Expansion Type Functional Neuron Network Model and Its Parameters to Directly Determine the Method

Peigang Guo[1] Yongquan Zhou[1]
[1]College of Information Science and Engineering, Guangxi University for Nationalities, Nanning, 530006, China
Email: peigang1015@126.com

Qian Xiao[2]
[2]School of Computer and Electronics Information, Guangxi University, Nanning 530004, China
Email: yongquanzhou@126.com

***Abstract*—In this paper, functional neuron structure will be expanded and deformed in the functional networks, then the expansion type functional neuron network model is obtained, and the pinv Moore-Penrose of matrix is employed, the optimal value of the functional parameters in the functional networks can be rapidly and accurately determined directly. Finally simulation experiments show that the proposed method is feasible and effective, can obtain higher approximation accuracy.**

***Index Terms*—Functional Network; Functional Neuron; Expansion Type Functional Network; Basis Function; Matrix pinv Moore-Penrose**

I. Introduction

Castillo introduced the functional network in 1998[1]. It is an extension of the standard neural network. Unlike neural networks, it deals with general functional models instead of sigmoid-like ones, in these networks there are no weights associated with the links connecting neurons, and the neural functions are unknown from given families to be estimated during the learning process. We can select appropriate families for each specific problem (such as polynomials, Fourier expansions and trigonometric functions, etc.). At present, the functional network is a very useful general framework for solving a wide range of problems: The solving of differential functional and difference equation [2], nonlinear time series and prediction modeling [3], factorization model of multivariate polynomials [4], the identification of nonlinear system [5], CAD, linear and nonlinear regression [4], etc. The functional networks have shown excellent performance in the above-mentioned problems.

Functional network achieved a greater success in the application, but its theoretical basis is imperfect, greatly limits the scope of application of the functional network. The key theories include: first, what type of network structure and the family of basis function, often by experts to make judgments based on empirical knowledge; second, the network structure and functional parameters, this point is often difficult to determine by artificial; third, the network structure is determined, also exist the local minima problem. At present, there are some using genetic programming [6] to solve the above problems, and have achieve good results; but the family of basis function how to select, the theory has not yet been given a general method.

This paper will attempt according to the structural characteristics of the functional networks, and functional neuron structure will be expanded and deformed in the functional networks, then the expansion type functional neuron network model is obtained, and the pinv Moore-Penrose of matrix is employed, the optimal value of the functional parameters in the functional networks can be rapidly and accurately determined directly. Finally, the numerical simulation results show that the proposed method is feasible and effective, and can obtain higher approximation accuracy.

The rest of the paper is organized as follows: Section 2 introduces the basic knowledge of the functional networks. The expansion type functional network is proposed in Section 3. The expansion type functional network parameters to directly determine the method is given in Section 4. The experimental results and analysis are summarized in Section 5, and Section 6 presents conclusions.

II. Functional Networks

A. Functional Neuron

The neurobiology research showed that the different regions of the brain of higher vertebrates have different functions (i.e. different regions have different local function). When handling the senior perception behavior, some local function in the brain is organically integrated to accomplish this task. For functional networks, each node of the output can be regarded as a local function; the output of each output node can be viewed as the

Manuscript received Dec. 18, 2012; revised June 1, 2012; accepted July 1, 2009.
Corresponding author: Yongquan Zhou
Email:yongquanzhou@126.com

doi:10.4304/jsw.8.2.443-450

functional networks to some local function integrated to accomplish a specific task.

Thus, Castillo Enrique proposed the functional neuron model in 1998, its neuron activation function is not fixed, but adjustable, each neuron is represented as a linear combination of the known functions of a given family, and according to different problem of background knowledge to select the different function clusters, which can achieve the senior perception behavior function of the strength of the points. In functional networks, commonly used the family of basic functions: polynomial basis functions: $\{1, x, x^2, x^3, \ldots\}$; exponential basis functions: $\{1, e^x, e^{2x}, e^{3x}, \ldots\}$; Fourier basis functions: $\{1, \sin x, \cos x, \sin 2x, \cos 2x, \ldots\}$, etc.

The functional neuron model is shown in Fig.1:

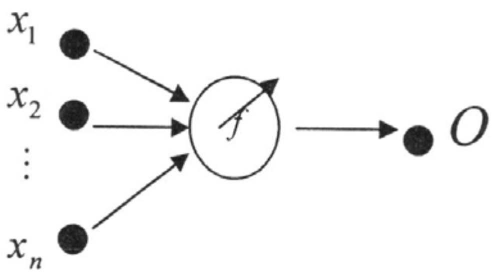

Figure 1. A functional neuron model.

