正交泛函网络函数逼近理论及算法

周永权 吕咏梅 申 芸

(广西民族大学数学与计算机科学学院 南宁 530006)

摘 要 基于正交函数的概念和特性,提出一种正交泛函网络新模型,给出了正交泛函网络学习算法。该算法是借助于正交函数性质和Lagrange乘数法做辅助函数,对泛函参数学习过程归结为求解一组线性方程组的过程。最后,通过函数逼近算例计算机仿真结果表明,该算法十分有效,具有模型简单、逼近精度高等特点。

关键词 正交函数,Lagrange乘数法,正交泛函网络,学习算法,函数逼近

中图法分类号 TP18

Orthogonal Function Network Approximate Theory and Learning Algorithm

ZHOU Yong-quan LU Yong-mei SHEN Yun

(College of Mathematics and Computer Science, Guangxi University for Nationalities, Nanning 530006, China)

Abstract This paper presented a new kind of functional network model which is based on orthogonal functions, a learning algorithm of orthogonal functional network was proposed, the learning of function parameters uses orthogonal function characteristic and Lagrange multipliers by means of auxiliary function and solving a system of linear equation obtains functions parameters. Finally, An experiment in approximating typical functions was given. The simulation results show that the learning algorithm presented in the paper has rapid convergence speed and powerful performance.

Keywords Orthogonal function, Lagrange multipliers, Orthogonal functional networks, Learning algorithm, Function approximate

1 引言

1998年E. Castillo提出了泛函网络[1],是近年来提出的一种新的对传统神经网络的推广。泛函网络处理的是一般的泛函模型,它在各个神经元之间连接无权值,并且神经元函数不固定,而是可学习的,常常是一些给定的函数簇(如多项式、三角函数、Fourier展开式等)的线性组合,人们可根据特定的问题选取不同的函数簇来满足不同背景问题系统的建模和求解的需求。泛函网络已经被成功地应用于非线性系统辨识、混沌时间序列预测、微分、差分和泛函方程求解、CAD、线性、非线性回归等领域[2]。在此基础上,作者近年来致力于泛函网络新模型、学习算法和应用基础方面的研究,特别在计算智能的数学理论方法上获得了一些研究成果[3-12],发现泛函网络在解决上述问题中都表现出较好的逼近性能。虽然泛函网络已经成功被应用于很多方面,但由于泛函网络是近年来提出的一种新的对神经网络的推广,有些理论和应用方面的基础还不太健全,需要我们不断地提出更适合所要解决的新的网络结构,完善基础理论,提出新的逼近算法。

本文的主要工作是提出一种新的泛函网络模型,即正交泛函网络,网络中的神经元函数是一些正交函数,如著名Fourier级数、Bessel函数和Legendre多项式等基函数的线性组合。由于正交函数除了具备一般函数的性质外,本身还具备一些特殊的性质[13,14],因而被广泛地用于一些复杂系统的建模和求解,如非线性系统辨识、系统控制建模等。近年来,神经网络(包括正交神经网络)在这方面已取得了许多令人鼓舞的成果[15]。系统建模为多层前馈神经网络,通过优化一个代价函数的方法(如梯度法等)来进行权值学习。尽管这些算法在许多情况下性能较好,但实际中神经网络存在结构过于复杂、结点过多,并且在学习过程中容易陷入局部极小点而导致逼近能力有限等问题,因此在一些复杂系统建模时,存在较大的建模误差等问题。

另一方面,尽管泛函网络具有学习能力、多输入并行处理能力、非线性映射和容错能力,以及通过新的学习获得自适应性的能力,但是在其应用过程中,与神经网络所不同的是泛函网络不是权的学习,而是结构和参数的学习,学习的目的就是求出神经元函数的精确表达式或近似表达式。泛函网络的数学本质对应的是一些泛函变换,它的拓扑结构描述的是一个函数变换。函数变换的类型很多,所以表示函数变换的泛函网络模型也很多,其拓扑结构也千姿百态,很难构造出一个通用的泛函网络来表示所有的函数变换。但对正交泛函网络来讲,由于它的拓扑结构描述的是一个正交函数变换,因此我们可用一个统一的通用结构来描述所有的正交泛函网络,也可用一个统一的泛函方程表示所有的正交泛函网络,这就是正交泛函网络的优势所在。因此,研究正交泛函网络函数逼近

