

一种基于进化泛函网络的建模与函数逼近方法

罗淇方 周永权 谢竹诚

(广西民族大学数学与计算机科学学院 南宁 530006)

摘 要 提出了一种进化泛函网络的建模与函数逼近方法,该方法把泛函网络建模过程转变为结构和泛函参数的优化搜索过程,利用遗传规划设计泛函网络神经元函数,对网络结构和参数共存且相互影响的复杂解空间进行全局最优搜索,实现泛函网络结构和参数的共同学习,并用混合基函数实现目标函数的逼近,改变了人们通常用同类型基函数来实现目标函数逼近的方式。数值仿真结果表明,提出的网络建模与逼近方法具有较高的逼近精度。

关键词 混合基函数,神经元函数,遗传规划,进化泛函网络,函数逼近

中图分类号 TP18 **文献标识码** A

Modeling and Function Approximation Approach Based on Evolutionary Functional Networks

LUO Qi-fang ZHOU Yong-quan XIE Zhu-cheng

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

Abstract A new genetic programming designing neuron functions, combining genetic programming and evolutionary algorithm, was proposed for hybrid identification of functional network structure and functional parameters by performing global optimal search in the complex solution space where the structures and parameters coexist and interact. The method using hybrid base functions is different from traditional method approximate object function. The computing results show the high precision by the new method.

Keywords Hybrid base functions, Neuron function, Genetic programming, Evolutionary functional networks, Function approximation

泛函网络(Functional Networks, FN)是 Enrique Castillo 于 1998 年提出的一种网络模型^[1],是对人工神经网络(Neural Networks, ANN)的一种有效拓广,仅处理一般泛函模型。它与 ANN 的主要区别在于它的泛函性质和无权值连接。一个泛函网络的拓扑结构,由于其神经元函数的不固定和可学习,往往能描述成一个函数簇(如多项式、Fourier 级数、三角函数等);其次,泛函网络在应用时通常根据待解问题的“先验知识”,采用“问题驱动”的方式来建立其初始的网络结构,而不像 ANN 那样采用“模型驱动”的方式来建立其网络的初始结构,该结构往往是一“黑箱”结构。因此,相比来说, FN 比 ANN 建模较容易一些。近年来,泛函网络已经成功地应用于混沌时间序列的预测、微分、差分 and 泛函方程的求解、CAD、线性、非线性回归^[2,3]、非线性系统辨识^[4-6]、故障检测与诊断^[7]、计算机代数^[8-18]等领域,在解决上述问题中都表现出了良好的逼近性能。

本文将遗传规划与泛函网络有机地结合起来,把泛函网络看作是结构和参数的优化搜索过程,对网络结构和泛函参数共存且相互影响的复杂解空间进行全局最优搜索,实现泛函网络结构和参数的共同学习,提出了一种基于遗传规划设计网络的方法。该方法最大的特点是结构和参数同时进

化,且给出了泛函网络神经元函数的描述方法;在适应值函数设计中考虑了网络结构的复杂性和训练误差,算法中采用复制、交叉和突变算子;在遗传进化过程中不断动态变更网络的结构和神经元函数结构的大小。本文与同类文献方法的主要区别在于实现神经元函数学习是基于进化机制的,采用混合基函数来实现学习,而不是人们通常采用的同类型基函数簇。计算机仿真实验表明,该方法同样能改善网络的整体逼近性能,最终获得更为简洁的泛函网络结构和神经元函数表达式。

1 泛函网络的构造

一般地,泛函网络由以下元素组成^[1]: (1)输入单元层。这是输入数据的一层单元,输入单元以带有相应名字的实心圆来表示。(2)输出单元层。这是最后一层单元,它输出网络的结果是数据。输出单元也用带有相应的名字的实心圆来表示。(3)一层或若干层神经元。每一个神经元是一个计算单元,它计算的是一组来自前一层神经元或输入单元的输入值,并给下一层神经元或输出单元提供数据。计算单元相互连接,每一个神经元的输出可作为另一个神经元或输出单元输入数据的一部分。一旦给定输入值,输出便由神经元的类型

