input layer neurons is the sample dimension; hidden layer

© 2011 ACADEMY PUBLISHER
and output layer neuron number are the classification number of samples. Unascertained RBF network is characterized by only one hidden layer; hidden layer neurons nodes select the Gaussian function to have a non-linear mapping of input and output, layer neurons are linear combine node. Its structure is shown in Fig.2.

\[ m_k = (m_{k1}, m_{k2}, \ldots, m_{kd})^T \]  

(22)

Unascertained classification in accordance with the point of view, give a classification, first of all concerned are in a given category, the characteristics of the classification of all make a little contribution, and contribution to the value of quantitative calculation. Hereinafter referred to as "normalized" after the classification of the characteristics of the contribution value of the characteristics regarding the classification of the classification weights. And, in the calculation of the sample about when various types of membership, in essence, to use a variety of characteristics of the weight classification.

In order to quantitatively describe the contributions of \( d \) characteristics to the initial classification.

\[ m = \frac{1}{C} \sum_{k=1}^{C} m_k \]  

(23)

Let \( \bar{m} = (\bar{m}_1, \bar{m}_2, \ldots, \bar{m}_d) \)

Let \( \sigma_j^2 = \frac{1}{C} \sum_{k=1}^{C} (m_{kj} - \bar{m}_j)^2 \), \( 1 \leq j \leq d \),

(24)

The size of variance \( \sigma_j^2 \) reflects the extent of discrete the type of \( K \) centers as \( m_1, m_2, \ldots, m_K \) in the first feature on values.

Let \( \sum_{j=1}^{d} w_j = 1 \) satisfied : \( 0 \leq w_j \leq 1 \) and \( \sum_{j=1}^{d} w_j = 1 \).

Then, \( w_j \) is called the classification weights of \( j \) characteristics under a given classification conditions.

Let \( \rho_{ik} = \sum_{j=1}^{d} w_j (x_{ij} - m_{kj})^2 \)

(26)

Where: \( \delta \) is non-negative real number, usually taken as \( \delta = 0.001 \sim 0.01 \).

In (26), if \( w_j = 0 \), it is illustrated that the characteristic \( j \) has no contribution of distinction between \( K \) categories, so \( j \) should not appear in the calculation of the weight in the distance.

Thus, we can calculate the possibility of some measure that the sample \( x_i \) belonging to the \( k \) th category as follows:

\[ \mu_{ik} = \mu(x_i \in C_k) = \frac{1}{\rho_{ik} + \varepsilon} \left( \sum_{k=1}^{K} \frac{1}{\rho_{ik} + \varepsilon} \right) \]  

(27)

Where, \( \varepsilon = 0.01 \sim 0.001 \),

2) The desired membership calculating method:

Unascertained RBF neural network in the desired output method of calculating degree of membership.

Given known \( n \) samples, each of the known samples \( x_i \) are point of \( d \) dimensional feature space, that is:

\[ x_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \]

The \( n \) samples are divided into \( K \) categories: \( C_1, C_2, \ldots, C_K \), \( m_k \) is the category center vector of \( C_k \) \( (k = 1, 2, \ldots, K) \). Considering the same type of sample point should be in-dimensional feature space with each other more "close" is reasonable. We have assumed that the "close" is the Euclidean distance proximity.

Supposing the \( i \) th training sample is \( x_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \), the \( j \) th \( (1 \leq j \leq d) \) data is \( x_{ij} \) that is the nominal quantity of data. Supposing \( m_k \) classified Center Vector of \( C_k \):

\[ m_k = \left( m_{k1}, m_{k2}, \ldots, m_{kd} \right)^T \]

(22)

Suppose the input sample was \( x \), then the output of the \( i \) th hidden layer nodes was as follows:

\[ \varphi_i = \exp \left( -\frac{\left\| x - m_i \right\|^2}{\nu_i^2} \right) \]  

(20)

Where \( \left\| \cdot \right\| \) is European norm, \( m_i \) and \( \nu_i \) were the centers and width of the \( i \) th hidden layer units of RBF.

The \( j \) th neuron actual output of output layer is:

\[ y_j = \sum_{i=1}^{K} w_{ij} \varphi_i \]  

(21)

\[ w_{ij} = 1 \]

Compared with BP neural network, RBF network many has quicker convergence speed, because \( \bar{m}_i \) has a larger output value, far away from \( \bar{m}_i \), and its output decreases rapidly.

Figure 5. The unascertained RBF network structure.

\[ \varphi_i = \exp \left( -\frac{\left\| x - m_i \right\|^2}{\nu_i^2} \right) \]  

(20)
Obviously, \( 0 \leq \mu_k \leq 1 \) and \( \sum_{k=1}^{K} \mu_k = 1 \), therefore, \( \mu_k \) known as the unascertained measure of samples \( x_i \) belonging to the \( k \) th category, that is, we want to calculate the expectations output membership degree \( d_k \), then \( d_k = \mu_k \).

