Abstract—In this paper two models using a functional network were employed to solving classification problem. Functional networks are generalized neural networks, which permit the specification of their initial topology using knowledge about the problem at hand. In this case, and after analyzing the available data and their relations, we systematically discuss a numerical analysis method used for functional network, and apply two functional network models to solving XOR problem. The XOR problem that cannot be solved with two-layered neural network can be solved by two-layered functional network, which reveals a potent computational power of functional networks, and the performance of the proposed model was validated using classification problems.

Keywords—Functional network, neural network, XOR problem, classification, numerical analysis method.

I. INTRODUCTION

CASTILLO et al. [1] present functional networks as an extension of artificial neural networks (ANNs). Unlike neural networks, in these networks there are no weights associated with the links connecting neurons, and the internal neuron functions are not fixed but learnable. These functions are not arbitrary, but subject to strong constraints to satisfy the compatibility conditions imposed by the existence of multiple links going from the last input layer to the same output units. In fact, writing the values of the output units in different forms. By considering these different links, a system of functional equations is obtained. When this system is solved, the number of degrees of freedom of these initially multidimensional functions is considerably reduced. To learn the resulting functions, a method based on minimizing a least squares error function is used, which, unlike the functions used in neural networks, has a single minimum.

Castillo et al. [2] resumes all the functional network models that can be applied. Include time series, chaotic series, differential equations, regression problems, etc. these works show that functional networks can indeed have lots of applications. We have been keeping an eye on the developing of applying functional networks, This paper seeks another application of functional networks, i.e., discuss how to use functional network models to perform some classification problems.

This paper is organized as follows. In Section 2, an element of the functional network is introduced. In Section 3, a framework for solving typical of classification problems with functional networks models is proposed. In Section 4, two functional function models for solving classification problem are discussed. In Section 5, conclusions this paper.

II. ELEMENTS OF A FUNCTIONAL NETWORK

A functional network [1] consists of the following elements (see Fig.1):

1. A layer of input storing units. This layer contains the input data. Input units are also represented by small black circle with their corresponding names (x1, x2 and x3 in Fig.1).

2. A layer of output storing units. This layer contains the output data. Output units are also be represented by small circles with their corresponding names (x6 in Fig.1).

3. One or several layers of processing units. These units evaluate asset of input values, coming from the previous layer (of intermediate or input units) and delivers a set of output values to the next layer (of intermediate or output units). To this end, each neuron has associated a neuron function, which can be multivariate, and can have as many arguments as inputs. Each component of a neural function is called a functional cell. For example, the functional networks in Fig.1 have 3 neurons f1, f2, f3.

4. None, one or several layers of intermediate storing units. These layers contain units store intermediate information produced by neuron units. Intermediate units are represented by small circle (x4, x5 in Fig.1).

5. A set directed links. They connect units in the input or intermediate layers to neuron units, and neuron units to intermediate or output units. Arrows represent connections, indicate the information flow direction.
Our definition is simple but rigorous: a functional network is a network in which the weights of the neurons are substituted by a set of functions. Some have the it’s advantages include the following issues:

(1) Unlike ANNs, functional networks can reproduce certain physical characteristics that lead to the corresponding network in a natural way. However, reproduction only takes place if we use an expression with a physical meaning inside the functions database, and we do not dispose of that kind of information, this particular advantage does not apply in our case.

(2) The estimation of the network parameters can be obtained by resolving a linear system of equations. It is a fast and unique solution, and the global minimum of an error function.

III. FUNCTIONAL NETWORK LEARNING MODELS FOR SOLVING CLASSIFICATION PROBLEM

In this section, we present some typical of functional network models for solving classify problem (See Fig.2).

![Fig.2 A functional network for the solving classify problems model](image)

How can we train the network? Consider the following equivalence:

\[ z_j = f_1^{-1}(f_i(x_i), f_j(x_j)) \]

\[ j = 1, 2, \ldots, n \] Training patterns. The error can be measured as follow:

\[ e_j = f_i(x_i) f_j(x_j) \Rightarrow f_j(x_i) f_j(x_j) \]

\[ j = 1, 2, \ldots, n \] Training patterns. The error can be measured as follows:

\[ e_j = f_i(x_i) f_j(x_j) - f_j(x_j) \]

\[ j = 1, 2, \ldots, n \] Training patterns.