到稿日期:2008-02-22 本文受国家自然科学基金(60461001),国家民委科研基金(08GX01)和广西自然科学基金(0542048)资助项目。

周永权(1962—),男,博士,教授,主要研究方向为计算智能、神经网络及应用,E-mail:yongquanzhou@126.com;**吕咏梅**(1981—),女,硕士研究生,主要研究方向为计算智能及其应用;**申 芸**(1981—),女,硕士研究生,主要研究方向为神经网络及其应用。

机理及其学习算法，对于完善泛函网络基础理论和拓广其应用领域有着重要的理论价值和应用背景。

本文首先给出正交函数的概念和性质，作为设计正交泛函网络的基本理论。在此基础上，设计出一类正交泛函网络模型，然后给出正交泛函网络学习算法，该算法是基于正交函数特性和 Lagrange 乘数法，通过解方程组的方法来确定待学习的泛函参数，算法简单、有效可行。最后，通过一些函数逼近实例，计算机仿真表明，该算法具有求解简单、逼近精度高等特点。

2 理论基础

根据正交函数理论[13,14]，对于任意一函数 $f(x)$，$f:[a,b]\to\mathcal{R}$(实数域)，存在一个正交多项式

$$F_n(x)=a_1\phi_1(x)+a_2\phi_2(x)+a_3\phi_3(x)+\cdots+a_n\phi_n(x) \quad (1)$$

使得下面的式子成立

$$\lim_{n\to\infty}\int_a^b (f(x)-F_n(x))^2\mathrm{d}x=0 \quad (2)$$

其中

$$\int_a^b \phi_i(x)\phi_j(x)\mathrm{d}x=\begin{cases}0, & i\neq j\\ A_i, & i=j\end{cases}$$

$$a_i=\int_a^b f(x)\phi_i(x)\mathrm{d}x/A_i, i=1,2,\cdots \quad (3)$$

$\{\phi_1(x),\phi_2(x),\cdots,\phi_n(x)\}$ 是一个正交函数序列。

式(1)－(3)表明了任意一函数可由一组正交函数序列和系数 a_i 来惟一逼近。若函数 $f(x)$ 是未知，通过方程(3)我们无法获得系数 a_i。恰好这种情形在动力系统许多复杂建模时都属于函数 $f(x)$ 是未知或难以确定的情形，因此我们寻找一种新的方法，通过输入输出数据来确定系数 a_i，有着重要的应用价值。

接着分析正交函数的收敛性。考虑下式：

$$\begin{aligned}0&\leqslant\int_a^b (f(x)-\sum_{i=1}^n a_i\phi_i(x))^2\mathrm{d}x\\&=\int_a^b f(x)^2\mathrm{d}x-2\sum_{i=1}^n a_i\int_a^b f(x)\phi_i(x)\mathrm{d}x+\sum_{i=1}^n a_i^2\\&=\int_a^b f(x)^2\mathrm{d}x-\sum_{i=1}^n a_i^2\end{aligned}$$

由此可得到

$$\sum_{i=1}^n a_i^2\leqslant\int_a^b f(x)^2\mathrm{d}x \quad (4)$$

这说明正交函数表达式(1)中系数是收敛的。

此外，假设 $\sum_{i=1}^n b_i\phi_i(x)$ 是任意一正交函数的展开式，为了实现最佳逼近目标函数，寻求最优的系数 b_i，考察误差代价函数

$$E=\int_a^b (f(x)-\sum_{i=1}^n b_i\phi_i(x))^2\mathrm{d}x \quad (5)$$

利用式(3)关于正交函数系数的表达式，式(5)可写成

$$\begin{aligned}E&=\int_a^b (f(x)-\sum_{i=1}^n b_i\phi_i(x))^2\mathrm{d}x\\&=\int_a^b f(x)^2\mathrm{d}x+\sum_{i=1}^n A_i(b_i-a_i)^2-\sum_{i=1}^n A_i a_i^2\end{aligned} \quad (6)$$

从上面的讨论我们可得到：对任意一函数都可用一正交函数来实现逼近，且正交函数表达式中系数是惟一和有界的。基于这些事实，将正交函数集作为泛函网络的基函数集，正交函数表达式中系数作为泛函网络的参数来构建正交泛函网络，通过学习来确定泛函网络的参数，最终实现对任意一函数都可用一正交函数实现最佳逼近。