到稿日期:2009-08-18 返修日期:2009-10-30 本文受国家自然科学基金项目(60461001),广西自然科学基金(0832084,0991086),国家民委科研项目(08GX01)资助。

罗淇方(1976-),女,硕士,工程师,主要研究方向为计算智能与智能系统;周永权(1962-),男,博士,教授,主要研究方向为神经网络、计算智能及其应用,E-mail:yongquanzhou@126.com;谢竹诚(1985-),男,硕士,主要研究方向为进化计算。

来确定,它由一函数定义。(4)存储层。该层单元不具有任何处理功能,只存储处理过程中的信息并使不同处理单元的输入相等。该层可以是一层,也可以是多层。(5)有向连接线。它们将输入层、中间层、输出单元层连接起来,箭头表示信息流向。所有这些元素一起形成了泛函网络的结构,它确定了网络的泛函能力。

一般地, n 层泛函网络的模型为

$$Y = f_n \cdot f_{n-1} \cdot \dots \cdot f_1(X) \quad (1)$$

对应的网络结构如图1所示。

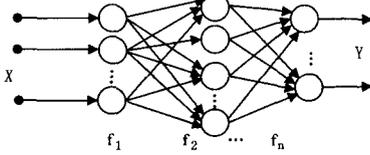


图1 一般泛函网络的拓扑结构示意图

其中, $Y \in R^m$ 表示系统的输出, $X \in R^n$ 表示系统的输入。 f_1, f_2, \dots, f_n 分别表示不同网络层的泛函神经元函数,其接受上层函数的输入(可能是多维输入),负责向下层函数输出数据(可能是多维输出)。

Castillo的构造方法是根据所给问题所蕴涵的“经验知识”采取一定的简化假设选取一些同类型基函数簇,将每一神经元函数 $f_i, i=1, 2, \dots, n$,表示成一些已知基函数簇的线性组合作为 f_i 的待定形式,再由最小二乘法等回归技术确定其中的泛函参数,通过对比选择其中最好的一个作为最终结果。然而,由于现实数据的极其复杂性,往往无法获取足够的信息来预见 f_i 的形式,片面的或局部硬性的指定将会导致大的预测误差。因此,须从全局对 f_i 进行优化。而遗传进化算法的广泛可用性和全局最优性提供了可靠的工具和手段。

2 基于进化机制的泛函网络设计方法

2.1 进化泛函网络设计方法

从函数变换的观点来看,实质上式(1)表示的是一个复杂的函数变换系统。而对于一个复杂的函数变换,应分析判断对应于该网络的输入是否都有相同输出的等价泛函网络。如果对任何一组给定输入都有相同输出的两个网络,则这两个网络是等价的。这样,对任意一泛函网络都可找到相等价的、结构简单的网络与它对应(如图1所示)。那么,在式(1),对每一个神经元函数(不妨考虑一元函数的情形)都可写成

$$f_i(x) = \sum_{j=1}^{m_i} a_{ij} \phi_{ij}(x) \quad i=1, 2, \dots, n \quad (2)$$

1) 个体的描述

为了叙述方便,再引入记号

$$GEA(\text{算术运算符集 } A; \text{函数集 } F; \text{终止符集 } T) \quad (3)$$

式中,算术运算符集为 $A = \{+, -, *, /, \dots\}$;函数集 F 为标准数学函数,而终止符集 T 是通常意义下的参变量或常量。式(3)可进一步表示成

$$GEA(+, -, *, /, pow, exp, log, sin, con, \dots, x_1, x_2, \dots) \quad (4)$$

这样,事先不需要对每一神经元函数 $f_i, i=1, 2, \dots, n$ 的形式作任何假定,按遗传进化机理对函数和变量进行随机选择,可得到形如式(5)的各种神经元函数表达式。

$$y_1 = \sum_{j=1}^{m_1} a_{1j} \phi_{1j}(x); y_2 = \sum_{j=1}^{m_2} a_{2j} \phi_{2j}(x); \dots; y_n = \sum_{j=1}^{m_n} a_{nj} \phi_{nj}(x) \quad (5)$$