We now approximate the functions \( \{f_i | i = 1, 2, 3\} \) by using a linear combination of the functions of a given family \( \Phi_s = \{\phi_{s1}, \phi_{s2}, \ldots, \phi_{sm}\}, s = 1, 2, 3 \), in other words:

\[ f_s(x) = \sum_{i=1}^{m} a_{si} \phi_{si}(x), s = 1, 2, 3. \]

The formula of the sum of the squares of the errors, with \( z = x_3 \), is:

\[ Q = \sum_{j=1}^{n} e_j^2 \]

\[ = \sum_{j=1}^{n} \left( \sum_{i=1}^{m} a_{si} \phi_{si}(x_i) \right) \left( \sum_{i=1}^{m} a_{sj} \phi_{sj}(x_j) \right) - \sum_{i=1}^{m} a_{si} \phi_{si}(x_i) \right)^2 \]

IV. FUNCTIONAL NETWORK MODELS FOR SOLVING XOR PROBLEM

In this section, we objective are to using functional networks model for solving XOR problem. Minsky and Papert [4] clarified the limitations of two-layered neural networks (i.e., no hidden layers): in a large number of interesting cases, the two-layered neural network is incapable of solving the problem. A classic example of this case the exclusive-or (XOR) problem, which has a long history in the study of neural networks and many other difficult problems involve the XOR as sub-problem. Rumelhart et al. [5] showed that the three-layered neural network (i.e., with one hidden layer) can solve such problems, include the XOR problem, and the interesting internal representations can be constructed in the weight-space.

As described above, the XOR problem cannot be solved with the two-layered neural network. We using two function network models for solving XOR problem as follows:

A. The 2 Element XOR Classic Functional Network Model

These Problems can be successfully solved by two-layered functional network (see Fig.3).

![Fig.3 A functional network for the solving two-element XOR problems](image)

The input-output mapping in the XOR problem is shown in Table I.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>0</td>
</tr>
</tbody>
</table>

Suppose that we use the family of functions \( \Phi_s = (\phi_{s1}, \phi_{s2}, x) = (1, x); s = 1, 2 \). (6) Thus, the Fig.3 functional network output is

\[ z = f_1(x_1) f_2(x_2) \] (7)

Let

\[ f_1(x_1) = a_1 + a_2 x_1, f_2(x_2) = b_1 + b_2 x_2 \] (8)

Where \( a_1, a_2, b_1 \) and \( b_2 \) are arbitrary real number. Substituting
In (8), we get
\[ z = f_1(x_1)f_2(x_2) = c_0 + c_1x_1 + c_2x_2 + c_3x_1x_2 \]  \hfill (9)
Where \( c_i \in R, i = 0,1,2,3 \). We use Fig. 3 functional network model and equation (9) simulation XOR function.

Let \( c_0 = 1, c_1 = c_2 = 1, c_3 = -2 \), in equation (9). Thus, we obtained
\[ z = -1 + x_1 + x_2 - 2x_1x_2 \]  \hfill (10)
To simulation XOR function, and let
\[
\begin{align*}
x_1' &= x_1' \cos 45^\circ + x_1' \sin 45^\circ \\
x_2' &= x_2' \sin 45^\circ - x_2' \cos 45^\circ 
\end{align*}
\]  \hfill (11)
Therefore, equation (10) can be rewritten into:
\[
-\left( \frac{x_1' - \sqrt{2}/2}{\sqrt{2}/2} \right)^2 + \left( \frac{x_2' + \sqrt{2}/2}{\sqrt{2}/2} \right)^2 = 1 
\]  \hfill (12)
In fact, equation (12) is a hyperbola (see Fig 4.). Which divides the input space into two sections \{ (0,0), (1,1) \} and \{ (1,0), (0,1) \}.

