

PCA-BP 算法模块化设计的编程实现

贾 群

(淮南师范学院 计算机与信息工程系, 安徽 淮南 232001)

摘 要: BP算法存在收敛性慢、逼近精度差等缺点。因此,在实际应用中往往需要对BP算法进行相应的改进。利用主元分析法(PCA)对数据样本预先进行降维处理,消除数据间的强耦合性,减少模型的复杂性,然后再作为BP的输入样本从而提高模型的解算速度。在此分析的基础上利用Visual Basic采用模块化的设计方法实现对PCA-BP算法的编程,使学习速率能够进行相应自调整和优化,以此来提高BP网络的泛化推广能力,并能够满足设定的误差精度从而达到现场实际运用需要的目的。

关键词: BP; PCA-BP算法; 算法编程

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2008)12-0098-04

Realizing PCA-BP Algorithm by Modular Design

JIA Qun

(Department of Computer and Information Engineering, Huainan Normal University, Huainan 232001, China)

Abstract: Traditional BP algorithm has slow convergence and weak approximability. Therefore, must improve BP algorithm accordingly. Introduces the method that can reduce dimensions of datum by PCA and eliminate the strong coupling metric, reducing complexity of model, the result is the input stylebook of BP in order to improve the speed of calculation. Depending on the base of analysis, can realize PCA-BP algorithm program by Visual Basic, accelerating and optimizing the rate of algorithm learning, in order to improve the generalization and satisfy given error precision reaching to practical application ability of BP network.

Key words: BP; PCA-BP algorithm; algorithm program

0 前 言

神经网络的理论研究是一门新兴的边缘和交叉学科,它的产生和发展一方面受到其它学科的影响,同时又在很大程度上影响和促进其它学科的进一步发展。从信息论的角度上来看,它是一种信息处理工具,它涉及到物理学、数学、信息学、认知学、电子学等学科,从应用的角度上来看,神经网络直接影响到系统论、控制论、协同论等现代科学理论,所以说研究人工神经网络具有很重要的现实意义^[1]。

人工神经网络的模型共有40多种,可以从不同的角度进行分类。按网络的结构可以分为前馈型和反馈型网络;按网络的性能可以分为连续性和离散性、确定性和随机性网络;按照学习方式可以分为有导师指导型和无导师指导型(包括自组织学习型)等网络;按照突触连接性可以分为一阶线性关联和高阶非线性关联

网络^[2]。

在这些算法中BP算法适用范围最广,其模型简单,易于扩展,可以对非线性动力学系统进行定性定量的分析^[3]。通过适当的设计和编程,可以被应用于实际生产当中,以提高工艺生产的精度和效率。

1 基于PCA-BP网络模型的确立

主元分析(回归分析)PCA(Principal Component Analysis)是将多个指标化为少数指标的一种统计和分析方法,PCA是一种经典的统计回归方法,是进行数据压缩和信息抽取强有力的工具^[4]。它是将所要研究对象的多个相关变量转化为较少的、互不相关的综合变量,以消除因观测数据具有一定的相关关系而反映的信息上的重迭,其目的是简化数据结构及揭示变量之间的关系,为全面分析和解决问题而减少模型的复杂性,从较多的指标中找出较少的几个综合指标,使这几个综合指标能够最大程度上反映原先指标上的信息。经过PCA降维以后的样本空间能够真实有效地反映原始样本数据的信息,忽略冗余信息的影响,有利

收稿日期:2008-04-14

基金项目:安徽省高校青年教师科研资助计划项目(2005jq1232)

作者简介:贾 群(1976-),男,讲师,硕士,研究方向为检测技术与自动控制。

于在建模过程中提高模型的解算速度^[5]。由于 PCA 的方法能够很好地解决数据相关性的问题,近年来 PCA 的方法被广泛地应用在过程软测量建模、故障诊断和过程监控中,对于建模过程中提高模型的运算速度十分有力。