In the functional neuron model shown in Fig.1:

$$o = f(X). \tag{1}$$

Where, o is the output of the neuron, $f(\bullet)$ is the function of neuron, $X = (x_1, x_2, \ldots, x_n)^T$ and o respectively represent the function of the input and output of the neuron (which can be one-dimensional or multi-dimensional vector). It follow from that learning of the neuron is equivalent to learning functional neuron function $f(\bullet)$. Under normal circumstances, the neuron function is represented as a linear combination of family of basic functions, i.e.

$$f(X) = \sum_{i=1}^{n} a_i \phi_i(X). \tag{2}$$

Where, $\{\phi_i(X) | i = 1,2,\ldots,n\}$ is the family of basic functions, for different problems to be solved in the specific by the background, often choose different family of basic functions; $a_i (i = 1,2,\ldots,n)$ is the functional parameters, and is obtained using the least squares method, the steepest descent method.

B. The Functional Networks Model

In general, a functional network consists of the following elements:

1. One layer of input storing units. This layer contains the input data. Input units are represented by small black circles with their corresponding names.

2. One layer of output storing units. This layer contains the output data. Output units are also represented by small black circles with their corresponding names.

3. One or several layers of processing units. These units evaluate a set of input values, coming from the previous layer (of intermediate or input units) and deliver 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 (univariate) of a neural function is called a functional cell. Neurons are represented by ovals with the name of the corresponding function inside. For example, assume that we have a neuron with s inputs $(x_1, x_2, \ldots, x_s)$ and k outputs $y_1, y_2, \ldots, y_k$, then, we assume that there exist k functions f_j, $j = 1,2,\ldots,k$, such that $y_j = f_j(x_1, x_2, \ldots, x_s)$. The functions f_j are not arbitrary, but determined by the structure of the network, as we shall see later. Neurons are represented by circles with the name of the corresponding f_j function inside.

4. None, one or several layers of intermediate storing units. These layers contain units that store intermediate information produced by neuron units. Intermediate units are represented by small black circles. These layers allow forcing the outputs of processing units to be coincident.

5. A set of directed links. They connect the input layer to the first layer of neurons, neurons of one layer to neurons of the next layer, and the last layer of neurons to the output units. Connections are represented by arrows, indicating the information flow direction. All these elements together form the network architecture, which defines the functional capabilities of the network. Network architecture refers to the organization of the neurons and the connections involved. In multilayer networks, units are organized in series of layers. Information flows in only one direction, from the input layer to the output layer. Neuron units receive information only from previous layers of the network, and output information to the next layer of neurons, or to the output units.

In Ref. [3] one example of a simple functional network is given in Fig.2, and its corresponding neural network architecture is also given in Fig.3.

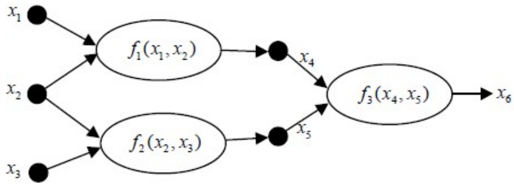

Figure 3. A typical functional network topology model.

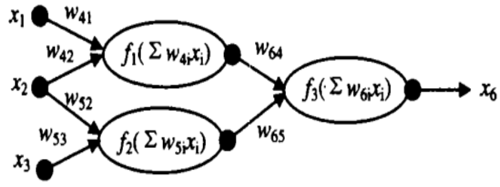

Figure 2. And Fig.2 equivalent neural network.

In Fig.2, where the input layer consists of the units $\{x_1, x_2, x_3\}$, the first layer of neurons contains neurons f_1 and f_2, the second layer of neurons contains neurons f_3, and the output layer reduces to the units x_6. One of the most important is the choice of neurons function $f_i(i=1,2,3)$. According to Castillo's approach will each neuron functions $f_i(i=1,2,3)$ is represented as a linear combination of the known functions of a given family. Such as, polynomials, trigonometric functions, Fourier expansion etc. In the neural network, each neuron function f_i, that is, the activation functions often take the Sigma function, hyperbolic tangent function, etc. In standard neural networks the neuron functions f_i are fixed, and some weights associated with the links or connections have to be learned. However, in functional networks there are no weights, and the neuron functions f_i must be learned. As for the other differences with the neural network, this paper will not repeat them, in Ref. [4]

III. Expansion Type Functional Networks

A. *Expansion Type Functional Neuron Model*

In functional networks, each functional neurons function can be represented as a linear combination of the family of basis functions. Therefore, functional neuron can be expanded by this feature of functional neuron in functional networks, and thus functional neuron model corresponding to the expansion model can be obtained. Without loss of generality, we can use the Eq. (2) represented by the functional neuron model (in Fig.4) to expand, thereby the corresponding functional neuron expansion model can be obtained and shown in Fig.5.