3) Unascertained RBF neural Network Learning Process:

RBF network has been used for the study, it is to classify the \( N \) known samples in \( K \) categories, and to determine the classification of unknown samples \( x \). Unascertained RBF network is not only stress the desired output 1 or 0, but also required specific calculation, and the rest are same with BP networks. Unascertained RBF network parameters need to learn there are three: the center of basis function and the variance as well as the weights of hidden layer to output layer connection. The learning steps are as follows:

Center adjust: Unascertained-means clustering algorithm.

Given classification number \( K \) and the system accuracy \( \varepsilon_1 \), set counter \( t = 0 \),

Give the initial classification of \( n \) samples, get \( K \) cluster center vector \( \mathbf{m}_k(t), (k = 1, 2, \ldots, n) \),

Calculating the unascertained measure \( \mu_k(t), t = 1 \sim n, k = 1 \sim K \) of samples \( x_i \) belonged to the \( k \) th category,

Determine a new type of center vector \( \mathbf{m}_k(t+1) \) from \( \mu_k \) as follows:

\[
m_k(t+1) = \sum_{i=1}^{n} \mu_k \cdot x_i / \sum_{i=1}^{n} \mu_k \tag{28}
\]

Calculate \( \text{err} = \sum_{k=1}^{K} \| m_k(t+1) - m_k(t) \| \), and if \( \text{err} \leq \varepsilon_1 \), so, stop iteration and turn to \( f \); Otherwise, let \( t = t + 1 \), turn to (c)

Recalculate unascertained measure of the sample \( x_i \) belonging to the \( k \) th category,

Determining the variance: In the center adjustment process, the variance \( \sigma_k \) is determined by (28).

The study of connection weights

Supposing \( \varphi_{ij}, (i = 1 \sim K) \) is the output of the \( i \) th neuron months in hidden layer, \( y_j, (j = 1 \sim K) \) is the actual output of the \( j \) neurons in output layer, \( d_j \) is the expectations corresponding output. Then, \( y_j = \sum_{i=1}^{K} w_{ij} \cdot \varphi_{i}, (j = 1 \sim K) \).

The output layer error is:

\[
E = \frac{1}{2} \sum_{j=1}^{K} (d_j - y_j)^2 \tag{29}
\]

\[
\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \cdot \frac{\partial y_j}{\partial w_{ij}} = - (d_j - y_j) \varphi_i \tag{30}
\]

\[
\Delta w_{ij} = - \eta \frac{\partial E}{\partial w_{ij}} = \eta (d_j - y_j) \varphi_i \tag{31}
\]

Weight correction formula is as follows:

\[
w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \tag{32}
\]

4) Unascertained RBF neural network learning process:

To identify samples \( x \), input \( x \) to the trained network, supposing the greatest output is of the \( k_0 \) th output node, then \( x \) belongs to the \( k_0 \) th category.

Recognition Criteria:

\[
k_0 = \max_k \{ \mu^k(x) \mid k = 1, 2, \cdots, K \} \tag{33}
\]

It's said that the sample \( x \) belongs to the \( k_0 \) th category.

E. Application case

In a market economy, enterprises are faced with a wide variety of risks. Therefore the establishment of a sound and effective financial risk early warning system is of great necessity to the monitoring and control of financial risk [11, 12]. We put the 45 selected sample data divided into training samples and test samples (30 as training samples, 15 as test samples) into unascertained neural network system. There were 15 nodes which value affect financial risk[11,12] is to input into neural network , 13 nodes in the hidden layer, and 1 node that indict the output value (‘1’ represents safety and ‘0’ represents unsaved) of the risk in the output layer.

The learning rate was 0.01, and expectative error was 0.001.Then the neural network was programmed by software Matlab7.1. The training results are shown in Table 1. The network structure is 15x13x1. The average variance EMS was 2.343 11×10-5, and training time was 54 second. Trained 2386 times, reaching the goal, training completed, the network convergence, when the total error is 0.000996. Re-enter the training samples to the best network training network detection, error rate to 0, and the network fitting fitting rate of 99.8%. Samples will be entered into the prediction network prediction, prediction results were shown in Table 1.

F. Conclusions

Comparing Table 1 with the sample data, there is only one sample of mistake. Therefore the misjudgment rate is 6.67%, that is the correct identification rate is 93.33%.

From this example, we can see unascertained neural network for classification has a high application value. So, not only in theory but also in practice, combining unascertained systems with feedforward artificial neural
networks can obtain more reasonable and more advantage of nonlinear mapping that can handle more complete type of and comprehensive data.

Comparing Table 2 with the sample data, there were no samples of mistake. Therefore, the misjudgment rate is 0, that is, the correct identification rate is 100%.

From this example, we can see unascertained RBF neural network for classification has a high application value. So, not only in theory but also in practice, combining unascertained systems with RBF artificial neural networks can obtain more reasonable and more advantage of nonlinear mapping that can handle more complete type of and comprehensive data.

REFERENCES


