

基于C#.NET与Matlab接口和BP网络的汽车产量预测

黄一丹, 严洪森, 冯丽娟, 吴 奇

(东南大学复杂工程系统测量与控制教育部重点实验室, 江苏 南京 210096)

摘要: 为了提高软件预测系统的开发效率和精度, 研究了误差反向传播神经网络(Error Back Propagation Neural Network, BPNN)在Matlab 2006a环境下的实现方法, 采用迭代法选择神经网络训练参数。探讨了Matlab与C#.NET开发平台的接口技术, 能成功实现C#.NET对Matlab神经网络工具箱的调用。以汽车产量预测为应用案例, 开发了基于C#.NET和Matlab接口的神经网络预测系统。实验结果表明, 该预测系统对汽车产量预测有着良好性能, 接口技术的应用提高了软件开发效率。

关键词: 人工神经网络; 接口技术; 预测

中图分类号: TP311

文献标识码: A

文章编号: 1673-629X(2008)11-0036-05

Vehicle Production Forecasting Based on BPNN and Interface Between C#.NET and Matlab

HUANG Yi-dan, YAN Hong-sen, FENG Li-juan, WU Qi

(Ministry of Education Key Laboratory of Measurement and Control of Complex Systems of Engineering, Southeast University, Nanjing 210096, China)

Abstract: In order to improve the efficiency and accuracy of software forecasting system, researched into the implementing method of BPNN (Error Back Propagation Neural Network) in Matlab 2006a. The training parameters of BPNN were chosen by iterative method. The interface technology between C#.NET and Matlab was discussed and then Matlab NN toolbox could be called by C#.NET successfully. A BPNN forecasting system based on the interface between C#.NET and Matlab was developed for vehicle production forecasting. Experiment results indicate that this system can forecast the vehicle production accurately and the application of interface technology can improve the software development efficiently.

Key words: artificial neural network; interface technology; forecast

0 引言

汽车产量受生产条件、天气状况、市场需求等诸多因素的影响, 使得汽车产量预测具有较大的复杂性和非线性等特点。常用的汽车产量预测方法包括时间序列方法、相关分析方法、灰度预测方法等, 但这些方法大都要建立复杂的数学模型, 而且所建立的模型不能全面地反映所预测数据的内在结构。人工神经网络(Artificial Neural Network)作为一种并行的计算模型,

一般不必事先知道被建模对象的结构、参数、动态特性等知识, 只需给出对象的输入、输出数据, 通过网络本身的学习功能就可以达到输入与输出的完全符合, 并且有很好的非线性映射能力。

Matlab是美国MathWorks公司推出的一套高性能的数值计算和可视化软件。Matlab平台上有专门针对神经网络计算的神经网络工具箱, 内含大量可用于设计神经网络的工具函数。.NET是微软公司推出的一种面向网络、支持各种终端的开发平台环境。利用.NET开发的系统具有界面友好、执行速度快、易维护和升级等优点, 能够生成可执行文件, 并能有效地保护算法和数据, 也可脱离编程环境运行^[1]。但是在工程计算方面, .NET跟Matlab相比编程则显得复杂得多。文献[2]实现了在VB.NET下调用神经网络工具箱, 大大简化了编程, 同时提高了软件的效能。但VB语

收稿日期: 2008-03-14

基金项目: 国家863计划资助项目(2007AA04Z112); 国家自然科学基金资助项目(60574062)

作者简介: 黄一丹(1983-), 女, 湖南津市人, 硕士研究生, 主要研究领域为需求预测、计算机集成制造; 严洪森, 教授, 博士生导师, 主要研究领域为生产计划与调度、知识化制造、并行工程等。

言复杂的语法规则给软件开发带来了难度。文献[3]实现了在 ASP.NET 中调用神经网络工具箱,采用 Active X 技术和 B/S 结构,提高了执行效率。但其开发底层 Matlab 的 M 文件入口参数杂乱冗余,而且静态设置神经网络训练参数,这样会影响算法的合理性和灵活性。另外,以上两篇文献中均是将神经网络运用于故障诊断,而非时序预测。为此,文中采用针对 .NET 平台专门设计的、语法简单、完全面向对象的 C# 语言,在 .NET 中实现对 Matlab 神经网络工具箱的调用。设计 BP 网络预测汽车产量,并采用迭代法动态选择神经网络训练参数,将两种软件的优点充分结合起来,必将提高软件开发效率,使其具有更高的性能和更大的应用范围。

