

基于遗传规划实现泛函网络神经元函数类型优化^{*}

周永权^{1,2} 焦李成²

(西安电子科技大学智能信息处理研究所 西安 710071)¹

(广西民族大学计算机与信息科学学院 南宁 530006)²

摘要 泛函网络是最近提出的一种对神经网络的有效推广。与神经网络不同,它处理的是一般的泛函模型,其神经元函数不固定,而是可学习的,且在各个处理单元之间没有权值。同神经网络一样,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。鉴于此,将整个泛函网络的设计分解为单个神经元的逐个设计;然后,在此框架下提出了基于遗传规划的单个神经元的设计方法,该方法可实现对神经元函数类型的优化。仿真实验表明,本方法是有效可行的,能用较小的网络规模获得更满意的泛化特性。

关键词 泛函网络,构造性设计,遗传规划,神经元函数,基函数

Optimizing Neuron Function Types Based on GP in Functional Network Design

ZHOU Yong-Quan^{1,2} JIAO Li-Cheng¹

(Institute for Intelligence Information Processing, Xidian University, Xi'an 710071)¹

(College of Computer and Information Science, GuangXi University for Nationalities, Nanning 530006)²

Abstract Functional network is a recently introduced extension of neural networks. Like neural networks, it deals with general functional models instead of sigmoid-like ones. And in these networks there are no weights associated with the links connecting neurons. Nowadays, there is no system designing method for designing approximation functional networks structure. So, a frame of the constructive functional network design is given, in which the design of the whole functional network breaks down to the design of single neurons one by one. Then, based on GP the design algorithm of single neuron, which realizes the auto-optimization of neuron function types, is proposed. Finally, with many function approximation experiments, it is shown that the proposed constructive functional network design scheme is feasible. Being able to achieve better Functional network generalization with small network size.

Keywords Functional network, Constructive design, Genetic programming, Neuron function, Basis function

1 引言

泛函网络是 Castillo 在 1998 年提出的一种对神经网络的有效推广。与神经网络不同,泛函网络处理的是一般泛函模型,它在各个神经元之间的连接无权值,并且神经元函数不固定,而是可学习的,常常是一个给定的函数簇(如多项式、三角函数、Fourier 展开级数等)。泛函网络已经成功地被应用于混沌时间序列的预测,微分、差分和泛函方程的求解, CAD, 线性、非线性回归^[2], 非线性系统辨识^[3,4]等;泛函网络在解决上述问题中都表现出较好的性能。

然而,泛函网络设计也是一个较为复杂的问题,其复杂性重要表现在 3 个方面:第一,采用什么类型的泛函网络结构,往往由人类专家根据经验知识作出判断;第二,泛函网络的结构和泛函参数的确定,这一点往往很难由人工确定;第三,在网络的结构确定的基础上,再用基于梯度法^[3]的学习算法来确定最优泛函参数,同样存在局部极小问题。而遗传规划^[5](genetic programming, GP)为解决这些问题提供了一条新的更有潜力的途径,其最大的特点是采用层次化的结构表达问题,它类似于计算机程序分行或分段描述问题。这种广义的

计算机程序能根据环境状态自动地改变程序的结构和大小,尤其是其符号回归技术(Symbolic Regression)更是直接对数学表达式进行操作,按算法树的遍历进行编码,通过对输入输出数据分析建立描述复杂系统的数学表达式模型。基于这些特性,本文将遗传规划与泛函网络有机地结合起来,将在泛函网络构造性设计框架下讨论基于遗传规划实现泛函网络神经元函数自动设计与优化方法。在泛函网络中目前应用最广泛的是可分离(separable)泛函网络^[6],它的泛函表达式是各个输入变量分离作用效果的组合,本文只针对可分离泛函网络进行讨论。

2 泛函网络简介

一般的泛函网络^[1]由以下元素组成:(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。输出单元也是带有相应的名字来表示;(3)一层或多层神经元。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

^{*} 本文受到国家自然科学基金[60461001]和广西自然科学基金[0542048]项目的资助。周永权 教授,博士,研究方向:计算智能、神经网络及应用;焦李成 教授,博士生导师,研究方向:神经网络、自然计算等。

