# Application of Functional Network to Solving Classification Problems

Yong-Quan Zhou, Deng-Xu He, and Zheng Nong

*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

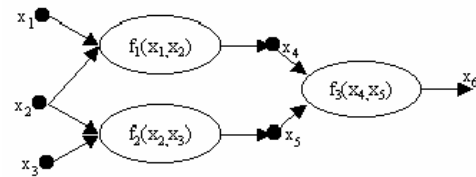A functional network [1] consists of the following elements (seen Fig.1):



Fig. 1 A functional network model

(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 ( $x_1, x_2$ and $x_3$ 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 ( $x_6$ 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 $f_1, f_2, f_3$.

(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 ( $x_4, x_5$ 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.

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:12, 2007

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 (Seen Fig.2).
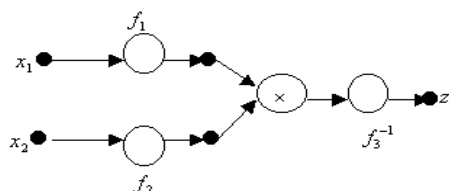


Fig. 2 A functional network for the solving classify problems model

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

$$z_j = f_3^{-1}(f_1(x_{1j})f_2(x_{2j})) \tag{1}$$

$j = 1,2,...,n$ Training patterns. The error can be measured as follow:

$$z_j = f_1(x_{1j})f_2(x_{2j}) \Leftrightarrow f_3(z_j) = f_1(x_{1j})f_2(x_{2j}) \tag{2}$$

$j = 1,2,...,n$ Training patterns. The error can be measured as follows:

$$e_j = f_1(x_{1j})f_2(x_{2j}) - f_3(z_j) \tag{3}$$

$j = 1,2,...,n$ Training patterns.

We now approximate the functions $\{f_i \mid i = 1,2,3\}$ by using a linear combination of the functions of a given family $\Phi_s = \{\varphi_{s1}, \varphi_{s2},...,\varphi_{sm}\}, s = 1,2,3$. in other words:

$$f_s(x) = \sum_{i=1}^{m} a_{si}\varphi_{si}(x), s = 1,2,3. \tag{4}$$

The formula of the sum of the squares of the errors, with $z = x_3$, is:

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

$$= \sum_{j=1}^{n} [(\sum_{i=1}^{m} a_{1i}\varphi_{1i}(x_{1j}))(\sum_{i=1}^{m} a_{2i}\varphi_{2i}(x_{2j})) - \sum_{i=1}^{m} a_{3i}\varphi_{3i}(x_{3j})]^2 \tag{5}$$

Our objective is to calculate the coefficients $a_{si}$. In next Section, two functional network modes for solving XOR Problem is proposed.

### 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 (seen Fig.3).
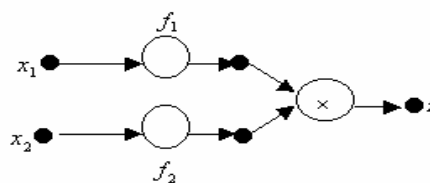


Fig.3 A functional network for the solving two-element XOR problems

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

TABLE I
THE XOR PROBLEM

| Input | | Output |
|---|---|---|
| $x_1$ | $x_2$ | $z$ |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Suppose that we use the family of functions

$$\Phi_s = (\varphi_{s1}(x), \varphi_{s2}(x)) = (1, x); s = 1,2. \tag{6}$$

Thus, the Fig .3 functional network output is

$$z = f_1(x_1)f_2(x_2) \tag{7}$$

Let

$$f_1(x_1) = a_1 + a_2 x_1, \ f_2(x_2) = b_1 + b_2 x_2 \tag{8}$$

Where $a_1, a_2, b_1$ and $b_2$ are arbitrary real number. Substituting

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:12, 2007

(7) in (8), we get

$$z = f_1(x_1)f_2(x_2) = c_0 + c_1 x_1 + c_2 x_2 + c_3 x_1 x_2 \qquad (9)$$

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

$$z = -1 + x_1 + x_2 - 2x_1 x_2 \qquad (10)$$

To simulation XOR function, and let

$$\begin{cases} x_1 = x_1' \cos 45^0 + x_2' \sin 45^0 \\ x_2 = x_1' \sin 45^0 + x_2' \sin 45^0 \end{cases} \qquad (11)$$