1 BP 网络及 Matlab 神经网络工具箱

人工神经网络是由大量的处理单元(神经元)互相连接而成的网络。人工神经网络的信息处理由神经元之间的相互作用来实现,知识与信息的存贮表现为网络元件互连分布式的物理联系,人工神经网络的学习和识别决定于各神经元连接权系数的动态演化过程^[4]。

多层神经网络模型的拓扑结构如图 1 所示。它由输入层、中间层(隐层)和输出层组成。在神经网络中引入隐层神经元,神经网络就具有更好的分类和记忆等能力。1985 年 Rumelhart 等提出了误差反向传播算法(Error Back Propagation,简称 BP 算法),系统地解决了多层神经元网络中隐层单元连接权的学习问题,并在数学上给出了完整的推导。采用 BP 算法的多层神经网络模型一般称为 BP 网络。

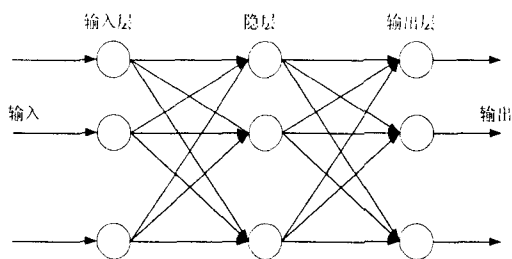


图 1 多层神经网络模型的拓扑结构图

BP 网络的学习过程由两部分组成:正向传播和反向传播。当正向传播时,输入信息从输入层经隐层处理后传向输出层。如果在输出层得不到希望的输出,则转入反向传播,将误差信号沿原来的神经元连接通路返回。返回过程中,逐一修改各层神经元连接的权值。这种过程不断迭代,最后使得信号误差达到允许的范围之内。

神经元传递函数(又称能量函数、网络节点函

数^[5])是用来模拟神经细胞的兴奋、抑制、疲劳以及阈值等非线性特性。常用的神经元传递函数有:S 型函数、高斯函数、线性函数、阈值函数等。具体的表达式如公式(1)到(4)所示。

$$\text{S 型函数: } f(x) = 1/(1 + e^{-x}) \quad (1)$$

$$\text{高斯函数: } f(x) = e^{-(x^2/\sigma^2)} \quad (2)$$

$$\text{线性函数: } f(x) = kx \quad (3)$$

$$\text{阈值函数: } f(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (4)$$

文中采用 S 型函数作为神经元传递函数,因为它具有可靠的生物学基础,其突变性和饱和性很好地反映了神经细胞兴奋过程所产生的神经冲动性和疲劳特性。

关于 BP 网络的映射能力,根据 Kolmogorov 连续性定理,具有一个隐含层的前向神经网络,能以任意精度逼近任意函数。实际上,为了避免三层 BP 神经网络使用过多的隐层节点,通常会增加网络的隐层层数。如何选取隐层层数和节点数,至今还没有确切的方法和理论,通常凭借对学习样本和测试样本的误差交叉评价的试错法选取^[6]。BP 网络强大的映射能力,使其成为预测非线性时间序列的主要方法。

Matlab 神经网络工具箱以神经网络理论为基础,用 Matlab 语言构造出典型神经网络的工具函数。Matlab 中专门编制了大量有关 BP 神经网络的工具箱函数,为 BP 神经网络的应用研究提供了强有力的工具。文中用到的主要工具箱函数^[7]有:

(1) 神经元传递函数:tansig()为双曲正切 S 型(Sigmoid)传递函数,用于将神经元上范围为 $(-M, +M)$ 的输入值映射到区间 $(-1, +1)$ 上,利用 BP 算法训练隐层神经元时经常采用它;purelin()为线性传递函数,利用 BP 算法训练输出层神经元时经常采用它。