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来确定,它由一函数定义。例如,假如有一个神经元具有 s 个输入 $(x_1, x_2, \dots, x_s)$ 及 k 个函数 $f_j, j=1, 2, \dots, k$, 使得 $y_j = f_j(x_1, x_2, \dots, x_s); j=1, 2, \dots, k$ 。函数 f_j 由网络的结构来确定,神经元由带有相应 f_j 函数的名称用圆圈来表示;(4) 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计算能力。

泛函网络能够扩展信号处理的范围,它是对神经网络的一种直接推广。图 1(a) 给出了一简单的泛函网络结构及对应神经网络结构(图 1(b))。图中的输入模式是 $\{x_1, x_2, x_3\}$, 神经元函数 $f_i (i=1, 2, 3)$ 输出 x_6 。特别神经元函数 $f_i (i=1, 2, 3)$ 的选择在泛函网络中,根据 Castillo 的做法是将每一神经元函数 f_i 表示成一些已知函数簇的线性组合的形式。如: 多项式、三角函数、Fourier 展开级数等。在神经网络中,对每一神经元函数 f_i , 即激活函数常取 Sigma 函数、双曲正切函数等。它与泛函网络的神经元函数区别在于神经元函数 f_i 是固定的,不可学习的。其次,泛函网络没有权值,而神经网络是带权值等等。至于其它区别,本文不再一一赘述,可参考文献[4]。

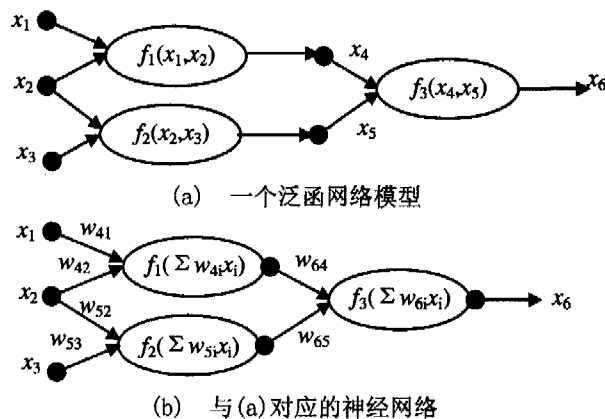


图 1

3 可分离泛函网络模型

泛函网络与神经网络一样,也有各种各样的结构,人们不可能用一个统一的通用的结构来描述所有的泛函网络,也不可能用一个统一的函数来表示所有的泛函网络。在泛函网络应用研究中,应用最广泛的是可分离(Separable)泛函网络^[6],它的泛函表达式是各输入变量分离作用效果的组合。下面介绍两种可分离泛函网络模型。

3.1 多输入、单输出可分离泛函网络模型

不妨考虑两个输入一个输出的可分离泛函网络模型(图 2)。

图 2 画出的是一类两个输入 x, y , 一个输出 z 可分离泛函网络模型。该网络的输出有如下表达式:

$$z = F(x, y) = \sum_{i=1}^n f_i(x) g_i(y) \quad (1)$$

引理 1^[6] 方程 $\sum_{i=1}^n f_i(x) g_i(y) = 0$ (写成向量的形式为 $f^T(x) g(y) = 0$) 的所有解可以表示成 $f(x) = A\phi(x), g(y) = B\varphi(y)$ 的形式,其中 A, B 分别为 $n \times r$ 和 $n \times n-r$ 常数矩阵,且 $A^T B = 0, \phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_r(x)), \varphi(y) = (\varphi_1(y), \dots, \varphi_{n-r}(y))$ 。

$\varphi_{r+1}(y), \dots, \varphi_n(y), \phi(x), \varphi(y)$ 是相互线性独立的函数集, $0 \leq r \leq n$ 。