式中, $a_{ij} \in T, \phi_{ij} \in F, x \in T, \sum, * \in A; i=1, 2, \dots, n; j=1, 2, \dots, m_i$,它们都能以树状结构的形式表达问题。

2) 初始种群的形成

初始种群用随机的方法产生,从函数集 F 及终止符集 T 中随机地选取基函数及变量 x ,组成各种复杂的数学函数。假设产生的初始群体有

$$\begin{aligned} \text{个体 } 1: & a_{11} \cdot \phi_{11}(x) + a_{12} \cdot \phi_{12}(x) + \dots + a_{1m_1} \cdot \phi_{1m_1}(x) \\ \text{个体 } 2: & a_{21} \cdot \phi_{21}(x) + a_{22} \cdot \phi_{22}(x) + \dots + a_{2m_2} \cdot \phi_{2m_2}(x) \\ & \dots \dots \dots \\ \text{个体 } n: & a_{n1} \cdot \phi_{n1}(x) + a_{n2} \cdot \phi_{n2}(x) + \dots + a_{nm_n} \cdot \phi_{nm_n}(x) \end{aligned} \quad (6)$$

在本问题中,染色体不再是传统的二进制位串,而是形如式(6)的函数表达式。在分析时把函数的表达式用数据结构中的二叉树来表示,节点集合= $\{\text{算术运算符}\}$ (非叶节点的节点)、叶节点= $\{\text{终止符集}\} = \{\text{函数中出现的变量和参数的集合}\}$ 。结合式(6),进而把基于二进制位串的遗传操作改进为针对二叉树的遗传操作。由运算的优先级可知,一个函数的二叉树表示可能不是唯一的,但由它的二叉树表示得到的节点集合和叶节点集合一定是唯一的。基于这些特性,下面给出基于二叉树的遗传操作。

2.2 基于二叉树的遗传操作

假设某一泛函网络中对应两个神经元函数个体表达式如图2所示。不妨考虑用 $n=2, m_1=2, m_2=3$ 的情形来说明其遗传操作算子。

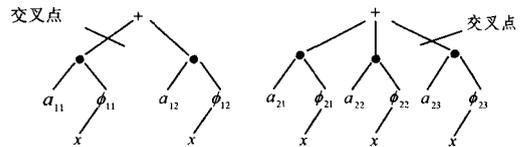


图2 交叉前的示意图

2.2.1 交叉操作

交叉后如图3所示。

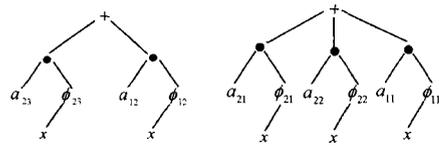


图3 交叉后的示意图

由于交叉点是随机选择的,因此即使两个相同的个体自身交叉,产生的个体也未必和原来的个体相同。

2.2.2 复制操作

由于复制操作仅仅是把染色体进行简单的拷贝,而染色体本身不发生变化,因此复制操作无须修改。即按轮盘法确定染色体的选择份数,然后进行复制操作。给适应度高的个体较高的生存概率,遵照“优胜劣汰”原则。

2.2.3 变异操作

由于函数中存在着两个不同的集合(节点集和叶节点集),因此变异操作也存在着运算符变异和终止符变异两种。

1) 运算符变异操作

从已知树中取一个节点运算符,随机地变成另一个与它具有相同运算目数的运算符,如图4所示。若取基函数 $(\phi_{11}, \phi_{12}) = \{1, \cos x\}$,则个体函数形如