(2) 网络建立函数:newff()用于建立一个前馈 BP 神经网络。

(3) 网络训练函数:train()用于训练建立的 BP 神经网络。

(4) 网络仿真函数:sim()用于仿真训练前后的 BP 神经网络。

2 C#.NET 与 Matlab 接口

.NET 与 Matlab 的接口方法主要有以下四种。

2.1 利用 Matlab 自身的编译器

Matlab 的编译器可将 Matlab 的 M 文件转换为 C 或 C++ 的源代码以产生完全脱离 Matlab 运行环境的独立的应用程序。当 Matlab 的编译器 mcc 加入适当的参数 -e(mcc e *.*)或 -p(mcc p *.*)就可

生成将输入的 M 文件转换为适用于特定运用的 C 或 C++ 源代码。

2.2 利用 COM 或 .NET 组件技术

组件对象模型 COM(Component Object Model)是一种软件架构^[8],它允许应用软件由不同的软件制造商的组件产品构建。

Matlab R2006a 新版本中,推出了 MATLAB Builder for .NET,它是 MATLAB Compiler 的扩展功能,能够从 Matlab 算法代码自动生成独立的 .NET assembly 类库或 COM 对象。生成的 .NET assembly 类库为 C#, VB.NET 或任何 CLS(Common Language Specification)兼容的编程语言调用。生成的 COM 对象则可由任何兼容 COM 技术的程序环境调用。MATLAB Builder for .NET 的应用可将用 Matlab 开发的算法集成到桌面和 Web 应用程序中。通过标准调用接口,在用户群间共享开发的算法。

2.3 利用 Mideva 平台

Mideva 是 Mathtools 公司推出的一种 Matlab 编译开发软件平台,是一个强大而完备的 M 文件解释和开发环境,它通过应用 Matcom 和实时编译技术而达到快捷的速度。该软件平台有为 Borland C++、Visual Basic 和 Delphi 等编程语言开发的不同版本。Mideva 提供了上千个基本功能函数,包括基本操作、命令等。Mideva 具有编译转换功能,能够将 Matlab 函数或编写的 Matlab 程序转换为 C++ 形式的 DLL,然后在 .NET 中调用动态库函数,这样就可能实现对 Matlab 各种工具箱的调用^[9]。

2.4 利用 Matlab 引擎技术

Matlab 引擎函数库是 Matlab 提供的一系列程序的集合,它允许用户在自己的应用程序中对 Matlab 函数进行调用,将 Matlab 做为一个计算引擎使用,让其在后台运行。当用户使用 Matlab 引擎时,相当于在后台启动了一个 Matlab 进程。Matlab 引擎函数在用户程序与 Matlab 进程之间起一个桥梁的作用,它完成两者的数据交换和命令的传递。在 Windows 环境中,Matlab 引擎是通过 Active X 来完成的。Active X 是由 Microsoft 制定的一种独立于编程语言的组件集成协议,它不受开发环境的限制。Active X 空间能够在不同的环境中使用,包括 VC、Delphi、C++ builder 等^[8]。

Matlab 提供了两种 Active X 技术支持:Active X 组件集成和 Active X 自动化服务。当用自己开发的 .Net 程序集来控制和使用 Matlab 的 Active X 组件时,实际上充当了自动化服务器的角色。应用程序运行时一般是先向 Matlab 计算引擎组件传送数据和命令,再由 Matlab 计算引擎与 Matlab 服务器交互,最后由

Matlab 计算引擎将运算结果返回到应用程序^[3]。

实验证明,以上的前三种方法都不支持 .NET 调用 Matlab 神经网络工具箱,只有 Matlab 引擎技术是可行的。C#.NET 下利用 Matlab 引擎技术实现神经网络的过程如图 2 所示。