再考虑另一种多输入、单输出可分离泛函网络模型(图 3)。

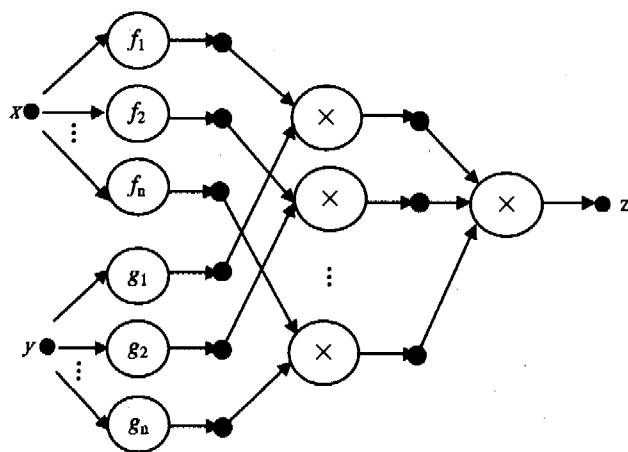


图 2 一类可分离泛函网络模型

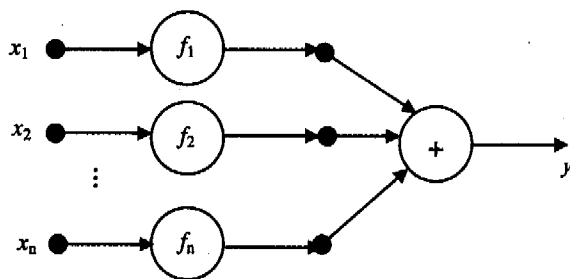


图 3 另一类多输入、单输出可分离泛函网络模型

该网络的输出有如下表达式:

$$y = F(x_1, x_2, \dots, x_n) = \sum_{i=1}^n f_i(x_i) \quad (2)$$

引理 2^[6] ($y = F(x_1, x_2, \dots, x_n) = \sum_{i=1}^n f_i(x_i)$ 表达式唯一性) 如果 $y = F(x_1, x_2, \dots, x_n)$, 有两种表达式:

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^n f_i(x_i) = \sum_{i=1}^n f'_i(x_i) \quad (3)$$

则有以下关系成立:

$$\begin{aligned} f'_1(x_1) &= f_1(x_1) + c_1, f'_2(x_2) = f_2(x_2) + c_2, \dots, f'_{n-1}(x_{n-1}) = f_{n-1}(x_{n-1}) + c_{n-1}, \\ f'_n(x_n) &= f_n(x_n) + (c_1 + c_2 + \dots + c_{n-1}) \end{aligned}$$

式中 $c_i, i=1, 2, \dots, n$ 是任意的常数,由给定的泛函网络的初始值确定。

事实上,由于 $f_i(x_i) (i=1, 2, \dots, n)$ 各个函数的输入变量不同及相互之间彼此线性无关性,我们很容易得出上述结论,该结论给以下基于 GP 的单个神经元优化设计奠定了理论基础。

进一步设

$$y_i = f_i(x_1, x_2, \dots, x_n), i=1, 2, \dots, N \quad (4)$$

则图 3 可分离泛函网络的输出为

$$y = \sum_{i=1}^N y_i \quad (5)$$

也就是说,可分离泛函网络可分解成如式(5)所示的 N 个神经元组合,每个神经元,根据 Castillo 的做法,是将每一神经元函数 f_i 表示成一些已知基函数簇的线性组合的形式,如在模型 3 中,每一神经元函数 f_i 可写成

$$f_i(x) = \sum_{j=1}^n a_{ij} \varphi_{ij}(x) \quad (6)$$

其中, a_{ij} 泛函参数, $\varphi_{ij}(x)$ 为基函数簇, 两者都可通过学习而获得。这样, 整个可分离泛函网络的构造性设计可变成单个神经元逐个优化的问题。

一般地, 可分离泛函网络的构造性设计步骤如下:

① 生成第一个神经元, 将初始的样本集 P 和目标集 T 置为当前训练集;

② 用当前训练集训练新生成的神经元。本文将基于 GP 进行设计, 直至训练误差达到要求, 训练停止, 则整个设计过程结束, 否则转③;

③ 对刚训练好的神经元, 输入 P 中各矢量 $x_1, x_2, \dots, x_m$, 计算其实际输出 $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_m$, 并将 T 调整为 $T' = \{y_j = y_j - \hat{y}_j, j=1, 2, \dots, m\}$;