父代: $f(x) = a + b \cos x$; 子代: $f'(x) = a * b \cos x$

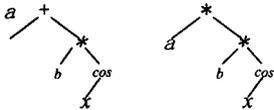


图4 运算符变异操作示意图

2) 终止符变异操作

从已知树中随机选取的一个终止符可变成任意一个终止符(如图5所示)。如

父代: $f(x) = a + b \cos x$; 子代: $f(x) = 2a \cos x$

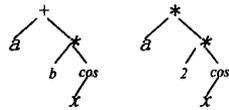


图5 终止符变异操作示意图

总之,按遍历树生成法随机产生一组函数表达式作为初始群体,进入遗传进化自适应迭代过程,按遗传机理产生新群体。特别注意,这一阶段得到的只是一个表达式,不能直接进行评价,需等泛函参数进化完成之后再行。

3) 进化泛函网络结构设计步骤

使用模型(6)的形式描述任一泛函网络的逼近系统,还需要确定每个个体参数 a_{ij} ,若取值不当,可能导致优良的泛函网络模型被淘汰。在这种情况下,将泛函网络结构和参数的组合优化过程分成相对独立且简单的结构与参数的寻优问题;对泛函网络结构中神经元函数的优化采用遗传规划符号回归技术,而对参数的优化则采用Lagrange乘数法完成,实现泛函网络结构和参数分别优化,共同识别的目的。归结起来,进化泛函网络结构设计包括以下3步。

(1) 泛函网络结构进化。式(2)中函数 f_i 由泛函网络结构来确定,每一神经元函数都可由一些基函数的线性组合来表示。因此,泛函网络结构进化过程只对基函数和变量进行进化操作,无须考虑泛函参数的影响。则按遍历树生成法随机产生一组基函数表达式,即如式(6)所示的个体作为初始群体进入遗传进化自适应迭代过程,按遗传机理产生新群体。特别注意,这一阶段得到的只是一个表达式结构,还不能直接进行评价,须等参数进化完成之后再行。

(2) 泛函参数进化。对在泛函网络结构进化过程中产生的每一个形如式(6)的框架,析取参数相关信息,包括参数的个数、取值范围等。这样对每一个泛函网络结构来说,问题转化为确定结构下的泛函参数优化问题。由于当前结构下每一组模型的参数个体都对应一个完整泛函网络学习模型,可用Lagrange乘数法,通过解方程组方法求最优泛函参数。

(3) 泛函网络结构性能评价。这一步对泛函参数进化得到的每个泛函网络结构模型下的优化模型,用整体适应度函数式(7)对其进行评价,并将评价结果用于泛函网络结构进化过程。在完成泛函网络结构进化的同时,也确定了相应的最优泛函参数。

重复(1)~(3)步,最终可获得所求的最优泛函网络结构建模。

4) 整体逼近性能测定

适应度是衡量网络整体逼近性能的主要尺度。每完成一代进化,就要对当前的泛函网络结构进行性能评价,其方法如下。

设采样点数为 m ,样本集合为 $\{(x_j, d_{ij} = f_i(x_j))\}_{j=1}^m$,我们对目标函数 $f_i(x), i=1, 2, \dots, n$ 进行逼近。设精度要求为 $\epsilon > 0$ (如 $\epsilon = 10^{-6}$),对每个样本的输入进行预测分析,则预测输出和实际值之间的误差为

$$E = \sum_{i=1}^n \sum_{j=1}^m [d_{ij} - \sum_{i=1}^n f_i(x_j)]^2 \quad (7)$$

则适应度函数定义为

$$Fitness = \frac{1}{1+E} \quad (8)$$

每一代都可得到泛函网络的预测输出和实际输出。在实际应用中,用式(7)和式(8)式检验泛函网络的逼近能力。

根据上面算法设计思路,泛函网络结构优化算法实施步骤如下。

初始化:置进化代数 $t=1$;确定交叉概率 p_c 和变异概率 p_m ;随机产生 N 个个体 $g_i(t, x)$ 作为构造泛函网络的神经元函数,构成种群 $P(t, x) = \{g_i(t, x) = \sum_{j=1}^M a_{ij}(t) \phi_{ij}(t, x) | i=1, 2, \dots, N\}$;