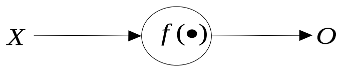

Figure 5. The functional neuron model.

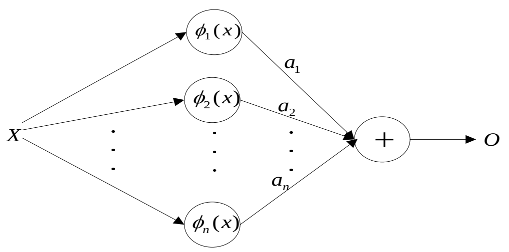

Figure 4. The expansion type functional neuron model.

In Fig.5, X denotes functional neuron function of input, $\phi_i(x),(i=1,2,\ldots,n)$ are given sets of given linearly independent functions, called basis functions; $a_i,(i=1,2,\ldots,n)$ are functional parameters; O denotes functional neuron of output.

B. *The Expansion Type associativity Functional Networks Model*

In a multitude of the functional networks model, the generalized associativity functional networks model has a wide range of applications [6] [12-14]. The following, we take for example the associativity functional networks, the expansion type associativity functional networks model is given. The associativity functional networks network topology is shown in Fig.6, the output of the corresponding expression is the Eq. (3), the output is the input of the function, $f_s(s=1,2,3)$ are a linear combination of the given basis function clusters,

$$z = f_3^{-1}(f_1(x) + f_2(y)) \tag{3}$$

We can based on the given sample data through training, it is determined to satisfy the accuracy requirements of the neuron function f_s.

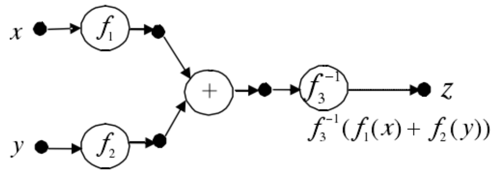

Figure 6. The associativity functional network model.

The Eq. (3) is deformed to obtain:

$$f_3(z) = f_1(x) + f_2(y) \tag{4}$$

Further can be deformed as follows:

$$O = f_1(x) + f_2(y) - f_3(z) \tag{5}$$

Then that is associativity expansion type functional network model can be denoted by the Eq. (5). In Eq. (5), O indicates the network the output, $f_s(x), s=1,2,3$ is represented as a linear combination of the known functions of a given family.

According to Fig.6 above, the single functional neuron model is expanded, thus the associativity functional network model can be converted into the expansion type associativity functional network model, as shown in Fig.7.

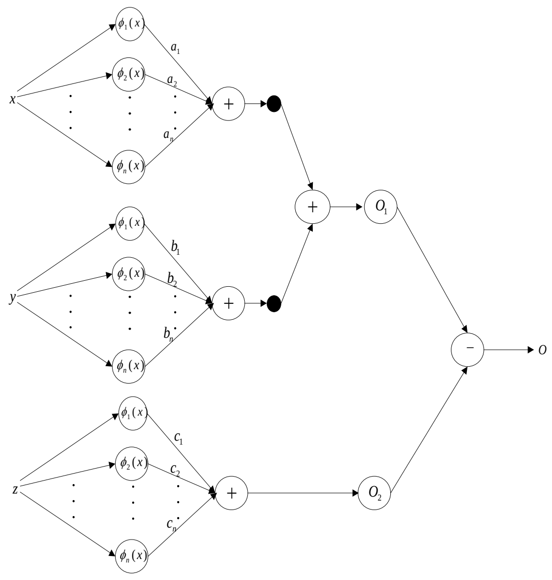

Figure 7. The expansion type associativity functional network model.