Therefore, equation (10) can be rewritten into:

$$-\frac{(x_1' - \sqrt{2}/2)^2}{(\sqrt{2}/2)^2} + \frac{(x_2')^2}{(\frac{\sqrt{2}}{2})^2} = 1 \qquad (12)$$

In fact, equation (12) is a hyperbola (seen Fig 4.). Which divides the input space into two sections $\{(0,0),(1,1)\}$ and $\{(1,0),(0,1)\}$.
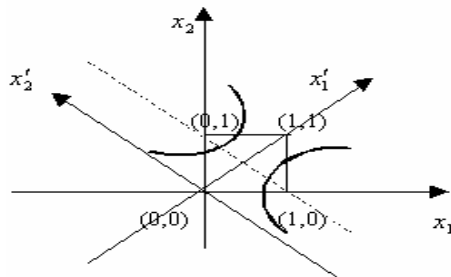


Fig. 4 The hyperbola function for solving XOR problem

In the previous examples we have considered one order polynomials family of functions for solving XOR problem. However, we can also consider any other high order polynomials family of functions such as

$$\Phi_s = (\varphi_{s1}(x), \varphi_{s2}(x), \varphi_{s3}(x)) = (1, x, x^2); s = 1,2,3. \qquad (13)$$

In this case, we shall obtain

$$\begin{aligned} z &= f_1(x_1)f_2(x_2) \\ &= d_0 + d_1 x_1 + d_2 x_2 + d_3 x_1 x_2 + d_4 x_1^2 + d_5 x_2^2 + d_6 x_1 x_2^2 + d_7 x_1^2 x_2 \end{aligned} \qquad (14)$$

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 $\{(1,0),(0,1)\}$. (Seen Fig.5).
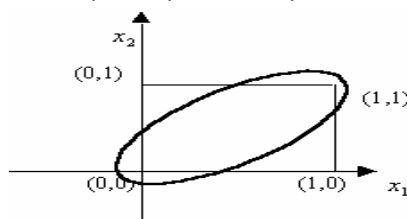


Fig. 5 The high order polynomial functions for solving XOR problem

*B. The n Element XOR Classic Functional Network Model.*

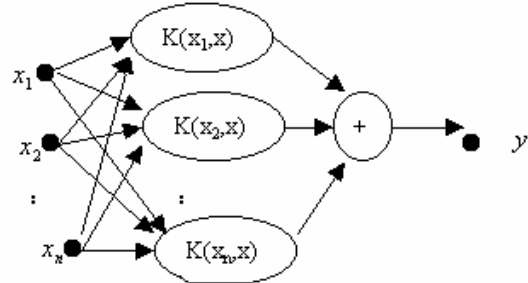This case can be successfully solved by two-layered functional network (seen Fig .6).



Fig. 6 n-element XOR classic functional network model

Where $K(x_i, x), i = 1,2,...,n;$ as functional network neuron function for resolving two-class problem is show in Table II.