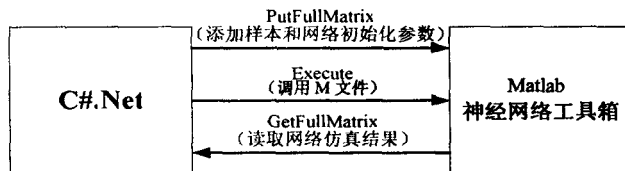


图 2 C#.NET 调用 Matlab 神经网络工具箱过程图

3 实现过程

3.1 编制 M 文件

神经网络在 Matlab 中实现的主要步骤如图 3 所示。

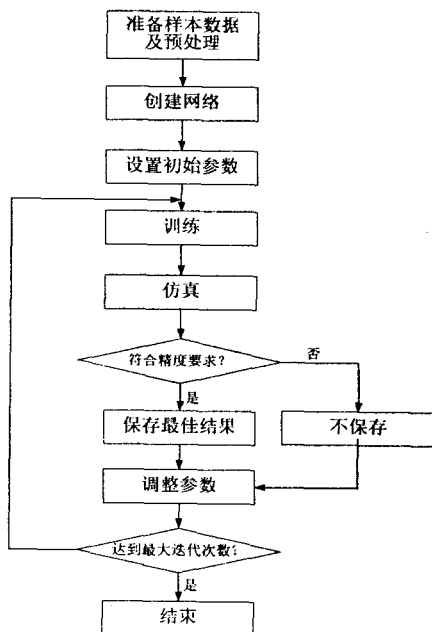


图 3 神经网络在 Matlab 中实现的主要步骤图

在 BpNeuralNet.m 文件中编制了神经网络的实现逻辑。其函数定义如下:

```
Function [Result] = BpNeuralNet(TrainIn, TrainOut, TestIn, TestOut, Par)
```

BpNeuralNet 为函数名称,等号右边的小括号内是输入参数,TrainIn 为训练输入矩阵,TrainOut 为训练输出矩阵,TestIn 为测试输入矩阵,TestOut 为测试输出矩阵,Par 为网络设置参数矩阵。等号左边的中括号内是输出参数,Result 为预测结果矩阵。

3.1.1 编制样本数据及预处理

将某汽车厂 2001 年 1 月至 2007 年 9 月的汽车月产量作为有效样本数据。这 81 个样本数据的前 60 个

作为训练集,后 21 个作为测试集。训练集数据用于更新网络权重和偏置量,测试集用作对未知样本进行测试并检验网络的计算精度。考虑到汽车月产量的特点,并要充分地利用历史信息,样本对可按照如下思路构成:要预测某年某月的汽车产量,可以将该月之前的 12 个月的汽车产量作为输入影响变量,这样就得到一组样本对的映射。基于这种思路,将样本分成若干段有一定重叠度的数据段,每一段的前 12 个数据作为训练输入,后一个数据作为其对应的目标输出。这样,可以得到 12×48 的训练输入矩阵, 1×48 的训练目标矩阵, 12×21 的测试输入矩阵, 1×21 的测试目标矩阵。

一般情况下,先将原始样本数据预处理,利用工具箱中的 mapstd 函数将其变换为均值为 0,方差为 1 的时序。

该部分主要代码如下:

```
[TrainIn1,ps1]=mapstd(TrainIn2);%样本预处理
```

3.1.2 创建网络及调整参数

创建 BP 网络,其中隐层和输出层神经元传递函数分别设置为 tansig() 和 purelin(),隐含层和输出层神经元个数、学习速率等参数则由接口输入。

该部分主要代码如下:

```
net=newff(minmax(TrainIn1),[Par(2),Par(3)],{'tansig','purelin'},'traingd');%创建网络
```

```
net.trainParam.mu=Par(1);%学习系数
```

接口输入的神经网络训练参数由研究者粗略估计得出,显然无法满足预测精度的需要,有待进一步调整。在众多参数的调整中,以隐层神经元个数和学习系数对预测结果影响最为明显。文中采取迭代法对这两个参数进行调整,其主要思想是固定其中一个参数,对另一个参数由大到小调整,每次调整后训练和仿真,保留最佳配置。采取迭代法调整参数的主要步骤如下:

STEP1:判断学习系数是否大于 0.01,若是,学习系数减 0.01 并继续,否则,停止迭代;

STEP2:判断隐层神经元个数是否大于 1,若是,隐层神经元个数减 1 并继续,否则,转向 STEP1;

STEP3:训练并仿真;