上面讨论的是一元正交函数情形。对于多元正交函数，我们可通过一元正交函数的合成或复合来实现对多元函数的逼近。例如一个 m 元函数 $f(X)$，可用 m 元正交函数集 $\{\Phi_1(X),\Phi_2(X),\cdots,\}$，其中每一个函数可定义为

$$\Phi_i(X)=\phi_{1i}(x_1)\phi_{2i}(x_2)\cdots\phi_{mi}(x_m);i=1,2,\cdots \quad (7)$$

这样一来，多元函数就转化成一元正交函数情形。

3 正交泛函网络

一般地，泛函网络由输入单元层、存储层、处理层、输出层和一组连接 5 部分组成[1]。它对应的是泛函变换，它的拓扑结构描述的是一个函数变换系统。那么，正交泛函网络结构除一般泛函网络特点外，它对应的是泛函正交变换，其拓扑结构描述的是一个正交函数变换系统。在实际应用中，根据问题的不同设计不同类型正交泛函网络，完成相应的正交函数变换。

3.1 正交泛函神经元网络模型

假设泛函神经元函数 $f(x)$ 是一个计算单元，它的计算模型如图 1 所示。

图 1 泛函神经元模型

根据泛函网络的结构特性，每个泛函神经元函数都可以由一些已知基函数簇的线性组合来逼近。根据第 2 节的知识，我们可将图 1 中泛函神经元表示成一组正交基函数的线性组合的形式。这里，我们称该神经元为正交泛函神经元。进一步，我们可将图 1 中的泛函神经元模型变为图 2 中的模型形式，即正交泛函神经元网络模型。

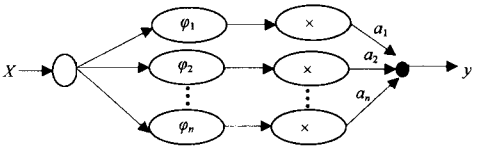

图 2 正交泛函神经元网络模型

实质上，图 2 给出的正交泛函网络是图 1 泛函网络的展开形式，这样显得结构直观明了，便于下面讨论。

在图 2 中，假设输入数据 $X=\{x_1,x_2,\cdots,x_n\}$，输出数据为 y，则该网络输出表达式

$$y=f(X)=\sum_{i=1}^n a_i\varphi_i(X) \quad (8)$$

其中，a_i 是正交函数 $\varphi_i(X)$ 的系数；$\{\varphi_1(X),\varphi_2(X),\cdots,\varphi_n(X)\}$ 是正交函数集。在正交泛函网络模型中，用途最广泛的是一些可分离正交泛函网络。

3.2 可分离正交泛函网络模型

一个可分离正交泛函网络结构如图 3 所示。

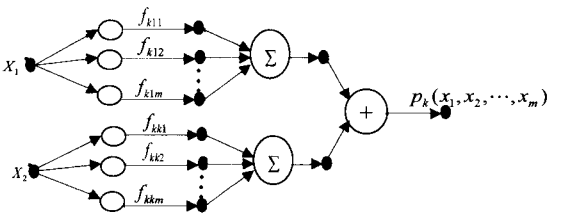

图 3 一个可分离正交泛函网络模型

设该网络的输入数据为 $X_1=\{x_1,x_2,\cdots,x_m\}$，$X_2=\{x_1,x_2,\cdots,x_m\}$，则该网络输出为

$$p_k(x_1,x_2,\cdots,x_m)=\sum_{k=1}^{p}\sum_{j=1}^{m}f_{kij}(x_j) \tag{9}$$

其中神经元$\{f_{kij}\mid k=1,2,\cdots,p;i=1,2;j=1,2,\cdots,m\}$是形如(1)式的正交多项式。

4 正交泛函网络学习算法

本节我们以可分离正交泛函网络为例来给出其学习算法。在图 3 所示的网络结构中，其中每一个泛函神经元可以是图 1 所示的模型，也可以是变形后图 2 所示的模型。以下分两种情形：

①若泛函神经元是图 2 所示的模型，事先给定一组正交基函数为

$$\{\varphi_{lij}(x_j)\mid i=1,2,\cdots,k;j=1,2,\cdots,m;l=1,2,\cdots,K\}$$

于是，在图 3 所示的模型中，每一个泛函神经元为 $f_{lij}=\sum_{l=1}^{K}a_{lij}\varphi_{lij}(x_j)$，则式(9)可写成

$$p_k(x_1,x_2,\cdots,x_m)=\sum_{i=1}^{k}\sum_{j=1}^{m}\sum_{l=1}^{K}a_{lij}f_{lij}(x_j) \tag{10}$$