图 1 显示出 PCA-BP 模型的基本结构和工作流程,样本数据经过 PCA 处之后,维数被大大降低,输入神经元的个数通过主元分析后由程序自动给出,只要确定中间层的个数,并根据需要自由地给出中间层神经元的个数,或通过经验公式来确定中间层的个数,最后可由软件自动生成网络模型,为下一步的神经网络解算做好准备。网络输出的预测值,可以通过与现场手工采样分析值进行比较后,将误差进行反传,以修正网络连接参数,使网络输出值与对应的样本输出值能够有较小的偏差。

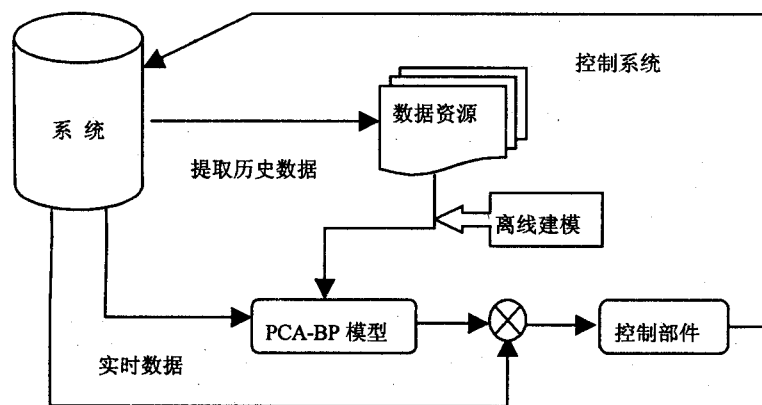


图 1 基于 PCA-BP 的网络模型控制图

2 算法流程的实现

2.1 模块的建立

根据设计需要和开发难度,一般以三层的神经网络的结构作为网络模型基本的设计形式就可以满足现场的应用。在 Visual Basic 中有两类模块,一个是模块,一个是类模块,可以存放一些公共变量、方法、属性;类模块的建立可以封装一部分私有属性和算法,以利于数据的保存和封装且不必受其它程序调用以后发生数据被更改的现象。

通过对现场样本数据的 PCA 预处理以后,样本数据之间相关性得到大大的降低,数据样本的复杂性得到很大的简化,使得神经网络计算的快速收敛性得到有效的保证。

在算法的实现过程中,主要的算法过程和步骤是在 FrmMain 中,即离线训练窗口代码中完成的,离线

算法的完成,需要调用一些相关的模块和类模块所封装的一部分方法和公共元素,基本模块和类模块的功能和作用^[6]。

图 2 为软件功能模块的工作流程。

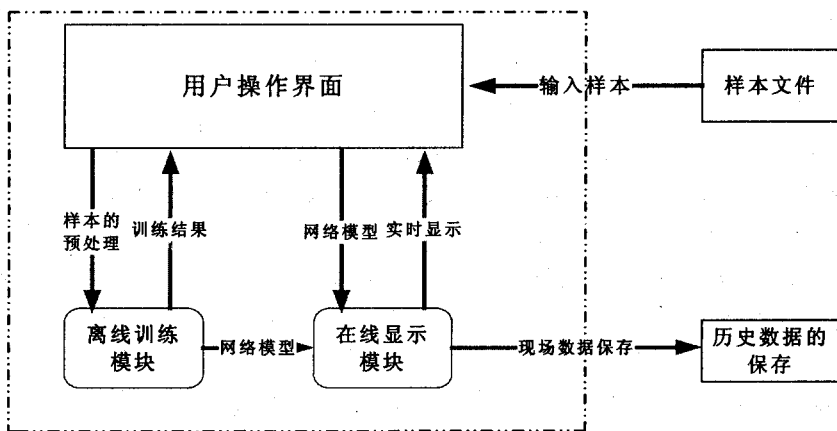


图 2 软件功能模块的工作流程

①模块 *.bas 的建立:

BasNetWork.bas 用于主训练窗口,用于组织和管理训练窗口上一些功能的设定网络训练的一部分必要的参数,并包含网络训练的图形显示等功能。

BasMain.bas 用于存放一些公共变量及数组以供不同的子窗体进行调用和显示,和通讯方面的一些相关参数。

BasPreDeal.bas 用于存放一些进行 PCA 处理相关的中间数据变量和数组,对 PCA 处理过程中的各种方法提供相关的计算支持。

②类模块 *.cls 的建立:

ClsBPNetWork.cls 用于整个网络的建立,用于组织和协调整个网络。包括对网络的初始化,参数的记录和调整、保存计算过程中的一些相关信息。

ClsBPNeuralCell.cls 用于输入层和中间层之间的网络计算,包括这两层之间的权重的初始化、权重的保存和反向调整时连接权重的调整。

ClsBPNeuralCellLay.cls 用于中间层和输出层之间的网络计算,包括这两层之间的权重的初始化、权重的保存和反向调整时连接权重的调整,接收由上一层网络的输出值并作为本层的输入值以利用其进行计算。

ClsNeuralWorkSample.cls 用于从文件中读取已经进行过主元分析以后的降维的数据,为 ClsBPNeuralCell 的 input() 提供数据来源。

2.2 类模块之间的引用关系

从图 3 可以清楚地看到各类模块之间的调用关系,各类模块之间紧密相连进行数据传递和转换。数

组 `NeuralSampleCell()` 是 `ClsBPNeuralWorkSample` 的引用对象,用于样本的输入输出和保存中间计算结果;`ClsBPNeuralCellLay` 派生出 `BPMid_ OutputLay` 和 `BPInput_ MidLay` 两个对象,用于对层与层之间的计算。

而 `NetWorkCell()` 是类 `ClsBPNeuralCell` 的数组引用对象,同时 `NetWorkCell()` 又存在于 `ClsBPNeuralCellLay` 类中,它是把 `ClsBPNeuralCell` 中的方法和属性扩展到 `ClsBPNeuralCellLay` 类中,用于保存各层之间的中间变量。

`FrmMain. frm` 是进行网络计算的主窗体,负责整个运算过程的协调、显示、计算、分析等工作,是实现离线算法的重要窗体,在图中可以看到 `BPNeuralWork` 是 `ClsBpNetWork` 类的对象,它是在 `FrmMain. frm` 代码中,引用 `ClsBpNetWork` 类所有的方法和相应的功能。在 `FrmMain. frm` 中还有一个 `ClsBPNeuralWorkSample` 的对象 `BP_ Test`,它是用于对网络进行测试,以验证网络的泛化推广能力。

2.3 类模块的算法安排

类模块的算法安排如图 4 所示。

1)`ClsBPNeuralCell. cls` 中封装了:

* `Initial()` 对经过主元分析后并且进行标准化以

后的样本矩阵和中间层的神经元之间的连接权重进行初始化,用随机函数给予其赋值以利于网络的计算收敛。

* `CalOut()` 用于计算神经元的输出值,神经元经所采用训练函数可以在训练窗体中进行选择。

2)`ClsBPNeuralCellLay. cls` 中封装了:

将定义 `Private NetWorkcell()` as `ClsBPNeuralcell` 用 `NetWorkcell` 数组来引用和继承 `ClsBPNeuralcell` 中的方法和属性, `InitialLay()` 用于初始化中间层和输出层之间的连接权重,根据所设定的中间层神经元的个数进行确定;用 `Call NetWorkcell(i). Initial()`, `Call NetWorkcell(i). CalOut()` 来实现对中间层和输出层之间权重的初始化和中间计算。

3)`ClsNeuralWorkSample. cls` 中封装了:

记录样本,并且负责记录训练输入和输出数据,及中间过程的数据和计算结果、记录上一次和本次权值的改变大小,并进行相应的权值调整和学习速率的改变等。

`Initial()` 在这个方法中,主要根据主元分析的结果,重新确定各个数组的大小。为了对数据进行很好的封装,里面的私有属性应用 `Let()`、`Get()` 方法将类中的 `Private` 定义的属性变量和数组进行调用和赋值。