IV. THE EXPANSION TYPE FUNCTIONAL NETWORK PARAMETERS TO DIRECTLY DETERMINE THE METHOD

A. The Expansion Functional Neuron Parameters to Directly Determine

For the supervised training learning algorithm, given in advance a set of learning sample sets used for the training of the functional neuron networks, according to the expansion type functional neuron model(Fig.5) and Eq.(2), and we may assume that the definition is the expansion type functional neuron networks of the error cost function E as follows:

$$E=\frac{1}{2}\sum_{i=1}^{m}(y_i-O_i)^2=\frac{1}{2}\sum_{i=1}^{m}\left((y_i-\sum_{p=1}^{n}a_p\phi_p(x_i))\right)^2 \quad (6)$$

Then shown in Fig.5 the expansion type functional neuron functional parameters a_j use adaptive adjustment method as follows:

$$a_j(k+1)=a_j(k)-\eta\frac{\partial E}{\partial a_j}$$

$$=a_j(k)-\eta\sum_{i=1}^{m}[\phi_j(x_i)\left(\sum_{p=1}^{n}a_p\phi_p(x_i)-y_i\right)];\ j=1,2,\ldots,n \quad (7)$$

Where, η is the learning rate.

According to the Ref. [7], and further can be written as the matrix-vector form as follows:

$$A(k+1)=A(k)-\eta X^T[XA(k)-Y] \quad (8)$$

Where, A is the functional parameters vector, X is input to the basis function cluster matrix, and Y is the objective output vector. They are defined as follows:

$$A=\begin{pmatrix}a_1\\a_2\\\vdots\\a_n\end{pmatrix}\in R^n, X=\begin{pmatrix}\phi_1(x_1)&\phi_2(x_1)&\cdots&\phi_n(x_1)\\\phi_1(x_2)&\phi_2(x_2)&\cdots&\phi_n(x_2)\\\vdots&\vdots&\ddots&\vdots\\\phi_1(x_m)&\phi_2(x_m)&\cdots&\phi_n(x_m)\end{pmatrix}\in R^{m\times n}, Y=\begin{pmatrix}y_1\\y_2\\\vdots\\y_m\end{pmatrix}\in R^m$$

Then, shown in Fig.5 the single functional neuron expansion model the optimal steady state functional parameters can further be directly given as:

$$A=(X^TX)^{-1}X^TY \quad (9)$$

Or can be written as $A=pinv(X)Y$, where $pinv(X)$ represents input to the pinv Moore-Penrose [8] of the basis function cluster matrix X (In addition, it is equal to $(X^TX)^{-1}X^T$ and you could call the MATLAB command $pinv$ to achieve).

B. The Expansion Type Functional Network Parameters Direct Determination

The associativity functional network model of each functional neurons are expanded by the above method, whereby to get Fig.7 model. For the supervised training learning algorithm, we use some data consisting of triplets $\{((x_{1j},x_{2j}),x_{3j})\mid j=1,2,\ldots,n\}$. We define the error cost function as follows:

$$e_j=\hat{f}_1(x_{1j})+\hat{f}_2(x_{2j})-\hat{f}_3(x_{3j});\ j=1,2,\ldots,n \quad (10)$$

This indicates that the Eq. (3) each of a function f_s can be approximated by a linear combination of the some known basis function cluster $(\phi_s=\{\phi_{s1},\ldots,\phi_{sm_s}\}, s=1,2,3)$, and can be represented as follows:

$$\hat{f}_s(x)=\sum_{i=1}^{m_s}a_{si}\phi_{si}(x);\ s=1,2,3 \quad (11)$$

Where, a_{si} are functional parameters, ϕ_{si} are given sets of given linearly independent functions, called basis functions. That is Eq. (10) can be written as follows:

$$e_j=\sum_{j=1}^{n}\sum_{i=1}^{m_1}a_{1i}\phi_{1i}(x_{1j})+\sum_{j=1}^{n}\sum_{i=1}^{m_2}a_{2i}\phi_{2i}(x_{2j})-\sum_{j=1}^{n}\sum_{i=1}^{m_1}a_{3i}\phi_{3i}(x_{3j})$$

$$=\sum_{j=1}^{n}\left(\sum_{s=1}^{2}\sum_{i=1}^{m_s}a_{si}\phi_{si}(x_{sj})\right)-\sum_{j=1}^{n}\sum_{i=1}^{m_1}a_{3i}\phi_{3i}(x_{3j}) \quad (12)$$

Note that, for the sake of simplicity, the negative sign associated with the function f_3 in Eq. (10) has been included in the coefficient a_{3i}.

$$e_j=\sum_{j=1}^{n}\left(\sum_{s=1}^{3}\sum_{i=1}^{m_s}a_{si}\phi_{si}(x_{sj})\right) \quad (13)$$

