第36卷 第19期 计 算 机 工 程 2010年10月
Vol.36 No.19 Computer Engineering October 2010
·开发研究与设计技术· 文章编号：1000—3428(2010)19—0277—03 文献标识码：A 中图分类号：TP393.02

可重构路由交换平台中的构件分类检索方法

曲 剑，汪斌强，曲 晶，刘 强
(国家数字交换系统工程技术研究中心，郑州 450002)

摘 要：针对可重构路由交换平台构件库中构件检索效率不高的现状，提出一种基于刻面的软硬件构件分类方法。基于该方法设计误差反向传播神经网络构件检索算法，分析表明该算法具有并行性强、效率高、速度快的特点，适合规模大、构成复杂的构件库检索问题。仿真结果证明，在构件数量较多的情况下，该算法性能优于传统检索算法。

关键词：可重构；路由；构件分类；构件检索；神经网络

Component Classification and Retrieval Method on Reconfigurable Routing Switching Platform

QU Jian, WANG Bin-qiang, QU Jing, LIU Qiang
(National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

【Abstract】 Aiming at lower efficiency of components retrieval from component database in reconfigurable network routing platform, a component classification method of software and hardware based on facet is presented. Then a component retrieval algorithm based on Back Propagation Neutral Networks(BPNN) is designed. It is analyzed that the algorithm fits for component database with large size and complicated constitution because of its parallel execution, high efficiency and speed. It is indicated that the algorithm improves the retrieval performance in contrast with most of current retrieval algorithms by the emulation, especially on large size component database.

【Key words】 reconfigurable; routing; component classification; component retrieval; neural network

1 概述

随着互联网的不断发展，面向服务提供的新型网络体系架构成为未来网络的发展趋势。而实现该新型网络技术体系的关键是研究开放式可重构路由交换节点技术。开放式可重构路由交换平台技术的基础是平台化支撑下的构件化处理技术。以可重构开发环境为基础，对构件库进行设计，可以最终实现可重构路由交换平台[1]。

可重构路由交换平台是通过构件的生成、组装和运行实现的。为了支持大规模软件构件、硬件构件[2]构件化，构件库中必须拥有大量构件以供使用。在构件数量达到一定数目时，人力对构件库信息的掌握将无法满足要求，因此，对具备一定规模的构件库进行高效的分类检索成为一个关键的问题。目前缺乏标准、有效的构件分类检索方法，从而导致用户不能确切判断构件库中是否存在合适的构件，给可重构带来很大的障碍。

目前有很多关于构件的分类和检索方法。从构件表示出发可分为基于信息科学的方法、基于人工智能的方法、基于形式化规约的方法3类[3]。

基于人工智能方法的缺点是理解性较差、适用范围受限、不易维护。基于形式化规约的分类检索方法往往有很高的查准率，但当选用的规约语言不当时，查准率会很低。信息科学方法是现阶段构件库管理系统常用的分类检索方式，枚举、刻面、属性/值、关键词方式都属于此方式。其优点就在于分类时添加了分类信息，从而可以提高构件的检索效率。其中，刻面分类方法能够表达丰富的构件信息，为绝大多数构件库所采用。

2 基于刻面的可重构路由交换平台分类方法

刻面分类法由Ranghanathan提出，Diaz最早将该方法应用于资产库中对可复用资产的分类信息进行描述，其刻面分类方案包括2个方面：“功能”和“环境”。刻面分类最早在REBOOT系统中得到成功应用，此后许多构件库系统都将刻面分类作为主要分类方法之一[4]。文献[5]提出用应用领域、表示方法、使用环境、抽象层次、功能5个刻面来描述构件。文献[6]采用了以刻面分类为主、多种分类方式相结合的方法对构件进行分类。文献[7]也针对其特定领域对构件分类方式进行了描述。