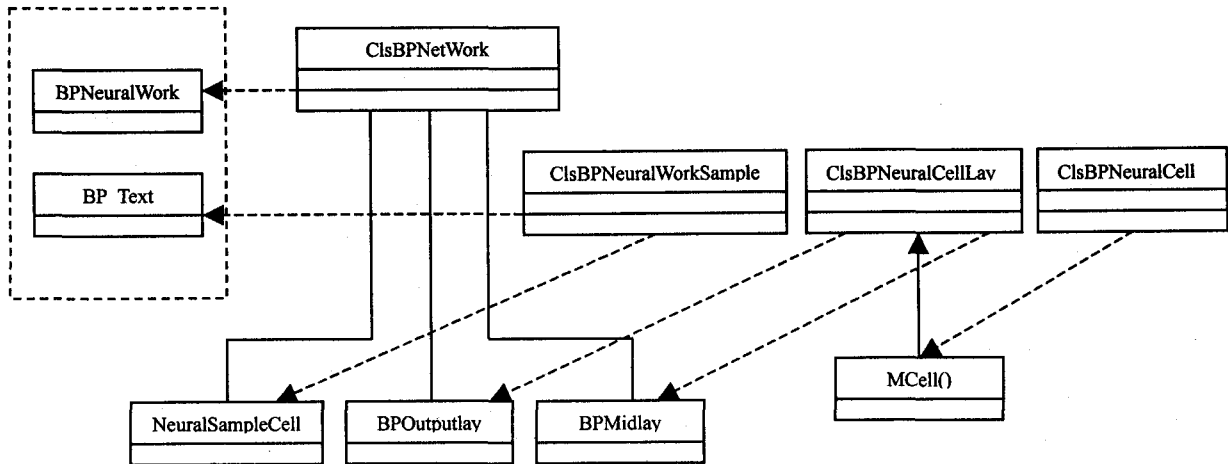


图 3 类之间的调用关系图

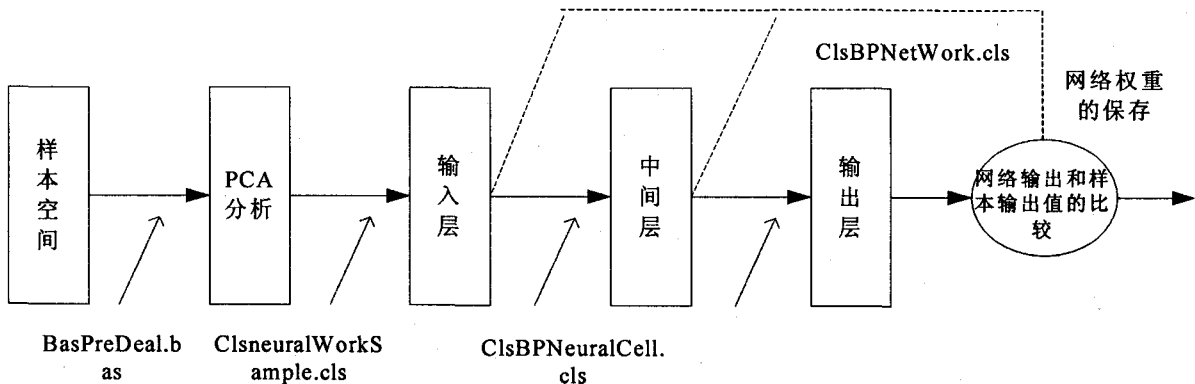


图 4 类模块的算法安排

4)ClsBPNetWork.cls 中封装了:

分别对 ClsBPNeuralCell.cls、ClsBPNeuralCellLay.cls、ClsNeuralWorkSample.cls 进行对象的引用:

```
Public BPInput_MidLay As New ClsBPNeuralCellLay
Public BPMid_Output Lay As New ClsBPNeuralCellLay
Private NeuralSampleCell() As ClsBP NeuralWorkSample
```