②若图 3 所示的模型中每一个泛函神经元本身就是给定的正交基函数为

$$\{\varphi_{ij}(x_j)\mid i=1,2,\cdots,k;j=1,2,\cdots,m\}$$

图 3 所示的泛函网络的输出为

$$p_k(x_1,x_2,\cdots,x_m)=\sum_{i=1}^{k}\sum_{j=1}^{m}a_{ij}\phi_{ij}(x_j) \tag{11}$$

不论哪种情形，系数代表是泛函网络的参数，通过学习来确定最优的参数。

在第②种情形下，我们给出可分离正交泛函网络学习算法。考虑误差代价函数

$$e_p=y_p-p_k(x_1,x_2,\cdots,x_m)=y_p-\sum_{i=1}^{k}\sum_{j=1}^{m}a_{ij}\phi_{ij}(x_{jp}) \tag{12}$$

其中，p 为当前训练模式数，y_p 是第 p 个训练模式的期望输出。为了找到最优的泛函参数，最小化误差平方和为

$$E_0=\frac{1}{2}\sum_{p=1}^{P}e_p=\frac{1}{2}\sum_{p=1}^{P}\Big[y_p-\sum_{i=1}^{k}\sum_{j=1}^{m}a_{ij}\phi_{ij}(x_{jp})\Big]^2 \tag{13}$$

为了保证泛函网络表达式的唯一性，需要给出网络的初始值。在这里，设网络的初始值为

$$f_{ij}(x_{j0})=u_{j0},j=1,2,\cdots,m \tag{14}$$

其中，u_{j0} 是给定的一些常数。我们的目标是计算能够使得误差函数尽量小的网络参数：

$$\{a_{ij}\mid i=1,2,\cdots,k;j=1,2,\cdots,m\}$$

利用 Lagrange 乘数法，通过加惩罚项式(14)到(13)中，构造辅助函数如下：

$$E_\lambda=E_0+\frac{\lambda}{2}\sum_{j=1}^{m}(f_{ij}(x_{j0})-u_{j0})^2=\frac{1}{2}\sum_{p=1}^{P}\Big[y_p-\sum_{i=1}^{k}\sum_{j=1}^{m}a_{ij}\phi_{ij}(x_{jp})\Big]^2+\frac{\lambda}{2}\sum_{j=1}^{m}(f_{ij}(x_{j0})-u_{j0})^2$$

分别对泛函参数$\{a_{ij}\mid i=1,2,\cdots,k;j=1,2,\cdots,m\}$和参数 λ 求偏导数，得

$$\begin{cases}\dfrac{\partial E_\lambda}{\partial a_{tr}}=-\sum_{p=1}^{k}e_p\Big[\sum_{t\neq i=1}^{i}\sum_{r\neq j=1}^{m}a_{ij}\phi_{ij}(x_j)\Big]\phi_{tr}(x_r)+\\ \quad\lambda(f_{ij}(x_{j0})-u_{j0})\phi_{tr}(x_{j0})=0\\ \dfrac{\partial E_\lambda}{\partial\lambda}=\dfrac{1}{2}\sum_{j=1}^{m}(f_{ij}(x_{j0})-u_{j0})^2=0\end{cases} \tag{15}$$

这样，任给定一些正交基函数$\{\varphi_{ij}(x_j)\mid i=1,2,\cdots,k;j=1,2,\cdots,m\}$，通过解方程组(15)，可求得泛函参数$\{a_{tr}\mid t=1,2,\cdots,k;r=1,2,\cdots,m\}$及 λ。由于在第 2 节，我们已经知道该泛函参数是惟一的，因此通过解方程组(15)所得到解是最优解。至于其他类型的正交泛函网络的学习算法，可仿照该过程给出，这里不再赘述。

5 函数逼近实例

本节仅考虑正交泛函网络函数的逼近性能。为了验证本文所提出的正交泛函网络学习算法的有效性，定义均方差：

$$RMSE=\sqrt{\frac{1}{P}\sum_{i=1}^{P}\|y_i-\hat{y}_i\|^2} \tag{16}$$

其中 $\hat{y}_i$ 表示网络的输出，y_i 表示网络的实际输出，P 表示学习样数。

例 1(离散数据情形) 选用时间序列中典型的 Henon 映射为例。

Henon 序列系统为[1]