STEP4:判断平均预测误差率 MAPE 是否小于 5%,若是,则保存最佳结果,否则,放弃保存。转向 STEP2。

3.1.3 训练并仿真

利用事先准备好的样本集对网络进行训练和仿真,仿真的结果需要进行反归一化,以得到跟原始样本相对应的结果。

```
[net,tr]=train(net,TrainIn1,TrainOut1);%训练网络
```

```
Result1=sim(net,TestIn1);%仿真
```

```
Result=mapstd('reverse',Result1,ps4);%反归一化
```

3.2 在 .NET 环境中实现 Matlab 引擎

.NET 是微软在总结吸收了多种编程开发语言的基础上最新发布的发展平台,在 .NET 环境下可以很方便地实现 Matlab 引擎,从而提供更灵活更高效的解决方案。

当 Matlab 系统在计算机上正确安装后,Matlab 计算引擎 Active X 组件将自动在系统注册。在 Visual Studio.NET 开发环境下使用 Matlab 计算引擎,首先添加 COM 引用“Matlab Application (Version 7.0) Type Library”。完成了 Matlab 计算引擎组件的引用后就可以在程序中实例化引擎对象,然后使用其提供的接口进行编程。引擎对象提供的最常用的接口有三个:Execute,PutFullMatrix 和 GetFullMatrix。它们分别是实现运行脚本,向 Matlab Server 中添加矩阵和从 Matlab Server 中读取矩阵功能。神经网络设置的部分参数由 .NET 开发的界面层输入,如图 4 所示。以 C# 语言为例,接口文件的主要代码如下:

```
MLAppClass matlab=new MLAppClass();
```

```
matlab.PutFullMatrix("TrainIn","base",TrainInput,TrainInputIm);//向 Matlab Server 中添加矩阵
```

```
matlab.Execute(@"[Result]=BpNeuralNet(TrainIn,TrainOut,TestIn,TestOut,Par)");
```

```
matlab.GetFullMatrix("Result","base",ref ReArr,ref ReArrIm);//从 Matlab Server 中读取矩阵。ReArr 和 /ReArrIm 为预先定义好的 System.Array 对象
```

```
ReArr.CopyTo(yc,0);//将预测结果保存在 double 型数组 yc 中
```

```
matlab.MinimizeCommandWindow();//最小化 matlab command 窗口
```

```
matlab.Quit();//关闭 matlab 服务器
```

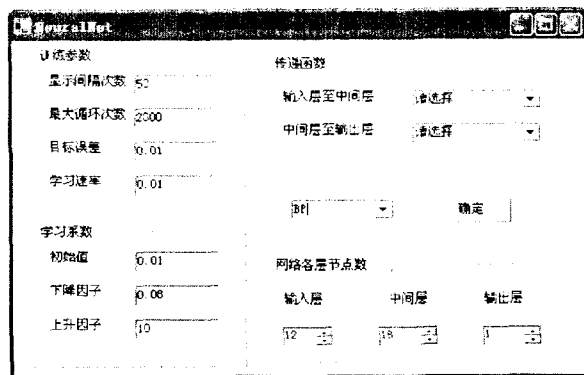


图 4 接口参数输入界面

3.3 预测结果及分析

经过多次试调参数,得到了最佳预测结果如表 1 所示。

表 1 使用 BP 网络进行的汽车产量预测结果

预测日期	实际值(辆)	预测值(辆)	相对误差(%)
2006.1	133	132.23	-0.6
2006.2	180	171.12	-4.9
2006.3	170	180.14	6.0
2006.4	200	216.82	8.4
2006.5	290	284.53	-1.9
2006.6	220	220.64	0.3
2006.7	210	212.85	1.4
2006.8	165	163.89	-0.7
2006.9	118	127.39	8.0
2006.10	98	102.21	4.3
2006.11	85	85.07	0.1
2006.12	75	82.88	10.5
2007.1	140	129.60	-7.4
2007.2	185	178.11	-3.7
2007.3	189	181.60	-3.9
2007.4	220	216.04	-1.8
2007.5	302	296.83	-1.7
2007.6	250	230.37	-7.9
2007.7	220	219.25	-0.3
2007.8	160	166.85	4.3
2007.9	110	124.88	4.1
平均预测误差率: MAPE=3.9%			
预测准确率: ACCU=98.3%			