Our goal is to let the output O as possible as equal to 0 or infinitely close to 0. Then, in order to obtain the optimal parameters of the network, it is necessary to explore minimize the sum of square errors:

$$E=\frac{1}{2}\sum_{j=1}^{n}\left(e_j-O_j\right)^2=\frac{1}{2}\sum_{j=1}^{n}\left(\sum_{s=1}^{3}\sum_{i=1}^{m_s}a_{si}\phi_{si}(x_{sj})-O_j\right)^2 \quad (14)$$

Similarly, the expansion type functional network model parameters adjustment and the single functional neuron expanded model parameters to adjust the same way, using the adaptive adjustment method as follows:

$$a_{tr}(k+1)=a_{tr}(k)-\eta\frac{\partial E}{\partial a_{tr}}$$

$$=a_{tr}(k)-\eta\sum_{j=1}^{n}\left[\phi_{tr}(x_{tj})\left(\sum_{s=1}^{3}\sum_{i=1}^{m_s}a_{si}\phi_{si}(x_{sj})\right)-O_j\right];$$

$$t=1,2,3;r=1,2,\ldots,m_t \quad (15)$$

According to the derivation of the functional parameters of functional networks in the single and functional neuron functional, further can be written as the matrix-vector form as follows:

$$A_s(k+1)=A_s(k)-\eta X_s^{\ T}[X_sA_s(k)-O];s=1,2,3 \quad (16)$$

Where, A_s is the functional parameters vector, X_s is input to the basis function cluster matrix. They are defined as follows:

$$A_s=\begin{pmatrix} a_{s1} \\ a_{s2} \\ \vdots \\ a_{sn} \end{pmatrix}\in R^n, X_s=\begin{pmatrix} \phi_{s1}(x_{s1}) & \phi_{s2}(x_{s1}) & \cdots & \phi_{sn}(x_{s1}) \\ \phi_{s1}(x_{s2}) & \phi_{s2}(x_{s2}) & \cdots & \phi_{sn}(x_{s2}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{s1}(x_{sm}) & \phi_{s2}(x_{sm}) & \cdots & \phi_{sn}(x_{sm}) \end{pmatrix}\in R^{m\times n}, O=\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}\in R^m$$

Then, shown in Fig.7 the expansion type associativity functional networks model the optimal parameters can further be directly given as:

$$A_s=(X_s{}^T X_s)^{-1} X_s{}^T O \tag{17}$$

Or can be written as $A_s = pinv(X_s)O$, where $pinv(X_s)$ represents input to the pinv Moore-Penrose of the basis function cluster matrix X_s (In addition, it is equal to $(X_s{}^T X_s)^{-1} X_s{}^T$ and you could call the MATLAB command $pinv$ to achieve).

V. Simulation Experiment

A. Experimental Platform

The experimental simulation platform: Operating system: Windows XP, CPU: AMD Athlon Core 5000+, Frequency: 2.61GHz, RAM: 1GB, Integrated development environment: MATLAB R2008a.

B. Experimental Results

1. The case of the single functional neuron expansion model: use two instances of the single functional neuron expansion model learning algorithm to test.

(1) The selected cosine objective function $y=\cos x$, for shown in Fig.5 the single functional neuron expansion model can be validated by computer simulation. Where, expand layer linear combination of basis function $\phi_n(x)$ respectively select the polynomial basis functions and cosine basis functions to test, and the expand layer of the basis functions $\phi_n(x)$ of number n is 10. The set of training samples generated from $[-2\pi, 2\pi]$ with the sampling interval 0.1.

①Using the polynomial basis functions, and after the proposed algorithm gets and the polynomial basis functions term corresponding to parameters are shown in Table Ⅰ.When the basis function chosen the polynomial basis functions, the test function approximation curve is shown in Fig.8.

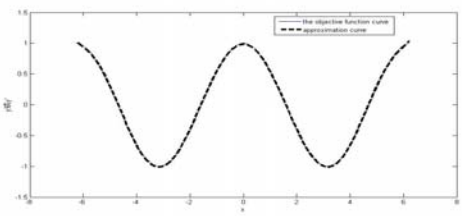

Figure 9. The test function approximation curve.

TABLE II.
THE POLYNOMIAL BASIS FUNCTIONS AND CORRESPONDING TO FUNCTIONAL PARAMETERS

The polynomial basis functions	The corresponding to functional parameters	The polynomial basis functions	The corresponding to functional parameters
1	0.983651299338308	x	0.001259351292262
x^2	-0.476433518207798	x^3	-0.000551183487709
x^4	0.036213675967953	x^5	0.000062626075084
x^6	-0.000936085524044	x^7	-0.000002560003003
x^8	0.000008246900005	x^9	0.000000034111147

Where, functional parameters can be obtained by Eq.(9) step of the calculation, the calculation time is 0.004507835493058s, the obtained $RMSE$(mean square error) is 8.873494726025255e-005.