$$x_n=1-1.4x_{n-1}^2+0.3x_{n-2}$$

在初始值 $x_0=0.5,x_1=0.5$ 给定的情况下，Henon 映射的时间序列点由图 4 给出。

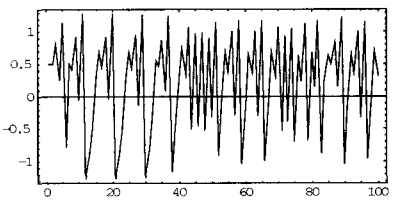

图 4 Henon 映射图

我们用正交泛函网络模型(如图 5)来进行计算，其中每个泛函神经元函数都是正交泛函神经元网络模型。选取 100 个样本点作为学习样本，采用 Mathematic 5 编程，仿真结果如图 5 和表 2 所示。为了便于比较，图 5 为给出泛函网络模型情形计算机仿真(表 1)结果。

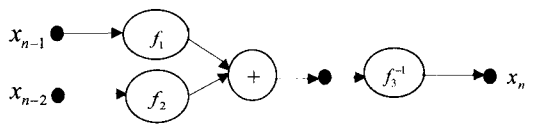

图 5 一个五层的正交泛函网络模型

表 1 Castillo 模型仿真结果

模型	ϕ_k k=1,2	ϕ_3	RMSE
1	$\{1,x,x^2\}$		0
2	{sinx,sin2x,cosx,cos2x}	{1,x}	0.005932
3	{sinx,sin2x,cosx,cos2x}		0.003378
4	{1,log(2+x),log(3+x),log(4+x),log(5+x)}		0.00001598

表 2 正交泛函网络仿真结果

Model	ϕ_k k=1,2	RMSE
1	Legendre polynomials basis:$\{1,x,\frac{1}{2}(3x^2-1),\frac{1}{2}(5x^3-3x)\}$	0
2	Chebyshev polynomials basis:$\{1,x,2x^2-1,4x^3-3x\}$	0
3	Fourier Series:{1,sinx,cosx,sin2x,cos2x}	0.00000231
4	Fourier Series:{1,sinx,cosx,sin2x,cos2x,sin3x,cos3x}	1.90767×10^{-7}

与 Castillo 模型相比较，仿真结果表明，选取正交泛函网络具有更高的逼近精度。

例 2(连续函数的情形)　选用常用函数

$y=\sin(x)\quad x\in[0,2\pi]$

我们选用具有一个输入、一个输出且含有两个神经元的正交泛函网络对该函数进行逼近，如图 6 所示。

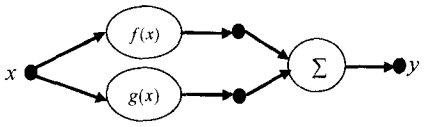

图 6　一个输入、一个输出的正交泛函网络

首先，随机地给定一组训练样本数据$\{(x_{0i},x_{1i})\mid i=1,2,\cdots,20\}$，其中 $x_{1i}=f(x_{0i})$，见表 3。

表 3　训练样本数据

x_0	x_1	x_0	x_1	x_0	x_1	x_0	x_1	x_0	x_1
1.9783	0.9181	6.0310	−0.2495	4.5812	−0.9914	4.9594	−0.9696	0.1069	0.1066
3.6584	−0.4941	5.2277	−0.8702	0.8489	0.7505	6.1066	−0.1756	6.1537	−0.1290
1.2498	0.9489	4.0215	−0.7707	4.4043	−0.9529	5.7016	−0.5494	3.0201	0.1212
5.3794	−0.7857	3.4010	−0.2565	0.7825	0.7051	1.0919	0.8875	2.7278	0.4020

若选用正交基函数 $\phi_1=\{\sin(x),\cos(x),\sin(2x),\cos(2x)\}$和 $\phi_2=(1,\sin(x),\cos(x))$来逼近神经元函数 $f(x)$和 $g(x)$，用 Mathematic 5 进行计算，得到均方差(RMSE)＝0.000311，逼近函数和原函数的逼近图形如图 7 所示。

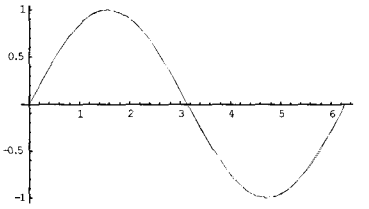

图 7　函数的逼近图形

例 3　逼近连续函数

$y=\cos(\sin(x)),x\in[0,1]$

我们选用具有两个输入、一个输出且含有两个神经元的正交泛函网络对该函数进行逼近(见图 5)。首先，我们给出一组训练样对$\{(x_{1i},x_{2i},x_{0i})\mid i=1,2,\cdots,20\}$，其中 $x_{0i}=f(x_{1i},x_{2i})$，如表 4。

