

基于 RBF 神经网络的时间序列预测

张玉瑞 陈剑波

(上海交通大学信息安全学院,上海 200030)

摘要 前馈神经网络在时间序列预测中的应用已得到充分地认可,一些模型已经提出,例如多层感知器(MLP),误差反向传播(BP)和径向基函数(RBF)网络等等。相对于其他前馈神经网络,RBF网络学习速度快,函数逼近能力强,因而在时间序列预测方面具有很好的应用前景。

关键词 RBF网络 BP网络 时间序列预测

文章编号 1002-8331-(2005)11-0074-03 文献标识码 A 中图分类号 TP18

Time Series Prediction Based on RBF Neural Network

Zhang Yurui Chen Jianbo

(College of Information Security Engineering, Shanghai Jiaotong University, Shanghai 200030)

Abstract: The use of feed forward neural network for time series prediction is fully recognized and a number of models have been developed for example Multilayer Perceptron(MLP), Back Propagation(BP) and Radial Basis Function(RBF) network etc. The RBF network has shown a great promise in this sort of problems because of its faster learning capacity and better capability of approximation to underlying functions compared with other feed forward networks.

Keywords: RBF network, BP network, time series prediction

1 引言

传统的时间序列分析是从纯数学的角度来分析的,但是由于实际应用中时间序列具有不规则、混沌等非线性特征,很难对系统建立理想的模型,预测更是无能为力。而人工神经网络由于其具有自组织、自学习能力、非线性逼近能力,将它用于时间序列预测更为有效,因为它可以对网络进行反复的训练,从而使网络学习存储的知识用于推理时更接近实际系统的值。随着学习样本及新知识的补充,网络的适应能力将不断增强。而当时间序列规律发生改变,只需将新的样本加入网络中进行重新学习即可得到新的知识,再用训练好的网络进行规律改变后的时间序列预测。当前用得最广泛的是BP网络。

BP网络虽然可以实现输入和输出间的任意非线性映射,但是由于BP神经网络是基于梯度下降的误差反向传播算法进行学习的,所以网络的学习速度通常很慢,而且很容易陷入局部极小点,尽管采用一些改进的快速学习算法可以较好地解决某些实际问题,但是在设计过程中往往都要经过反复的试凑和训练过程,无法保证每次训练时BP算法的收敛性和全局最优性。此外,BP网络隐层神经元的作用机理及其个数选择也是一个难点。而径向基函数(RBF)网络是以函数逼近理论为基础而构造的一类前向网络,这类网络的学习等价于在多维空间中寻找训练数据的最佳拟合平面。径向基函数网络的每个隐层神经元的函数都构成了拟合平面的一个基函数,网络也因此得名。径向基函数网络是一种局部逼近网络,即对于输入空间的某一个局部区域只存在少数的神经元用于决定网络的输出。BP网络则是典型的全局逼近网络,即对每一个输入/输出数据对,网络的所有参数都需要调整。由于二者的构造本质不同,径向基函数网络相比在函数逼近能力、分类能力和学习速度等方面均优于BP网络^[1],因而将它应用于复杂的时间序列预测会取得很好的效果。

2 RBF网络

RBF神经网络由三层组成,其结构如图1所示。输入层节点只传递输入信号到隐层,隐层节点由像高斯函数那样的辐射状作用函数构成,而输出节点是简单的线性函数。

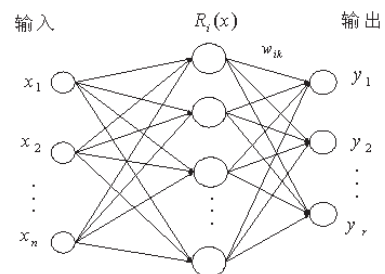


图1 RBF神经网络

隐层节点中的作用函数(基函数)对输入信号将在局部产生响应。最常用的基函数是高斯函数:

$$R_i(x) = \exp\left[-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right] \quad i=1, 2, \dots, m \quad (1)$$

其中 x 是 n 维输入向量, c_i 是第 i 个基函数的中心,与 x 具有相同维数的向量, σ_i 是第 i 个感知的变量,它决定了该基函数中心点的宽度, m 是感知单元的个数; $\|x - c_i\|$ 表示 x 和 c_i 之间的距离。

输入层实现从 $x \rightarrow R_i(x)$ 的非线性映射,输出层实现从 $R_i(x)$ 到 y_k 的线性映射,即:

$$y_k = \sum_{i=1}^m w_{ik} R_i(x) \quad k=1, 2, \dots, r \quad (2)$$

其中 r 是输出节点数, w_{ik} 是权值。

3 RBF 网络的学习算法

RBF 网络的隐层单元数目、基函数的中心和权值都需要通过学习决定 ,OLS 算法^[2]是众多 RBF 网络学习算法中较为有效的算法。OLS (Orthogonal Least Squares) 算法 ,即正交最小平方算法 ,它是 S.chen 等人提出来的 ,这种算法一次选择一个样本 ,方式是使解释期望输出方差的递增值最大 ,从而找出最佳隐层单元数目 ,同时算出输出权值。由于通常的预测模型 ,用的都是多输入单输出的结构 ,所以设 :

$y=y(1) \dots y(N)^T$ 为期望输出序列 ;
 $P=[p_1 \ p_2 \ \dots \ p_n]$ 为隐层输出阵 ,其中 $p_i=[p_i(1) \ p_i(2) \ \dots \ p_i(N)]^T, 1 \leq i \leq n$;