② Using the cosine basis functions, and after the proposed algorithm gets and the polynomial basis functions term corresponding to parameters are shown in Table Ⅱ.

TABLE I.
THE COSINE BASIS FUNCTIONS AND CORRESPONDING TO FUNCTIONAL PARAMETERS

The cosine basis functions	The corresponding to functional parameters	The cosine basis functions	The corresponding to functional parameters
1	0.000000000000000	$\cos x$	0.999999999999999
$\cos 2x$	0.000000000000000	$\cos 3x$	0.000000000000000
$\cos 4x$	0.000000000000001	$\cos 5x$	0.000000000000000
$\cos 6x$	0.000000000000000	$\cos 7x$	0.000000000000000
$\cos 8x$	0.000000000000000	$\cos 9x$	0.000000000000000

When the basis function chosen the cosine basis functions, the test function approximation curve is shown in Fig.9.

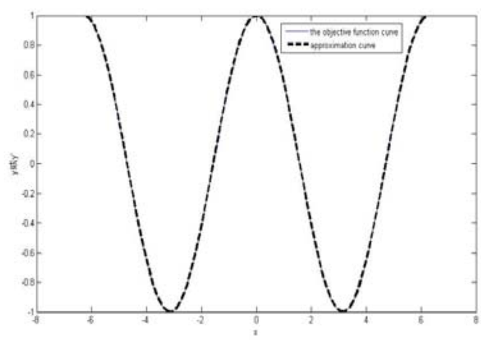

Figure 8. The test function approximation curve.

Where, functional parameters can be obtained by Eq.(9) step of the calculation, the calculation time is 0.004470400567670s, the obtained $RMSE$ is 2.787640534972165e-031.

(2) In the chemical reaction [9], the relations data tables between the product concentration y and time t are measured, as shown in Table III.

TABLE III.
THE RELATIONS BETWEEN THE PRODUCT CONCENTRATION y AND TIME t

x / min	1	2	3	4
y / %	4.00	6.40	8.00	8.80
x / min	5	6	7	8
y / %	9.22	9.50	9.70	9.86
x / min	9	10	11	12
y / %	10.0	10.2	10.32	10.42
x / min	13	14	15	16
y / %	10.50	10.55	10.58	10.60

The selected using as shown in Fig.5 the single functional neuron expansion model can be validated by computer simulation. Where, expand layer linear combination of basis function $\phi_n(x)$ select the polynomial basis functions and logarithmic basis functions to test, and the expand layer of the basis functions $\phi_n(x)$ of number n is 10.

①Using the polynomial basis functions, and after the proposed algorithm gets and the polynomial basis functions term corresponding to parameters are shown in Table Ⅳ.

TABLE IV.
THE POLYNOMIAL BASIS FUNCTIONS AND CORRESPONDING TO FUNCTIONAL PARAMETERS

The polynom -ial basis functions	The corresponding to functional parameters	The polynomial basis functions	The corresponding to functional parameters
1	1.906966847807739	x	0.883316600424253
x^2	2.092578677326181	x^3	-1.123561306128441
x^4	0.275417976556213	x^5	-0.039091533788201
x^6	0.003399991548682	x^7	-0.000178953408965
x^8	0.000005236463156	x^9	-0.000000065393005

When the basis function chosen the polynomial basis functions, the test function approximation curve is shown in Fig.10.

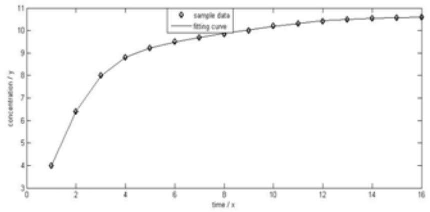

Figure 10. The product concentration y and time t relationship diagram.

Where, functional parameters can be obtained by Eq.(9) step of the calculation, the calculation time is 6.034286480544316e-004s, the obtained $RMSE$ is 7.921847102245422e-005. Compared with the Ref. [9] achieve results, the proposed method is simple, and the obtained $RMSE$(In the Ref. [9] of the obtained $RMSE$ is 0.021718) lower three orders of magnitude.

②Using the logarithmic basis functions, and after the proposed algorithm gets and the logarithmic basis functions term corresponding to parameters are shown in Table Ⅴ.