一般来说,MAPE 在 5% 以下,ACCU 保持在 95% 以上,就可满足实际需要^[10]。由表 1 可以看出,预测值与实际值的变化趋势基本吻合。预测误差呈随机波动状态,说明预测误差无显著趋势,达到了良好的预测效果。

4 结束语

研究了基于 C#.NET 的 Matlab 引擎技术,成功调用 Matlab 神经网络工具箱,设计并实现 BP 网络对某汽车厂汽车产量的预测,使研究者从大量烦琐的数

学运算中解脱出来,极大地提高了编程效率。不仅如此,还可以优化 BP 网络,以达到更高的预测精度。此外,Matlab 环境之外的引擎技术和接口调用的效率问题也值得进一步思考。

参考文献:

- [1] 赵士伟,赵明波,陈平.基于 COM 的 MATLAB 与 C#.NET 混合编程的实现与应用[J].山东理工大学学报:自然科学版,2006,20(4):28-29.
- [2] 王耀龙,梁小冰,黄萍.Visual Basic.NET 调用 Matlab 神经网络工具箱的实现方法[J].现代计算机,2005(1):78-80.
- [3] 陈涛,杨建国,杨江云.基于 MATLAB 与 .Net 的神经网络推理应用[J].微机发展(现名:计算机技术与发展),2004,14(11):33-36.
- [4] 杨建刚.人工神经网络实用教程[M].杭州:浙江大学出版社,2001:11-44.
- [5] 段登伟.电力系统短期负荷预测的研究[D].成都:四川大学,1999.
- [6] 蒋良孝,李超群.基于 BP 神经网络的函数逼近方法及 MATLAB 实现[J].微型机与应用,2004(1):52-53.
- [7] Matlab Neural Network Toolbox user's Guide[M].US: Mathworks,2006.
- [8] 董维国.深入浅出 MATLAB 7.x 混合编程[M].北京:机械工业出版社,2005:260-262.
- [9] 王俊忠,梁海刚,石旭斌.MATLAB 外部程序接口实现方法研究[EB/OL].2006-07-12.[2007-11-1].http://www.xinxijishu.org/Article/pc/kaifayingyong/200607/1576.html.
- [10] 王剑俊.基于 RBF 网络的城市供水短期负荷预测[J].西南给排水,2007,29(3):24-27.

(上接第 35 页)

- 1879-1882.
- [2] 麻志毅,陈泓婕.一种面向服务的体系结构参考模型[J].计算机学报,2006,29(7):1011-1019.
- [3] Benatallah B, Dumas M. Declarative composition and peer-to-peer provisioning of dynamic Web Services[C]//The 18th International Conference on Data Engineering (ICDE'02). San Jose:[s.n.],2002.
- [4] Zhang Liang-jie, Li Bing, Chao Tian, et al. Requirements driven dynamic services composition for Web services and Grid solutions[J]. Journal of Grid Computing,2004,2(2):121-140.
- [5] Ponnekanti S R, Fox A. SWORD: A developer toolkit for Web service composition[C]//In: proceedings of International World Wide Web Conference. Honolulu, Hawaii, USA:[s.n.],2002:83-107.
- [6] Thkkar S. Dynamically composing Web services from on-line source[C]//In: Proceedings of AAAI Workshop on Intelligent Service Integration. Edmonton, Alberta, Canada:[s.n.],2002:1-7.
- [7] Benatallah B, Dumas M. The self-serv environment for Web services composition[J]. IEEE Internet Computing, 2003,7(1):40-48.
- [8] Tosic V, Mennie D, Pagurek B. On dynamic service composition and its applicability to e-business software systems[C]//In: Proceedings of Workshop on Object-Oriented Business Solutions. Budapest, Hungary:[s.n.],2001.
- [9] Li Jing-Shan, Liao Hua-Ming, Hou Zi-Feng, et al. A dynamic service composition method based on semantic interface description in pervasive computing[J]. Journal of Computer Research and Development, 2004, 41(7):1124-1134.