While(迭代次数小于预定的进化代数)

{对每一 $i=1, 2, \dots, N$;根据 $g_i(t, x)$ 构造出相应的泛函网络;

基于对泛函网络结构和泛函参数分别优化和共同识别的原则来评价泛函网络的性能;根据式(8)计算当前网络的适应值 $Fitness_i(t)$;如果 $Fitness_i(t)$ 满足要求,则结束;否则,置 $t=t+1$;对第 $t-1$ 代群体进行选择操作,形成一新的GEA集;分别以交叉概率 p_c 和变异概率 p_m 对GEA集进行交叉和变异操作,形成新一代群体 $P(t, x)$;

}

5) 泛函参数直接确定

在泛函网络结构进化过程中,当完成进化的第(1)~(2)步后,泛函网络产生每一个表达式框架,并且在当前结构下每一组模型的泛函参数个体都对应一个完整泛函网络结构。从而可以析取泛函参数相关信息,包括参数的个数、取值范围等。这样对网络结构来说,问题转化为确定结构下的参数优化问题。因此,不妨以图6为例,来描述泛函网络的学习过程。

考察泛函方程

$$F(x, y) = f_3^{-1}[f_1(x) + f_2(y)] \quad (9)$$

则对应泛函网络如图6所示。

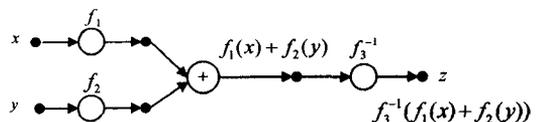


图6 用于逼近函数 $F(x, y)$ 的泛函网络

对函数 $F(x, y)$ 的学习实际上等价于对一元函数 $f_1(x), f_2(x)$ 和 $f_3(x)$ 的学习。任意给定一组训练数据 $\{(x_{1j}, x_{2j}, x_{3j}) | x_{3j} = F(x_{1j}, x_{2j}); j=1, 2, \dots, n\}$,我们给出其学习过程如下。

首先,经过上面进化后得到任意一组混合基函数集 $\phi_s = \{\phi_{s1}, \phi_{s2}, \dots, \phi_{sm}\}, s=1, 2, 3$ 。对每一神经元函数来说,都有表

达式

$$\hat{f}_s(x) = \sum_{i=1}^{m_s} a_{si} \phi_{si}(x), s=1,2,3 \quad (10)$$

式中, a_{si} 是泛函网络参数。我们必须有

$$x_{3j} = f_3^{-1}(f_1(x_{1j}) + f_2(x_{2j})), j=1,2,\dots,n \quad (11)$$

实质上,式(10)等价于

$$f_3(x_{3j}) = f_1(x_{1j}) + f_2(x_{2j}), j=1,2,\dots,n$$

定义误差函数

$$e_j = (\hat{f}_1(x_{1j}) + \hat{f}_2(x_{2j}) - \hat{f}_3(x_{3j})), j=1,2,\dots,n \quad (12)$$

计算其适应度:

$$Fitness = \frac{1}{1 + \sum_{j=1}^n e_j^2} = \frac{1}{1 + \sum_{j=1}^n (\sum_{s=1}^3 \sum_{i=1}^{m_s} a_{si} \phi_{si}(x_{sj}))^2} \quad (13)$$

为了保证网络的唯一性,任意选取一常数,使得

$$\hat{f}_k(x_0) = \sum_{i=1}^{m_k} a_{ki} \phi_{ki}(x_0) = \alpha_k, k=1,2,3 \quad (14)$$

定义辅助函数

$$Q = \sum_{j=1}^n e_j^2 = \sum_{j=1}^n (\sum_{s=1}^3 \sum_{i=1}^{m_s} a_{si} \phi_{si}(x_{sj}))^2 + \sum_{k=1}^3 \lambda_k (\sum_{i=1}^{m_k} a_{ki} \phi_{ki}(x_0) - \alpha_k) \quad (15)$$