TABLE V.
THE LOGARITHMIC BASIS FUNCTIONS AND CORRESPONDING TO FUNCTIONAL PARAMETERS

The logarithmic basis functions	The corresponding to functional parameters	The logarithmic basis functions	The corresponding to functional parameters
1	0.000037217997596	$\log(1+x)$	-0.002124028767105
$\log(2+x)$	0.033939731506954	$\log(3+x)$	-0.245616652499821
$\log(4+x)$	0.967692701720154	$\log(5+x)$	-2.255084096761261
$\log(6+x)$	3.198257785643372	$\log(7+x)$	-2.712896972843567
$\log(8+x)$	1.265702027077774	$\log(9+x)$	-0.249907644514683

When the basis function chosen the logarithmic basis functions, the test function approximation curve is shown in Fig.11.

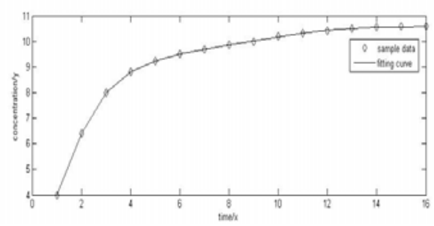

Figure 11. The product concentration y and time t relationship diagram.

Where, functional parameters can be obtained by Eq.(9) step of the calculation, the calculation time is 3.574421008353680e-004s, the obtained $RMSE$ is 3.134975467910307e-005. Compared with the Ref. [9] achieve results, the proposed method is simple, and the obtained $RMSE$(In the Ref. [9] of the obtained $RMSE$ is 0.021718) lower three orders of magnitude.

2. The case of the expansion type associativity functional network model:

Use the Ref. [6] of data (shown in Table VI) to test the expansion type associativity functional network model.

TABLE VI.
THE TEST SAMPLE DATA POINTS

x_{1j}	x_{2j}	x_{3j}	x_{1j}	x_{2j}	x_{3j}
0.9078	0.3165	1.0991	0.0281	0.6757	0.5170
0.0707	0.0285	0.0331	0.2018	0.5162	0.4569
0.6604	0.8147	1.0320	0.5256	0.3907	0.6061
0.8771	0.0558	0.8236	0.5379	0.2867	0.5414
0.4151	0.8210	0.7717	0.5973	0.4516	0.7294
0.4409	0.3413	0.4880	0.5810	0.9317	0.9960
0.5331	0.0248	0.3087	0.5529	0.2561	0.5337

The selected using as shown in Fig.7 the expansion type associativity functional network model can be validated by computer simulation. Where, each neuron expand layer linear combination of basis function $\phi_n(x)$ select the polynomial basis function (The Other basis functions can also be used) to test, and the expand layer of the basis functions $\phi_n(x)$ of number is 8.

In the expansion type associativity functional network model, the neuron function f_1 linear combination the polynomial basis functions term corresponding to functional parameters are shown in Table VII.

TABLE VII.
THE POLYNOMIAL BASIS FUNCTIONS AND CORRESPONDING TO FUNCTIONAL PARAMETERS

The polyno-mial basis function-ns	The corresponding to functional parameters	The Polyno mial basis functio n-ns	The corresponding to functional parameters
1	-0.000019743027333	x	0.000948660129663
x^2	-0.010184765166761	x^3	0.046299221414445
x^4	-0.106109149835774	x^5	0.128921017151726
x^6	-0.079255666061135	x^7	0.019391418200755

In the expansion type associativity functional network model, the neuron function f_2 linear combination the polynomial basis functions term corresponding to functional parameters are shown in Table VIII.

In the expansion type associativity functional network model, the neuron function f_8 linear combination the polynomial basis functions term corresponding to functional parameters are shown in Table IX.

TABLE VIII.
THE POLYNOMIAL BASIS FUNCTIONS AND CORRESPONDING TO FUNCTIONAL PARAMETERS

The polynomi-al basis functions	The corresponding to functional parameters	The Polynom-ial basis functions	The corresponding to functional parameters
1	0.000006748246098	x	0.000018073786960
x^2	-0.003817622768116	x^3	0.034645058774920
x^4	-0.121994543858773	x^5	0.205859063059051
x^6	-0.167085733530262	x^7	0.052449653668762

TABLE IX
THE POLYNOMIAL BASIS FUNCTIONS AND CORRESPONDING TO FUNCTIONAL PARAMETERS

The polynomi-al basis functions	The corresponding to functional parameters	The Polyno-mial basis functio-ns	The corresponding to functional parameters
1	-0.000066443507273	x	0.003071336537922
x^2	-0.025956494218575	x^3	0.094500170528520
x^4	-0.178906315219441	x^5	- 0.184426401517957
x^6	-0.098299435759611	x^7	- 0.021232046598695

Where, neuron function f_1, f_2 and f_3 functional parameters can be obtained by Eq. (17) step of the calculation, the final Fig.7 the expansion type associativity functional network model $RMSE$ can be obtained as 7.435405972405223e-004.