TABLE II
A TWO-CLASS CLASSIFICATION PROBLEM

| Input vector $X$ | output $d$ |
|---|---|
| $(-1,-1)$ | $-1$ |
| $(-1,+1)$ | $+1$ |
| $(+1,-1)$ | $+1$ |
| $(+1,+1)$ | $-1$ |

We use a functional network model in Fig. 6 with neuron function as follows:

$$K(x, x_i) = (1 + x^T x_i)^2 \qquad (15)$$

Where $x = [x_1, x_2]^T, x_i = [x_{i1}, x_{i2}]^T$. Then, the neuron function is represented as

$$K(x,x_i) = 1 + x_1^2 x_{i1}^2 + 2x_1 x_2 x_{i1} x_{i2} + x_2^2 x_{i2}^2 + 2x_1 x_{i1} + 2x_2 x_{i2} \qquad (16)$$

And, the neuron functions $K(x, x_i)$ to obtain the nonlinear mapping

$$\varphi(x) = [1, x_1^2, \sqrt{2}x_1 x_2, x_2^2, \sqrt{2}x_1, \sqrt{2}x_2]^T \qquad (17)$$

Similarly, we can obtain

$$\varphi(x_i) = [1, x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}]^T, i = 1,2,3,4; \qquad (18)$$

Suppose that we use the family of functions in Fig.6

$$\Phi_i = (\varphi(x), \varphi(x_i)); i = 1,2,3,4. \qquad (19)$$

Then, the functional network output

$$z = K(x, x_i) = \varphi(x) \bullet \varphi(x_i) \qquad (20)$$

Where $\bullet$ represent the dot product which gives the family of functions used to simulate the data in Table II. We can obtain

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:1, No:12, 2007

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

Using the Lagrange multipliers we define the auxiliary function

$$Q(\alpha) =$$

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 + 2\alpha_1\alpha_4$$

$$+ 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4 + 9\alpha_4^2) \quad (21)$$

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

$$\begin{cases} 9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1 \\ -\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1 \\ -\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1 \\ \alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1 \end{cases} \quad (22)$$

Then, we obtain

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}$$

Which show that

$$Q_{\min}(\alpha) = \frac{1}{4} \quad (23)$$

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

$$w^\bullet \bullet x_i - b \geq 1, \qquad \text{If } y_i = 1$$

$$w^\bullet \bullet x_i - b \leq 1, \qquad \text{If } y_i = -1$$

In this example, we obtain

$$\frac{1}{2} \| w^\bullet \|^2 = \frac{1}{4}.$$

And optimal factional parameter

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

$$= \left[ -\begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{bmatrix} \right] = \begin{bmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (24)$$

Where $b = 0$, then, we obtain optimal equation

$$w^{\bullet T}\varphi(x) = 0 \quad (25)$$

Substituting equation (25) in equation (16) and (23), we obtain

$$-x_1x_2 = 0 \quad (26)$$

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

[1] Enrique Castillo. Functional Networks. Neural Processing Letters, 1998,7:151~159
[2] Enrique Castillo, Angel Cobo, Jose Manuel Gutierrez. Functional Networks with Applications. Kluwer Academic Publishers, 1999
[3] Enrique Castillo, Gutierrez JM, Cobo A, Castillo C. A minimax method for learning functional networks. Neural Processing Letters, 2000, 11(1): 39~49
[4] Minsky, M, L., & Papert, S.A. Perceptrons. Cambridge, MA: MIT Press, 1969
[5] Rumelhart, D.E., Hinton, G.E., & Williams, R.J. Parallel distributed processing (Vol.1). Cambridge, MA: MIT Press, 1986a.
[6] James Tin-Yau. Moderating the outputs of support vector machine classifiers. IEEE Transactions On Neural Networks, 1999,10(5): 1018-1031
[7] Alfonso Iglesias, Bernardino Arcay, J, M. Cotos, J. A. Taboada, Carlos Dafonte. A comparison between functional networks and artificial neural networks for the prediction of fishing catches. Neural comput & applie. 2004,13:24~31
[8] Igor T. Podolak. Functional graph model of a neural network. IEEE Transactions On System, Man, And Cybernetics-Part B: Cybernetics. 1998,28(6): 876-881
[9] Yong-Quan ZHOU. Polynomial Function Recurrent Neural Networks Model and Apply. Chinese Journal of Computers, (in Chinese), 2003,26(9): 1196-1200.

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