鉴于路由领域构件的特点，这些通用的模型不能完全满足路由领域构件描述的需要。为了更加贴切地创建路由领域构件表示模型，本文提出了适合路由领域的分类方法，如表1所示。

如表1所示，该分类模型将构件分为7个刻面，每个刻面又包含不同的子刻面值。下面将这种分类方法用数学语言描述。假设一个构件库中的构件分为 m 个刻面，即 $\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_m$，每个刻面下又有 n 个子刻面，即 $\boldsymbol{F}_i=\{R_{ij}, j=1,2,\cdots,n_i\}$，于是构件的刻面空间可表示为 $\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_m$。表1中的构件分类模型按照这种方法可以对应

基金项目：国家“863”计划基金资助项目(2008AA01A323)
作者简介：曲 剑(1982－)，男，硕士研究生，主研方向：下一代网络架构，构件的分类检索；汪斌强，教授、博士生导师；曲 晶，讲师、硕士；刘 强，博士研究生
收稿日期：2010-04-30 **E-mail：**coolsword@foxmail.com

表示为 $\boldsymbol{F}_1 \times \boldsymbol{F}_2 \times \cdots \times \boldsymbol{F}_7$。

表 1　基于刻面的可重构路由交换平台构件分类方法

刻面	子刻面
开发语言	.NET、JAVA、BASIC、DELPHI、C/C++、ASP/PHP、FORTRAN、PERL/PYTHON/RUBY、VHDL、VERILOG、…
封装形式	ACTIVE/COM/COM+、JAVABEAN、WEB SERVICE、JAVASCRIPT/AJAX、DLL、MFC、VBX、VCL/CLX、CORBA、…
使用环境	WINDOWS、VXWORKS、LINUX、UNIX、ORACLE、MY SQL、MS SQL SERVER、…
功能	配置管理、资源管理、设备及用户状态管理、RIP、OSPF、BGP、IPv4/IPv6、MPLS、Multicast Routing、SNMP、7 号信令功能、链路层分组处理、电路帧适配、转发表管理、差异化报头处理、查表、优先级调度、分组处理、流量整形、队列管理、QoS、网络安全、……
应用领域	互联网、电信网、广电网、……
接口	配置管理接口、软件行为接口、软件型构接口、硬件元构件间的数据接口、……
粒度	二进制串、源代码、可部署实体、……

3　基于 BPNN 的构件检索算法分析

近年来出现了借鉴树匹配(Tree Matching, TM)的相关结论来研究构件检索技术的方法[8]。这种方法将一个构件库中的刻面分类模式映射成一棵刻面树，检索构件时根据用户输入的信息和刻面分类模式生成一棵查询树，于是将构件的检索问题转化为两棵树的匹配问题，这也是目前刻面分类检索方式的主要研究方法。文献[9]将用户在刻面树中选择的节点根据合理的并和交运算的规则，组合生成查询条件，结合数据库检索技术来实现构件的匹配。但这种树结构构件匹配算法复杂度较高，不利于可重构路由交换平台实时性要求。文献[10]提出了基于单层感知器网络的构件提取算法，但如果尝试利用感知器对非线性可分的向量集合进行分类时，学习算法就会处于一种无休止的循环状态，浪费大量的计算资源。鉴于上述算法不足，本文提出了一种适合路由领域的基于误差反向传播神经网络(BPNN)的构件检索算法。BPNN 结构如图 1 所示。

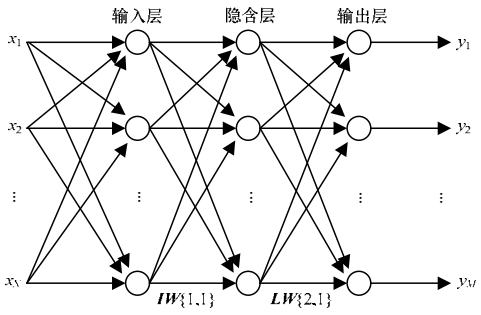


图 1　BPNN 结构

图中 $x_1, x_2, \cdots, x_N$ 为 BPNN 输入向量，$y_1, y_2, \cdots, y_M$ 为 BPNN 输出向量，$\boldsymbol{IW}\{1, 1\}$表示输入层到隐含层的连接权值，$\boldsymbol{LW}\{2, 1\}$表示隐含层至输出层的连接权值。当使用 S 型激活函数时 BPNN 输入输出关系如下：