④ 生成一个新的神经元。将 P 和调整为 T' 置为当前训练集, 转向②。

4 基于 GP 的单个泛函神经元优化设计

所谓单个泛函神经元设计, 就是根据当前训练集优化确定(6)式中各项待定的参数, 其中关键是处理的泛函神经元函数类型的选取与优化, 其中基函数 $\{\varphi_{ij}(x) | j=1, 2, \dots, n\}$ 可以是一些标准的数学函数与若干个运算符构成的一些表达式。遗传规划正是在数学表达式优化方面有着独特优势的一种优化方法。

遗传规划优化的一般步骤是: ①根据问题定义适当的终止符集、函数(运算符)集、目标函数和适应度函数; ②根据函数集和终止符集随机地生成若干个数学表达式(个体)组成父代种群; ③根据种群中各个体的适应度值, 对父代种群进行选择、交叉、突变等遗传操作产生新一代群体; ④若新一代群体中有满足要求的个体或迭代次数条件, 则停止, 否则将新一代种群置为父代种群转③。

4.1 终止符集、函数(运算符)集定义

在基于 GP 的泛函神经元优化设计中, 终止符集就是由若干个标准数学函数类型(如 $\text{pow}, \exp, \log, \sin, \cos, \dots, x_1, x_2, \dots, x_n$) 构成的一个函数库, 可以把任何标准数学函数都放入到这个库中去, 库中函数愈多, 愈有利于构造出与问题匹配的泛函神经元类型。根据泛函神经元的特点, GP 优化单个泛函神经元函数类型的运算符集包括 $+, -, \times, /$ 。因为每个神经元函数可由一些标准数学函数的线性组合。

4.2 目标函数、适应度函数的定义

一般地, 对于训练神经元而言, 训练目标是理想输出与实际输出的均方差 RMSE 最小。我们不妨定义待训练神经网络的目标函数为

$$y_i = F(x_1, x_2, \dots, x_n) = \sum_{j=1}^n f_j(x) \quad (7)$$

式中 $f_i(x) = \sum_{j=1}^n a_{ij} \varphi_{ij}(x)$, a_{ij} 为泛函参数, $\{\varphi_{ij}(x) | j=1, 2, \dots, n\}$ 为基函数集。适应度可定义为

$$\text{Fint} = \text{RMSE} = \frac{1}{m} \sqrt{\sum_{j=1}^m (y_j - \hat{y}_j)^2} \quad (8)$$

式中, m 是学习样本总数, y_j 和 $\hat{y}_j$ 分别是泛函网络的预测输出和实际输出。在实际应用中, 为了检验泛函网络的逼近能力, 人们将学习样本分成两部分, 一部分用于计算适应值, 另一部分用于测试样本。

4.3 GP 中神经元个体的描述

在 GP 中每一代包含众多的个体, 组成这一代的群体。每一个体用广义的层次状计算机程序表达, 它由函数集 F 及终止符集 T 组成。函数集 F 包含 n 个函数:

$$F = \{f_1, f_2, \dots, f_n\} \quad (9)$$

函数集内的函数 f_i 可以是运算符, 如 $+, -, \times, /$ 等算术运算符和标准数学函数, 如 $\sin, \cos, \log, \exp, \dots$ 。若令 $G = \{\text{算术运算符}\}, H = \{\text{标准数学函数}\}$, 则有

$$F = G \cup H \quad (10)$$

终止符集 T 包含 m 个终止符:

$$T = \{t_1, t_2, \dots, t_m\} \quad (11)$$

终止符集中终止符可以是变量, 如 $x, y, \dots, z$; 或常量如 $a, b, \pi, \dots$; 同样若令 $U = \{\text{变量}\}, V = \{\text{常量}\}$, 则有

$$T = U \cup V \quad (12)$$

用 GP 对单个神经元进行优化, 每给定神经元一组式(4)、(6)所示的待定参数就产生了该神经元的一个个个体, 在本问题中, 染色体不再是传统的二进制位串, 而是采用实数直接编码表示, 神经元函数的类型是若干个标准数学函数的线性组合形式, 显然, 神经元的整个个体在程序中只能表示为一个结构体。我们在分析时把函数的表达式用数据结构中采用的方法——二叉树表示, 节点集合 $= F$, 叶节点 $= T$, 进而把基于实数编码的遗传操作改进为针对二叉树的遗传操作。我们由运算的优先级知道, 一个函数的二叉树表示可能不是惟一的, 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惟一的。基于这些特性, 我们利用二叉树作为算法树, 容易直观地完成对 GP 中复制、交换和突变算子的操作。