在 Initial (ByVal Input_num As Long, ByVal Mid_num As Long, ByVal Output_num As Long, ByVal Sample_num As Long) 中, 分别通过调用这三个对象的 Initial() 方法对各层的连接权重进行初始化。CalNeuralCellOut (ByRef pSample As ClsBPNeuralWorkSample) 用于计算单个神经元的输出, 主要是将神经元的输入值进行求和后, 再根据所选用的激活函数, 进行计算, 其结果作为此层这个神经元的输出值, 并与下一层某个神经元之间的连接权重相乘后的积作为这个神经元的输入值。在 BPTrain() 中是实现训练的主要方法函数, 通过 NeuralSampleCell(n). Inpuu(i) 将保存对应的 PCA 样本。

通过 BPMid_OutputLay.NetWorkcell(i).m_In(j) = pSample(n).Input 将上一层的数据传入下一层, 再利用 Call BPMid_Output Lay.NetWorkcell(i).CalOut() 计算的结果作为下一层的输入值, 实现网络的正向计

算。反向计算如此类推, 计算出 pSample(n). Error-Output(j) 网络输出的误差值, 并通过各层的连接, 反向调整各层的神经元连接权重。

3 误差分析及界面显示

在离线训练中, 用误差曲线来显示网络训练的效果和误差趋势, 不失为一种好的、直观的方法。图 5 显示了网络训练时的变化。

从图 5 上可以看出经过主元分析和 BP 算法改进的样本即 PCA-BP 训练样本以后, 在开始很短的步数内, 误差曲线就下降地很快, 学习速率进行相应自调整和优化, 能够满足设定的误差精度, 有利于现场运用的实际需要^[7]。

4 结束语

建立的 PCA-BP 模型可以克服 BP 算法收敛较慢的特点, 其学习算法简单, 通过调整相应的参数, 改变相应的权值, 使学习进度加快, 便于实时应用。在线运行可以积累足够的样本, 并重新进行离线的训练, 调整模型神经元之间的连接权重, 以此来提高 BP 网络的泛化推广和应用能力。