利用 Lagrange 乘法法,分别对 a_{si}, λ_k 求偏导,得

$$\begin{cases} \frac{\partial Q}{\partial a_{sr}} = 2 \sum_{j=1}^n (\sum_{s=1}^3 \sum_{i=1}^{m_s} a_{si} \phi_{si}(x_{sj})) \phi_{sr}(x_{sj}) + \lambda_r \phi_{sr}(x_0) = 0 \\ \frac{\partial Q}{\partial \lambda_r} = \sum_{i=1}^{m_r} a_{ri} \phi_{ri}(x_0) - \alpha_r = 0, t=1,2,3, r=1,2,\dots,m_t \end{cases} \quad (16)$$

通过求解关于泛函参数 a_{si}, λ_r 的线性方程组,可得到所求最优泛函参数。

3 数值仿真实例

在实际应用中,先给出进化方法所需要的基本参数。在以下实例中,设置各种参数对应的值,如表 1 所列。

表 1 遗传规划所需要的基本参数

参数	值
目标	确定泛函神经网络及最佳结构
终止符集 T	变量 x_1, x_2, x_3
基函数集 F	pow, exp, log, sin, cos, ..., x^0, x, x^2, \dots
算术运算符集 A	+, -, *, /, ...
群体规模	N=25
交叉概率	$p_c=0.8$
突变概率	$p_m=0.2$
终止准则	规定最大迭代代数 200
交换后二叉树深度	k=6
初始群体产生方法	生长法

下面以典型的 Hénon 函数为例来说明本文提出的方法的有效性。

例 1 Hénon 函数为

$$x_n = 1 - 1.4x_{n-1}^2 + 0.3x_{n-2} \quad (17)$$

在初始值 $x_0=0.5, x_1=0.5$ 的情况下, Hénon 映射由图 7 给出。

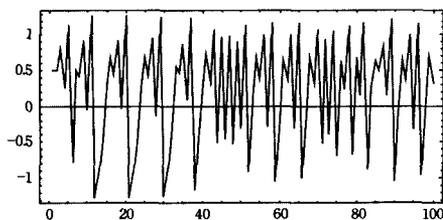


图 7 Hénon 函数图

用一个五层泛函网络模型(如图 8 所示)来进行仿真,说明本文方法的有效性、正确性。

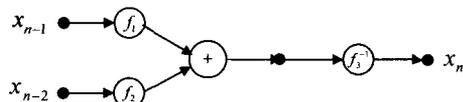


图 8 一个五层泛函网络模型

根据 Castillo 的方法,分别采用表 2 中的基函数 $\phi_k, k=1,2,3$ 来逼近 Hénon 映射函数,其逼近精度见表 2 中 RMSE (Root Mean Square Error)^[5] 的值。

表 2 用 Castillo 模型逼近的精度^[13]

模型	$\phi_k, k=1,2$	ϕ_3	RMSE
1	{1, x, x ² }		0
2	{sinx, sin2x, {1, x} cosx, cos2x}		0.00593219
3	{sinx, sin2x, cosx, cos2x}		0.00337892
4	{1, log(2+x), log(3+x), log(4+x), log(5+x)}		0.00001598

若用本文提出的方法对例 1 进行求解,在表 1 中给定对应的参数意义下,通过进化得到最优的混合基函数簇为 $\phi_k = \{e^{3x}, e^{4x}, \cos(x), \cos(4x), \sin(2x), x\}, k=1,2,3$,对应的图 8 中 3 个神经元函数表达式为

$$\begin{aligned} f_1 &= 2.9607 \cos(x_1) x_1 + 0.030073 e^{4x_1} x_1 \\ f_2 &= -1.3518 \cos(4x_2) x_1 - 1.4526 \cos(4x_2) x_3 \\ f_3 &= 0.49206 e^{3x_1} x_3 - 0.16725 e^{4x_1} x_3 - 0.29114 \sin(2x_3) x_2 \end{aligned} \quad (18)$$