表 4　用于逼近的训练数据对

x_1	x_2	x_0	x_1	x_2	x_0	x_1	x_2	x_0	x_1	x_2	x_0
.8933	.7792	.7115	.7883	.7092	.7589	.7706	.6965	.7671	.8537	.7537	.7291
.1232	.1229	.9924	.1640	.1633	.9867	.3136	.3085	.9528	.8300	.7379	.7399
.0321	.0321	.9995	.2172	.2154	.9769	.1009	.1007	.9950	.1487	.1481	.9890
.2112	.2097	.9781	.9858	.8337	.6721	.9057	.7868	.7061	.8840	.7733	.7156
.0510	.0509	.9987	.0301	.0301	.9995	.4671	.4503	.9003	.2131	.2115	.9777

我们选择用于逼近泛函神经元的两组正交基函数为

$\phi_1=\{1,\sin(x),\cos(x),\sin(2x),\cos(2x)\}$；$\phi_2=(1,\sin(x),\cos(x))$

用 Mathematic 5 进行计算，得到均方差(RMSE)＝0.0000116，逼近函数和原函数的逼近图形如图 8 所示。

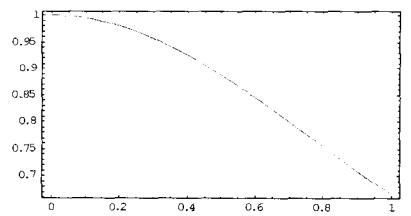

图 8　函数的逼近效果

从上面的仿真实例可得出，正交泛函网络模型用于函数逼近是可行的，而且有更高的逼近精度。

结束语　本文提出一种正交泛函网络模型，给出了正交泛函网络的学习算法，而网络的参数是通过求解一组线性方程组得到，算法简单可行。仿真结果验证了该算法在函数逼近方面具有很高的逼近精度。

参 考 文 献

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

[2] Castillo E, Cobo A, Cutiérrez J M, et al. Functional Networks with Applications. Kluwer Academic Publishers,1999

[3] Zhou Yong - quan, Jiao Li - cheng. Interpolation Mechanism of Functional Networks. Lecture Notes in Computer Science 3697. 2005:45-51

[4] Zhou Yong - quan, He Deng - xu, Nong Zheng. Application of Function al Networks to Solving Classification Problems. ENFORMATIKA,2005,7:390-393

[5] Zhou Yong - quan, Jiao Li - cheng. Approximate Factorization Learning Algorithm of Multivariate Polynomials Based on Functional Networks. Journal of Information and Computational Science,2005,2(1):205-210

[6] Zhou Yong-quan, He Deng-xu, Jiao Li-cheng. A Neural Computation Structure for Solving Functional Equations with Functional Networks. Journal of Information and Computational Science,2005,2(4):835-842

[7] He Deng-xu, Nong Zheng, Zhou Yong-quan. Approximate Factorization of Univariate Polynomials Using Functional Networks. Journal of Information and Computational Science,2005, 2(3):473-479

[8] Zhou Yong-quan, Jiao Li-cheng. Application of Functional Networks to Solving Functional Equations // Proceedings 2005 International Conference on Neural Networks & Brain. IEEE Press, Beijing China, Oct. 2005:1378-1383

[9] 周永权. 一种基于泛函网络的多项式 Euclidean 算法. 计算机科学,2006,33(9):131-134

[10] 周永权,焦李成,李陶深. Fuzzy 插值及其 Fuzzy 泛函网络构造理论. 计算机科学,2007,34(7):5-9

[11] 周永权,焦李成. 基于遗传规划实现泛函网络神经元的函数类型优化. 计算机科学,2007,34(2):7-9

[12] 周永权,焦李成. 层次泛函网络整体学习算法. 计算机学报, 2005,28(8):1277-1286

[13] Courant R, Hilbert D. Methods of Mathematical Physics. New York: Interscience Publishers,1955:49-111

[14] Hildebrand F B. Advanced Calculus for Applications. Englewood Cliffs, NJ: Prentice-Hall,1976

[15] Yang Shiow - shung, Tseng Ching - shiow. An Orthogonal Neural Network for Function Approximation. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 1996, 26 (5):779-785