4.4 GP 的进化策略

由于神经元的个体是一个含有标准数学函数的复杂结构体, 为了方便, 可不使用交叉操作, 而使用 2 种变异操作, 完全变异操作和参数调整操作。前者随机地产生与父代个体完全不同的全新个体, 以取代种群中劣质个体; 后者只是针对优良父代个体的操作, 不变化其神经元函数的类型, 只在小范围内对其他参数进行调整产生新个体。将选择操作和上述变异操作相结合则实现进化, 直至获得最优群体。

5 仿真实验

实验中泛函网络神经元函数类型库由 $F = G \cup H$, 其中 $G = \{+, -, \times, /, \dots\}$, $H = \{\sin, \cos, \tan, \text{ctg}, \ln, \dots\}$ 组成, 实际中可以根据需要任意增减, 增减数据库中的函数类型与方法本身无多大的关系; 其次终止符集 $T = U \cup V$; 其中 $U = \{x, y, \dots, z\}$; $V = \{a, b, \dots, c, \pi, \dots\}$, 同样实际中可以根据需要任意增减, 也与方法本身关系不大。

5.1 实验 1

实验 1 测试相同网络规模下选取不同函数基的逼近性能。用本文方法进行典型的可分离泛函网络构造性设计, 当训练误差达到要求精度, 停止训练并记下此时所需神经元个数及构成每一神经元函数所需基函数。测试函数为: ① $1 + x + y^2 + xy^2$; ② $x^2 + 4\sin(y)$; ③ $0.1\sin x + \lg(\sin y)$ 。训练样本数据用函数 $\text{Random}[]$ 随机产生, 样本数据个数 ≥ 10 , 训练误差用适应度(8)式计算, 即均方差来计算, 仿真结果见表 1。

5.2 实验 2

某物理实验中, 使电流 i 通过 2Ω 的电阻, 用伏特表测得电阻两端的电压为 V , 多次实验测得的数据如表 2 所示, 要求找出 i 与 V 的关系式 $\varphi(i)$, 使其满足 $V = \varphi(i)$ 。

(下转第 16 页)

生成集合 $S_{out}^B - S_{in}^A$, 即推导出在实时合成构件语义规约中作为合成构件服务提供端口的构件方法描述体原子进程集合的组成部分。

由此, 实时合成构件 $A+B$ 的语义规约进程可描述为 $Component_{realtime}^{A+B} = \square [(S_{in}^A - S_{out}^B) \cup (S_{in}^B - S_{out}^A)]; \square [S_{com}^A \cup S_{com}^B \cup (S_{in}^A \cap S_{out}^B) \cup (S_{out}^A \cap S_{in}^B)]; \square [(S_{out}^A - S_{in}^B) \cup (S_{out}^B - S_{in}^A)]$, 其中“ $\square$ ”为进程间并发操作符, 如“ $\square S$ ”表示集合 S 中各元素进程间的并发行为。另外, 在实时合成构件 $A+B$ 的语义规约中带时间约束特征的方法描述体 rtM 仍是构件规约描述的基本载体, 那么实时合成构件 $A+B$ 的服务请求端口、服务提供端口以及内部行为语义的实时性, 则是由这些方法描述体 rtM 偏序构成的实时迹 (trace) 来刻画与计算的。

结束语 软件合成是以构件合成为基础, 构件合成又是以构件自身外部接口间的连接为操作单元的, 所以在软件合成之前, 设计人员有必要从构件库中提取构件实现体相关的语义配置信息, 从实时构件的语义规约描述层来分析构件自身外部接口间的连接性, 以及实时构件间的可适配性, 以判断

软件合成的可行性。因此, 本文中我们首先建立了一个实时构件反射式语义模型 RCRSM, 用于对实时构件合成进行理论分析, 验证实时构件间的适配性, 并由此规约实时合成大粒度构件的语义描述机制, 同时从语义层支持构件的演化, 如此提高了软件复用的效率, 降低了合成软件的不稳定性。