对应得到的最佳逼近函数为

$$\begin{aligned} x_3 = & 0.030073 e^{4x_1} x_1 + 2.9607 \cos x_1^2 + 0.589005 e^{3x_1} x_2 - 5.4072 \cos x_1 x_2 \\ & - 0.49206 e^{3x_1} + 0.16725 e^{4x_1} + 5.8104 \cos x_2 - 0.58228 \sin x_2 \end{aligned} \quad (19)$$

其逼近精度 $RMSE=5.6198 \times 10^{-5}$ 。进化泛函网络方法得到的逼近函数曲线如图 9 所示。

图 10 给出了精确的 Hénon 函数与本文进化方法得到的近似于 Hénon 函数的误差变化曲线。

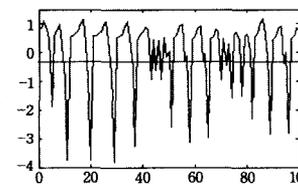


图 9 本文进化方法得到的近似 Hénon 函数曲线图

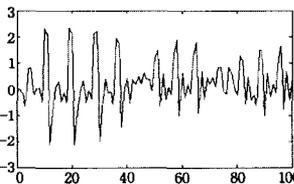


图 10 本文进化方法得到的近似函数与原 Hénon 函数的误差变化曲线图

从算例可看出,若计算精度用 RMSE 的值来评判,本文提出的进化泛函网络方法的计算精度在表 2 中的模型 1 与模型 4 之间。实际上,随着进化代数的增加,其逼近的精度愈来愈高。该进化泛函网络逼近方法的最大好处在于它可用混合基函数实现目标函数的逼近,改变了人们通常用同类型的基函数来实现目标函数逼近的方式,本文在这方面做了一有益的探索和尝试。

结束语 泛函网络是最近提出的一种新的对神经网络的有效推广,至今还没有系统设计方法能够对给定问题设计出近似最优的结构。本文把泛函网络建模过程转变为结构和泛函参数的优化搜索过程,利用遗传进化设计泛函网络的神经元函数,对网络结构和泛函参数共存且相互影响的复杂解空