输入：$net = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n$

输出：$y = f(net) = \dfrac{1}{1+\mathrm{e}^{-net}}$

下面建立构件检索 BPNN 模型。从构件库中检索构件可转化为在构件刻面空间和构件库中各个构件的位置 ID 间建立映射关系。刻面空间中由表示刻面值的神经元组成的集合，可看作输入神经元集合 $\boldsymbol{X}$，构件位置 ID 神经元集合，则看作输出神经元集合 $\boldsymbol{Y}$，这就构成了本文可重构路由交换平台构件库构件检索的 BPNN 模型：

$$\boldsymbol{Y} = \boldsymbol{X} \times \boldsymbol{IW}\{1,1\} \times \boldsymbol{LW}\{2,1\}$$

其中，$\boldsymbol{X}$ 表示输入矩阵；$\boldsymbol{Y}$ 表示输出矩阵；$\boldsymbol{IW}\{1, 1\}$表示输入层到中间层的连接权值矩阵；$\boldsymbol{LW}\{2, 1\}$表示中间层至输出层的连接权值矩阵。

由于表 1 所示的构件分类数量较多，不便于说明算法流程，因此本文简化了构件分类模型，只取开发语言、封装形式、应用领域 3 个刻面。如表 2 所示。

表 2　简化的构件分类

刻面	子刻面
开发语言	JAVA、C/C++、VHDL
封装形式	ACTIVE/COM/COM+、JAVABEAN、CORBA
应用领域	互联网、电信网、广电网

输入 BPNN 的单元长度为 $\sum n_i$，i 表示第 i 刻面，n 表示第 i 个刻面下的子刻面的个数。表 2 所示的分类策略的子刻面值一共有 9 项，将每一个子刻面值由 1~9 编号，由此构成一个 9 位的二进制串。当用户产生查询需求时，1 代表在用户对当前位的刻面值有查询需求，0 代表无查询需求。例如第一个子刻面值 Java 位置为 1，那么可将它表示为 100 000 000。由此可以确定 BPNN 的输入层的维数为 9，9 个输入单元为 $x_1, x_2, \cdots, x_9$。

刻面空间对应于 BPNN 的输入构件库中构件的存放位置 ID 对应于 BPNN 的输出，假设构件库中有 9 个构件，期望输出向量被定义成一个变量，期望输出向量含有 9 个元素，构件在构件库中所在位置处元素为 1，其余为 0。例如，构件库中第 1 个构件的期望输出向量为 100 000 000。由此可以确定 BPNN 的输出层的维数为 9，9 个输出单元为 $y_1, y_2, \cdots, y_9$。

在用户进行构件检索前，要先对网络进行训练。通过 BPNN 把用户查询需求子刻面编码作为人工神经网络的输入矩阵，把构件的位置 ID 作为构件的输出矩阵，经过一定次数的训练就可以把构件位置 ID 和用户查询子刻面的关系映射到权重矩阵 $\boldsymbol{IW}\{1, 1\}$、$\boldsymbol{LW}\{2, 1\}$中。当查询用户产生新的需求时，通过该 BPNN 就可以迅速检索到所需构件。

4　BPNN 构件检索算法仿真及性能对比

根据 Kolmogorov 定理[11]，采用一个 $N \times 2N + 1 \times M$ 的 BPNN 结构。其中，N 表示输入特征向量的分量数；M 表示输出状态类别总数。如前所述，输入和输出层数皆为 9，N=9，M=9。由此可得，该 BPNN 结构为：输入层有 9 个神经元，中间层有 19 个神经元，输出层有 9 个神经元。隐含层和输出层的神经元传递函数均为 logsigmoid()，这是因为 logsigmoid()函数输出量在(0, 1)区间内，恰好满足输出布尔值的要求。