\[
\begin{align*}
\Phi_3 &= (\phi_1(x),\phi_2(x),\phi_3(x)) = (1,x,x^2) ; s = 1,2,3. 
\end{align*}
\]  \hfill (13)
In this case, we shall obtain
\[
z = f_1(x_1)f_2(x_2) = d_0 + d_1x_1 + d_2x_2 + d_3x_1x_2 + d_4x_1^2 + d_5x_2^2 + d_6x_1x_2^2 + d_7x_1x_2 
\]  \hfill (14)
Let \( d_6 = d_7 = 0 \), other coefficients are arbitrary chosen. Similarly, Which divides the input space into two sections \{ (0,0), (1,1) \} and \{ (1,0), (0,1) \}. (See Fig.5).

\[
\begin{align*}
\Phi_4 &= (\phi_1(x),\phi_2(x),\phi_3(x)) = (1,x,x^2) ; s = 1,2,3. 
\end{align*}
\]  \hfill (15)
Where \( x = [x_1,x_2]^T, x_i = [x_{i1},x_{i2}]^T \). Then, the neuron function is represented as
\[
K(x_i,x) = 1 + x_1'x_1 + 2x_1x_2x_{i1}x_{i2} + x_1'^2 + 2x_{i1} + 2x_{i2} 
\]  \hfill (16)
And, the neuron functions \( K(x_i,x) \) to obtain the nonlinear mapping
\[
\phi(x) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]^T 
\]  \hfill (17)
Similarly, we can obtain
\[
\phi(x) = [1, x_1^2, \sqrt{2}x_1x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]^T, i = 1,2,3,4 
\]  \hfill (18)
Suppose that we use the family of functions in Fig.6 \( \Phi_i = (\phi_i(x), \phi_i(x)) ; i = 1,2,3,4 \). Then, the functional network output
\[
z = K(x_i,x) = \phi(x) \cdot \phi(x) 
\]  \hfill (19)
Where \( \cdot \) represent the dot product which gives the family of functions used to simulate the data in Table II. We can obtain
Using the Lagrange multipliers we define the auxiliary function

\[ Q(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}(9\alpha_1^2 - 2\alpha_2\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4 + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_2^2 - 2\alpha_3\alpha_4 + 9\alpha_2^2) \]  

(21)

The minimum can be obtained by solving the following system of linear equations:

\[
\begin{align*}
9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 &= 1 \\
-\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 &= 1 \\
-\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 &= 1 \\
\alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 &= 1
\end{align*}
\]

(22)

Then, we obtain

\[ \alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8} \]

Which show that

\[ Q_{\text{min}}(\alpha) = \frac{1}{4} \]

(23)

If the two classes are linearly separable, then one can find an optimal factional parameter vector (equivalently weight vector) \( w^* \) such that \( \|w^*\|^2 \) is minimum; and

\[ w^* \cdot x_j \geq 1, \quad \text{if } y_j = 1 \]

\[ w^* \cdot x_j \geq -1, \quad \text{if } y_j = -1 \]

In this example, we obtain

\[ \frac{1}{2} \|w^*\|^2 = \frac{1}{4} \]

And optimal factional parameter

\[ w^* = \frac{1}{8}([-\varphi(x_1) + \varphi(x_2) + \varphi(x_3) - \varphi(x_4)]) \]

(24)

In fact, equation (26) also is a hyperbola (similarly to Fig. 4). Which divides the input space into two sections \( \{(-1,-1), (+1,+1)\} \) and \( \{(-1,+1), (+1,-1)\} \).

V. CONCLUSIONS

Functional networks, as opposed to neural networks, learn the neural the sigmoid functions instead of link weights. In this paper, we use functional network technique to solving classification problem. The XOR problem that cannot be solved with two-layered neural network can be solved by two-layered functional network, which reveals a potent computational power of functional neural nets. These models are of the properties such as easily trained and simply structured. Finally, given examples show that the proposed model is effective and practical.

REFERENCES


Yong-Quan Zhou:Ph.D., professor. His research interests include neural network, computation intelligent with application and computer mathematics.