间进行全局最优搜索,实现泛函网络结构和泛函参数的共同学习,有效提高了泛函网络的收敛精度,并可获得更为简洁的网络结构。

参 考 文 献

- [1] Castillo E. Functional Networks[J]. Neural Processing Letters, 1998,7:151-159
- [2] Castillo E, Cobo A, Gutierrez J M. Functional Networks with Applications[M]. Kluwer Academic Publishers, 1999
- [3] Castillo E, Hadi Ai S, Locruz B. Semi-parametric Nonlinear Regression and Transforming Using Functional Networks[J]. Computational Statistics & Data Analysis, 2008, 52: 2129-2157
- [4] Castillo E, Gutierrez J M, Cobo A, et al. A minimax method for learning functional networks[J]. Neural Processing Letters, 2000, 11(1): 39-49
- [5] Castillo E, Cobo A, Gutiérrez J M, et al. Functional networks with applications[M]. Kluwer Academic Publishers, 1999
- [6] Iglesias A, Arcay B, Cotos J M, et al. A comparison between functional networks and artificial neural networks for the prediction of fishing catches[J]. Neural computer & applied, 2004, 13:24-31
- [7] Fontenla-Romero O, Castillo E, Alonso-Betanzos A. A Measure of Fault Tolerance for Functional Networks[J]. Neurocomputing, 2004, 62:327-347
- [8] Zhou Yongquan, Jiao Licheng. Interpolation Mechanism of Functional Networks[J]. Lecture Notes in Computer Science, 2005, 369:45-51
- [9] Zhou Yongquan, Jiao Licheng. Approximate Factorization Learning Algorithm of Multivariate Polynomials Based on Functional Networks[J]. Journal of Information and Computational Science, 2005, 2(1): 205-210
- [10] Zhou Yongquan, He Dexu, Jiao Licheng. A Neural Computation Structure for Solving Functional Equations with Functional Networks[J]. Journal of Information and Computational Science, 2005, 2(4): 835-842
- [11] He Dexu, Nong Zheng, Zhou Yongquan. Approximate Factorization of Univariate Polynomials Using Functional Networks[J]. Journal of Information and Computational Science, 2005, 2(3): 473-479
- [12] Zhou Yongquan, Jiao Licheng. One-variable interpolation function based on functional networks[J]. International Journal of Information Technology, 2006, 12(2): 120-129
- [13] 周永权, 焦李成. 层次泛函网络整体学习算法[J]. 计算机学报, 2005, 28(8): 1277-1286
- [14] 周永权, 赵斌, 焦李成. 一种复值泛函网络逼近理论及学习算法[J]. 系统工程与电子技术, 2006, 28(8): 1244-1248
- [15] 周永权, 焦李成. 基于泛函网络的多维函数逼近理论及学习算法[J]. 系统工程与电子技术, 2005, 27(5): 906-909
- [16] 周永权, 赵斌, 焦李成. 序列泛函网络模型及其学习算法与应用[J]. 计算机学报, 2008, 31(7): 1073-1081
- [17] 周永权, 吕咏梅, 申芸. 正交泛函网络函数逼近理论及算法[J]. 计算机科学, 2009, 36(1): 138-141
- [18] 周永权, 刘晓冀, 张明. 基于多项式基函数的泛函网络构造方法与逼近理论[J]. 计算机科学, 2008, 35(08): 119-122
- [19] 张铃, 张钊. 遗传算法机理的研究[J]. 软件学报, 2000, 11(7): 945-952
- [20] 云庆夏. 进化算法[M]. 北京: 冶金工业出版社, 2000 (5/6): 451-495
- [7] De Kleer J. Focusing on probable diagnoses[C]//Proc. 9th National Conference on Artificial Intelligence. 1991:842-848
- [8] Poole D. A logical framework for default reasoning[J]. Artificial Intelligence, 1988, 36(1): 27-47
- [9] Kohlas J, Monney P A, Anrig B, et al. Model-based diagnosis and probabilistic assumption-based reasoning[J]. Artificial Intelligence, 1998, 104(1): 71-106
- [10] Heidtmann K D. Smaller Sums of Disjoint Products by Subproduct Inversion[J]. IEEE Transactions on Reliability, 1989, 38(3): 305-311
- [11] Kohlas J. Mathematical foundations of evidence theory [C] // Coletti G, Dubois D, Scozzafava R, eds. Mathematical Models for Handling Partial Knowledge in Artificial Intelligence. New York: Plenum Press, 1995: 31-64
- [12] Kohlas J, Haenni R. Propositional information systems[J]. Journal of Logic and Computation, 1999, 9(5): 651-681
- [13] Laskey K B, Lehner P. Assumptions, Belief and Probabilities [J]. Artificial Intelligence, 1989, 41(1): 65-77
- [14] Provan G M. A Logic-based Analysis of Dempster-Shafer Theory[J]. International Journal of Approximate Reasoning, 1990, 4 (5/6): 451-495
- [15] 邓勇. 基于模型诊断的贝叶斯解释及应用[J]. 上海交通大学学报, 2003, 37(1): 5-9
- [16] Tuhim S. An experimental study of criteria for hypothesis plausibility[J]. Journal of Experimental and Theoretical Artificial Intelligence, 1991, 3(2): 129-144
- [17] Pearl J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference[M]. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1988: 120-145
- [18] Ahmed Y, Neufeld E. Model-based Diagnosis: A Probabilistic Extension[J]. Lecture Notes in Computer Science, 1998, 1455(1): 379-396
- [19] Flesch I, Lucas P. Conflict-based Diagnosis: Adding Uncertainty to Model-based Diagnosis[C]//Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI). 2007: 380-385
- [20] Flesch I, Lucas P. The Probabilistic Interpretation of Model-based Diagnosis[C]//European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty. 2009: 204-215