由此可以确定构件检索 BPNN 模型，如图 2 所示。图中 $\boldsymbol{IW}\{1, 1\}$表示输入层到中间层的连接权值矩阵，$\boldsymbol{LW}\{2, 1\}$表示中间层至输出层的连接权值矩阵，$\boldsymbol{b}\{1\}$表示中间层各单元的输出阈值矩阵，$\boldsymbol{b}\{2\}$表示输出层各单元的输出阈值矩阵。其中，$\boldsymbol{X}$ 是一个 1×9 的矩阵；$\boldsymbol{IW}\{1, 1\}$是一个 9×19 的矩阵；$\boldsymbol{LW}\{2, 1\}$是一个 19×9 的矩阵；$\boldsymbol{Y}$ 是一个 1×9 的矩阵。

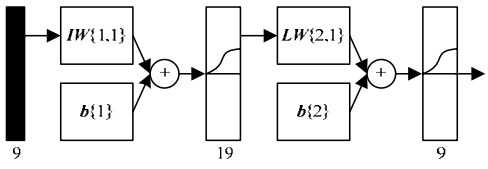


图 2　BPNN 构件检索模型

网络创建后，必须先经过训练并且达到要求后，才可以使用。假设获得的构件库构件输入输出样本如表 3 所示。

表 3　BPNN 训练学习样本

构件在构件库中的位置 ID	输入向量 $x_1, x_2, \cdots, x_9$	输出向量 $y_1, y_2, \cdots, y_9$
1	001 001 001	100000000
2	100 001 010	010000000
3	001 001 100	001000000
4	010 001 010	000100000
5	001 010 010	000010000
6	001 010 100	000001000
7	001 100 001	000000100
8	100 010 010	000000010
9	001 100 010	000000001

BPNN 的训练函数分别采用拟牛顿法、一步正切拟牛顿法和共轭梯度法。拟牛顿法的迭代方程为

$$\Delta w(k+1) = w(k) - D^{-1}\nabla f(w(k))$$

共轭梯度法的迭代方程为

$$\Delta w(k+1) = w(k) + \rho(k)S(k)$$

$$S(k) = -\nabla f(w(k)) + v(k-1)S(k-1)$$

$$v(k-1) = \frac{\|\nabla f(w(k))\|^2}{\|\nabla f(w(k-1))\|^2}$$

采用 2 种方法训练得到训练曲线，如图 3、图 4 所示。由图 3 可知，拟牛顿法经过 361 次训练后算法收敛；由图 4 可知，共轭梯度法经过 263 次训练后网络的目标误差达到要求。可见，在本例中，共轭梯度法收敛速度更快，更适合构件检索实时性需求。