参考文献

- 1 黄靖, 赵海光. 软件复用、软件合成与软件集成. 计算机应用研究, 2004(9)
- 2 黄靖, 卢炎生, 徐丽萍. 基于反射技术的有性能约束特征的软件集成方法研究. 小型微型计算机系统, 2005, 26(7)
- 3 黄靖, 卢炎生, 徐丽萍. 基于反射的实时构件模型规约描述研究. 计算机科学, 2006(10)
- 4 Szyperski C. Component Software: Beyond Object-Oriented Programming. Addison-Wesley, 1998
- 5 Hoare C A R. Communicating Sequential Processing. Prentice Hall, 1985

(上接第9页)

表1 相同网络规模下泛函神经元函数的逼近性能

测试函数	近似神经元函数		训练误差 RMSE	神经元个数
	$f_1(x)$	$f_2(x)$		
①	$\{1, x, x^2\}$	$\{1, y, y^2\}$	0	2
②	$\{1, x^2\}$	$\{1, \sin y\}$	0	2
③	$\{1, \sin x, \cos x, \lg(\sin x)\}$	$\{y, \lg(\sin y)\}$	9.47857×10^{-6}	2

表2 电流 I 与电压 V 的数据关系

i/A	1	2	4	6	8	10
V/V	1.8	3.7	8.2	12.0	15.8	20.2

当取 $F=\{+, -, *, \sin, \cos, \exp, \ln\}$, $T=\{i\}$ 时, 树深为 7 时, 神经元个数为 1 时, 计算得到精度高的模型:

$$V = 0.05133534253 \sin i^2 \times 1.990574910i + 0.0016913771i^2 - 0.01404056012 - 0.03087686991 \cos 2i - 0.003207754783i \times \sin i - 0.2870149533 \sin i.$$

其均方差为 $RMSE = 0.9414691249e-13$. 其拟合的准确度易从图 5 看出。

V (电压)

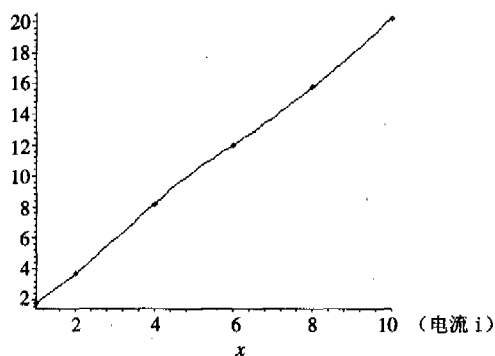


图5

该模型是对多个简单函数复合而成复杂函数, 它的拟合精度比线性函数的拟合精度要高得多。

结论 GP 是一种复杂的进化计算方法, 不如遗传算法那样成熟和完善, 至今仍无法从理论上严密预测优良模式如何在运行中壮大发展, 这主要是由于遗传规划的个体表达式经常变化其结构及大小。本文方法的最大特点是: 提出的可分离泛函网络的设计框架将整个网络设计分解成单个神经元的设计, 使基于 GP 优化神经元函数的类型具有较强的可操作性。由于神经元函数的类型库基于标准数学函数和运算符构成, 因此 GP 实际随机搜索的函数类型空间很大, 在实际应用中, 我们可根据实际需要加以选取, 这使得设计出的网络的泛化性更有保证。总之, 本文提出的方法是可行、有效的。

参考文献

- 1 Castillo E. Functional Networks. Neural Processing Letters, 1998, 7: 151~159
- 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, 2000, 11(1): 39~49
- 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, 2004, 13: 24~31
- 5 Koza J. Genetic programming: on the programming of computers by means of natural selection [M]. Cambridge, MA: The MIT Press, 1992, 1~45
- 6 Castillo E, Gutierrez J M. Nonlinear time series modeling and prediction using functional networks, Extracting Information Masked by Chaos [J]. Phys Lett A, 1998, 244: 71~84
- 7 张铃, 张钊. 遗传算法机理的研究. 软件学报, 2000, 11(7): 945~95
- 8 Zhou Yong-Quan, Jiao Li-Cheng. Interpolation Mechanism of Functional Networks. Lecture Notes in Computer Science, 2005, 3697: 45~51
- 9 Zhou Yongquan, Jiao Licheng. Approximate Factorization Learning Algorithm of Multivariate Polynomials Based on Functional Networks. Journal of Information & Computational Science, 2005, 2(1): 205~210
- 10 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