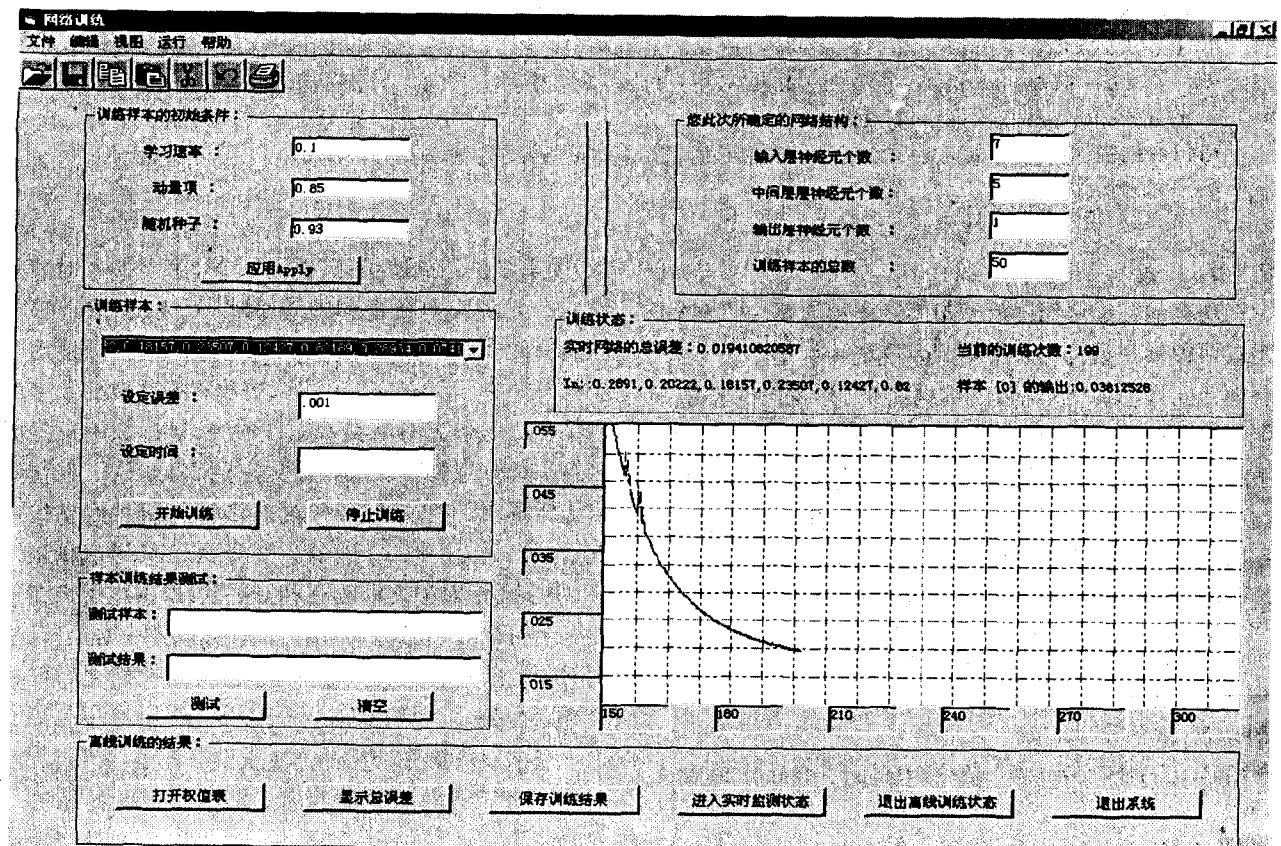


图 5 PCA-BP 的训练误差曲线

对象 G_1 右移 50cm, 用户 U_2 将对象 G_2 左移 50cm, 用户 U_3 将对象 G_3 上移 30cm (如图 1、2、3、4 所示)。用户 U_1 、 U_2 、 U_3 都是针对关联对象组中不同对象的位置属性做修改操作, 因为关联对象组之间有相对固定位置关系, 所以它们之间产生冲突。

假设共享对象组接受三个用户对位置关系的修改。当用户 U_1 先访问位置属性共享锁标记, 此时标记值为零, 故 U_1 为第一个修改用户, 它将其锁集中所有对象属性标记加 1, 并在本地站点产生复制临时版本 V_1 , 最后将版本 V_1 发送给操作管理者。

由于并发关系, 用户 U_2 和 U_3 同时也访问锁集标记锁值, 并将标记锁值累加 1 操作, 分别产生两个临时版本 V_2 和 V_3 (如图 3 和 4 所示), 并将 V_2 和 V_3 分别发送给协同管理者。

对于操作管理者来说, 接受版本 V_1 、 V_2 和 V_3 。每当接收到一个临时版本时, 操作管理者将临时版本计数器加 1, 当临时版本数与属性锁标记值都为 3 时, 表明所有临时版本接收完毕。最后由操作管理者实行仲裁, 决定采用哪个版本为最终版本, 并将最终版本发送到 U_1 、 U_2 和 U_3 的本地界面窗口中。

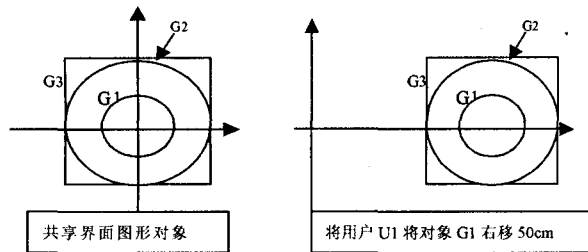
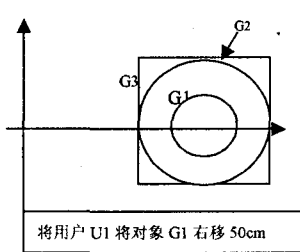
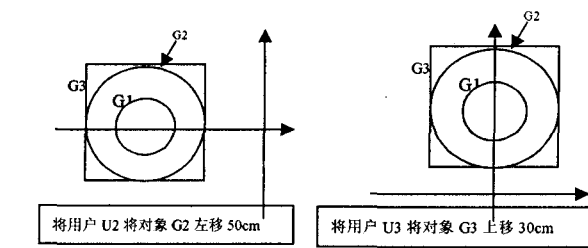
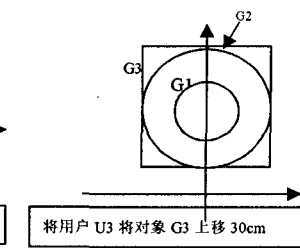