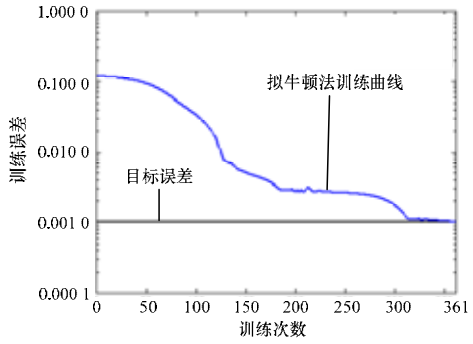


图 3　拟牛顿法训练曲线

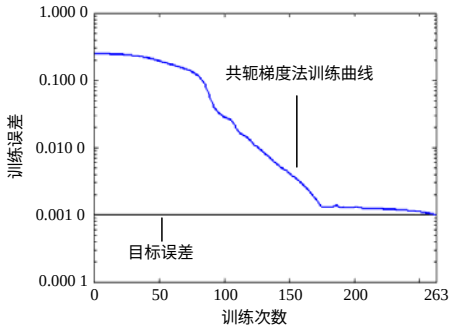


图 4　共轭梯度法训练曲线

网络训练结束后，利用另外一组数据对网络进行测试，检查输出是否满足用户查询需求。比如某查询用户指定检索构件库子刻面值为 Java、JAVABEAN、互联网，将查询需求转换为输入向量 100 010 100，将其输入训练好的 BPNN，经过计算，得到输出向量：[0.001 8, 0.104 5, 0.995 4, 0.048 9, 0.100 3, 0.008 5, 0.043 1, 0.000 8, 0.009 4]。说明构件库中的第 3 个构件被识别出来，与预期一致。

为了测试和验证 BPNN 构件检索算法的性能，本文测出一组数据，将 BPNN 算法与现有主流树匹配(TM)构件检索算法[8]做性能比较。对 2 种算法的检索时间做数据统计，结果如图 5 所示。

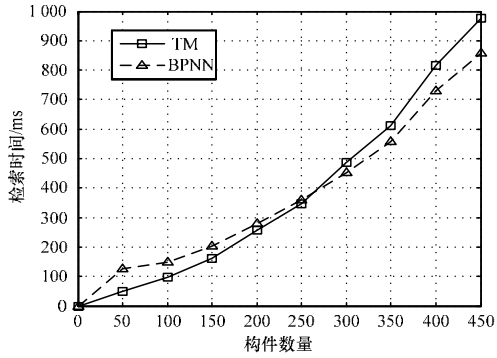


图 5　BPNN 算法与 TM 算法的性能比较

从图 5 中可以看出，在构件数量小于 270 时，完成检索所需要的时间 TM 检索算法较优，但当构件数量超过 270 时，BPNN 检索算法就凸显出其优势。随着构件数量的增多，这种优势将越来越明显。由此可以预见，在构件库中构件数量巨大的情况下，BPNN 算法的性能比现有主流算法有较大提升，在大容量构件库中采用 BPNN 算法可大量节省用户检索时间。

5　结束语

本文在总结现有构件分类、检索方法的基础上，提出了适合于可重构路由交换平台的基于刻面的构件分类方法和 BPNN 可重构路由交换平台软硬件构件检索方法。经仿真验证，在构件数量较多的情况下，此方法比现有主流构件检索算法性能有较大改善，且构件数量越多，其性能提升越明显。且此方法能够满足路由交换平台查全率、查准率、实时性等特定需求，达到了预期设计目标。

由于 BPNN 易陷于局部最优，且该方法未能体现不同用户对构件刻面值选择的优先级概念，下一步拟针对可重构路由交换节点需求完善构件分类方案，并通过遗传算法、模糊神经网络对 BPNN 构件检索算法做进一步改进。除此之外，将算法应用于工程实际，开发出适合不同用户需求的可重构路由交换平台构件检索原型系统，也将是未来工作的重点。

参考文献

[1] 汪斌强. 新一代高可信网络——可重构路由器构件组研制[R]. 郑州：国家数字交换系统工程技术研究中心，技术报告：2008AA01A323, 2008.

[2] 黄万伟, 兰巨龙, 于　婧, 等. 硬件构件的形式化描述及其组装机制[J]. 计算机工程, 2010, 36(8): 230-232.

[3] 谢　冰, 王亚沙, 李　戈. 面向复用的软件资产与过程管理[M]. 北京: 清华大学出版社, 2008.

[4] Morel J M, Faget J. The REBOOT Environment[C]//Proc. of the 2nd International Workshop on Software Reusability. Lucca, Italy: [s. n.], 1993.

[5] 常继传, 李克勤, 郭立峰, 等. 青鸟系统中可复用软件构件的表示与查询[J]. 电子学报, 2000, 28(8): 20-24.

[6] 北京大学软件工程研究所. 青鸟构件库概念模型[EB/OL]. (2000-03-10). http://sei.pku.edu.cn/jadebird/index.html.

（下转第 282 页）