C. Experimental Analysis

From the Fig.8 and Fig.9 give the test functions approximation curves which prove that the approximation effect of the proposed algorithm is very well, the solid line as the objective function curve, and the dashed line is the approximation curve. The Fig.10 and Fig.11 are the data fitting in the chemical reaction, and finds the relation between the product concentration y and time t; thereby from in Fig.10 the fitting curve of fitting effect can be seen to be very well. Moreover, when the functional neuron function linear combination, we want to use the basis functions and the corresponding functional parameters are given in table.

VI. CONCLUSIONS

In this paper, according to the functional networks of functional neuron function can be used the other known form of a linear combination of the basis functions, and functional neuron model is expanded, then the expansion type functional neuron model is obtained, and functional parameters direct determination, in that case, not only the calculation amount is relatively small, but also the time consumed less, this fully reflects the expansion type functional network model and its parameters direct determination method advantage. Finally, simulation experiments show that the proposed method is feasible and effective.

ACKNOWLEDGEMENTS

This work is supported by National Science Foundation of China under Grant No. 61165015. Key Project of Guangxi Science Foundation under Grant No. 2012GXNSFDA053028, Key Project of Guangxi High School Science Foundation under Grant No. 20121ZD008 and the Funded by Open Research Fund Program of Key Lab of Intelligent Perception and Image Understanding of Ministry of Education of China under Grant No. IPIU01201100.

REFERENCES

[1] Castillo E. Functional Networks. *Neural Processing Letters*, vol.7, pp.151-159, 1998.

[2] Castillo E, Cobo A, Gutierrez J M. Functional Networks with Applications. *Kluwer Academic Publishers*, 1999.

[3] Castillo E, Gutierrez J M, Cobo A, et al. A minimax method for learning functional networks. *Neural Processing Letters*, vol. 1, pp. 39-49, November 2000.

[4] Iglesias A, Arcay B, Cotos J M, et al. A comparison between functional networks and artificial neural networks for the prediction of fishing catches. *Neural computer & applied*, vol.13, pp.24-31, 2004.

[5] Yongquan Zhou, Dongdong Wang, Ming Zhang. Designing Functional Networks Through Evolutionary Programming. *Proceedings of the 6th World Congress on Intelligent Controland Automation*, pp. 21-23, June 2006.

[6] Enrique Castillo, Angel Cobo, Jose Manual Gutiérrez, Rosa Eva Pruneda. *Functional Network with Applications*. Kluwer Academic Publishers, 1999.

[7] Yunong Zhang. Feed-forward Neural Network Activated with Power-activation and Its Weights-Determination Method. *The First Chinese Conference on Pattern Recognition*, pp.72-77, 2007. (In Chinese)

[8] Yunong Zhang, Yiwen Yang, Wei Li. *Weights Direct Determination of Neural Networks*, Sun Yat-sen University Press, 2010. (In Chinese)

[9] Yongquan Zhou. *Functional networks model and its learning algorithm with applications*, Publishing House of Electronics Industry, 2011. (In Chinese)

[10] Enrique Castillo, Ali S. Hadi, Beatriz Lacruz, Rosa E. Pruneda. Semi- parametric nonlinear regression and transformation using functional networks. *Computational Statistics & Data Analysis*, vol. 52, pp.2129- 2157, 2008.

[11] Yongquan Zhou, Bin Zhao, Licheng Jiao. Serial Functional Networks Model and Learning Algorithm with Applications. *Chinese Journal of Computers*, vol.31, pp.1073-1081, 2008.

[12] Yongquan Zhou, Licheng Jiao. Universal Learning Algorithm of Hierarchical Functional Networks. *Chinese Journal of Computer*, vol.28, pp.1277-1286, 2005. (In Chinese)

[13] Yongquan Zhou, Bin Zhao, Licheng Jiao. Recurrent functional networks model and learning algorithm. *Systems Engineering and Electronics*, vol.29, pp.1727-1731, 2007. (In Chinese)