图 1 共享对象

图 2 U_1 用户操作后界面图 3 U_2 用户操作后界面图 4 U_3 用户操作后界面

(上接第 101 页)

参考文献:

- [1] Hecht - Nielsen R. Neurocomputer Applications[J]. Neural Computer, 1988, 2(1): 455 - 453.
- [2] 李士勇. 模糊控制·神经控制·智能控制论[M]. 哈尔滨: 哈尔滨工业大学出版社, 1996.
- [3] Piovoso M J, Kosanovich K A. Application of Multi - Variable Statistical Methods to Process Monitoring and Controller Design[J]. Int J. Control, 1994, 59(3): 734 - 765.

4 结束语

提出了一种解决对象关联操作冲突的多版本动态属性锁并发控制协议, 设计实现了相应的算法。该协议利用操作关联对象集, 检测当多个用户同时修改关联对象组同一个属性产生不同值时发生的操作冲突情况, 解决了一次操作针对在多个对象体现操作效果的操作冲突。分析了协议的正确性, 并通过实例说明该协议能很好地解决操作关联冲突。但该协议所有版本都通过服务器控制版本, 虽然可以使得版本和并发控制得到保证, 但是也造成了服务器与客户器之间频繁通信, 容易导致网络瓶颈。另外, 当版本过多时, 如何智能化地选择版本是多版本技术中有待研究的问题。

参考文献:

- [1] 高曙明, 何发智. 分布式协同设计技术综述[J]. 计算机辅助设计与图形学学报, 2004, 16(2): 149 - 157.
- [2] Sun C Z, Chen D. Consistency Maintenance in Real - time Collaborative Graphics Editing Systems[J]. ACM Transactions on Computer - Human Interaction, 2002, 9(1): 1 - 41.
- [3] Xue Liyin, Orgun M, Zhang Kang. A Multi - versioning Algorithm for Intention Preservation in Distributed Real - time Group Editors[C] // Twenty - Fifth Australasian Computer Science Conference(ACSC2003). Adelaide, Australia: [s. n.], 2003: 19 - 28.
- [4] Citro S, McGovern J, Ryan C. Conflict Management For Real - Time Collaborative Editing in Mobile Replicated Architectures[C] // Proceedings of the Thirtieth Australasian Conference on Computer Science. Ballarat, Australia: [s. n.], 2007.
- [5] Kimwoosacng, Srivastavaajadee P. Enhancing real - time DBMS Performance with multiversion data and Priority based disk scheduling[C] // Proceedings of the 12th Real - time Systems Symposium. Neworlean: [s. n.], 1991: 222 - 231.
- [6] Sun C Z, Jia X, Zhang Y, et al. Achieving convergence, causality - preservation, and intention - preservation in real time cooperative editing systems[J]. ACM Transaction on Computer - Human Interactions, 1998, 5(1): 59 - 68.

- [4] 尚长军, 陈 曦, 钱积新. 基于 PCA - BP 神经网络的精馏塔产品组成软测量模型[J]. 工业仪表与自动化装置, 2003 (4): 33 - 36.
- [5] 刘克显. 软测量及智能控制在复杂工业过程参数检测与控制中的应用研究[D]. 沈阳: 东北大学, 2001.
- [6] 贾 群, 李和平. 软测量与工控组态软件在 DMF 微量水工艺中的应用研究[J]. 中国仪器仪表, 2007(10): 47 - 50.
- [7] 贾 群. OPC 技术在过程工业中的应用研究[J]. 中国仪器仪表, 2006(5): 72 - 74.