$w=[w_1 \ w_2 \ \dots \ w_n]^T$ 为输出权阵 ;
 $E=[e(1) \ \dots \ e(N)]^T$ 为学习后误差序列。

将 P 奇异分解 $P=C \cdot A$ A 是 $n \times n$ 阶奇异阵 ,有 :

$$A=\begin{pmatrix} 1 & \alpha_{12} & \alpha_{13} & \dots & \alpha_{1n} \\ 0 & 1 & \alpha_{23} & \dots & \alpha_{2n} \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & \\ 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

而 C 是 $N \times n$ 阶矩阵 ,且列向量 c_i 是正交的 :

$\therefore C^T C=H$ (3)

这里 H 是对角阵 ,其对角元素 :

$h_i=c_i^T c_i = \sum_{j=1}^N c_i(t) c_i(t), 1 \leq i \leq n$ (4)

基于 OLS 解 $\hat{g}=H^{-1} C^T y$ (5)

或者 $\hat{g}=\frac{c_i^T y}{c_i^T c_i} \quad 1 \leq i \leq n$ (6)

并且 $A \hat{w}=\hat{g}$ (7)

确定隐层单元的中心点的具体递推如下 :

(1) 第一步 ,对于 $1 \leq i \leq n$,计算 :

$c_1^i=p_i$
 $g_1^i=(c_1^i)^T y / (c_1^i)^T c_1^i$
 $[err]_1^i=(c_1^i)^T c_1^i / y^T y$
寻找 $[err]_1^{i_1}=\max\{[err]_1^i, 1 \leq i \leq n\}$

选择 $c_1=c_1^{i_1}=p_{i_1}$

(2) 第 k 步 $k \geq 2$,对于 $1 \leq i \leq n \ i \neq i_1 \ \dots \ i \neq i_{k-1}$,计算 :

$\alpha_{jk}^i=c_j^T p_i / (c_j^T c_j), 1 \leq j \leq k$
 $c_k^i=p_i - \sum_{j=1}^{k-1} \alpha_{jk}^i c_j$
 $g_k^i=(c_k^i)^T y / (c_k^i)^T c_k^i$
 $[err]_k^i=(c_k^i)^T c_k^i / y^T y$
寻找 $[err]_k^{i_k}=\max\{[err]_k^i, 1 \leq i \leq n \ i \neq i_1 \ \dots \ i \neq i_{k-1}\}$

选择 $c_k=c_k^{i_k}=p_{i_k} - \sum_{j=1}^{k-1} \alpha_{jk}^{i_k} c_j$

直到第 n_s 步 ,有 $1 - \sum_{j=1}^{n_s} [err]_j < \rho$ 则结束 ,其中 $0 < \rho < 1$ 是选定的允许误差 ,这样就可确定中心点 ,然后由式(7)算出输出权阵 \hat{w} 。

4 时间序列的 RBF 神经网络预测模型

为了预测时间序列中的 $y(i)$,令输入矢量 $x(t)=[y(t-1), y(t-2) \dots y(t-n)]^T$,它是一个 n 维矢量 ,代表过去的 n 个样本 ,其中 n 是历史数据长度。因此整个网络结构等效为 $y(t) \approx y(t-1) y(t-2) \dots y(t-n)$ 。

由于不同的时间序列前后数据关联程度很不相同 ,尤其是复杂的时间序列 ,不同的历史数据长度预测结果大相径庭 ,且在不同阶段采用的历史数据长度也不宜相同。该文不是按经验和试凑的方法 ,而是采用截取预测数据前的一段数据 ,分别采用不同的历史数据长度用 RBF 网络进行一步预测 ,选取最小平方误差的那个长度 ,以此优化历史数据长度 ,提高网络的预测性能。

5 应用例子

将基于 OLS 算法的 RBF 神经网络应用到时间序列预测 ,如图 2 所示 ,为某公司国际航班 1949-1959 年每月的乘客数。

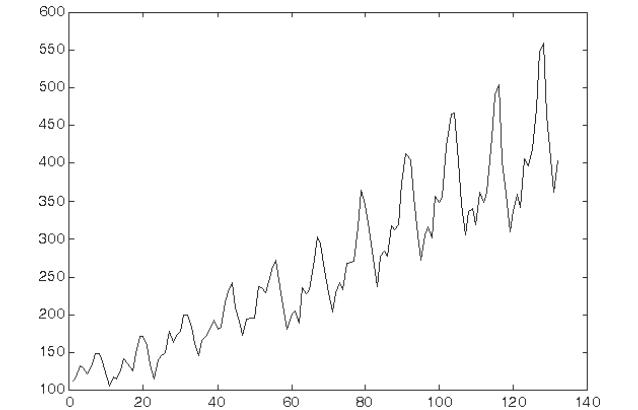


图 2 1949-1959 年每月乘客数

根据这些数据来预测 1960 年的乘客数 ,选择 1953-1959 年进行一步预测 ,得到 $n=16$ 为最佳历史数据长度。对于复杂的时间序列 ,为避免过拟合 ,允许误差不宜过小 ,训练样本不宜过多 ,这里选择允许误差为 0.001 ,训练样本为 1954-1959 年的 72 个样本 ,对 1960 年进行预测。实线为实际值 ,虚线为网络输出值 ,网络预测 1960 年结果如图 3 所示。

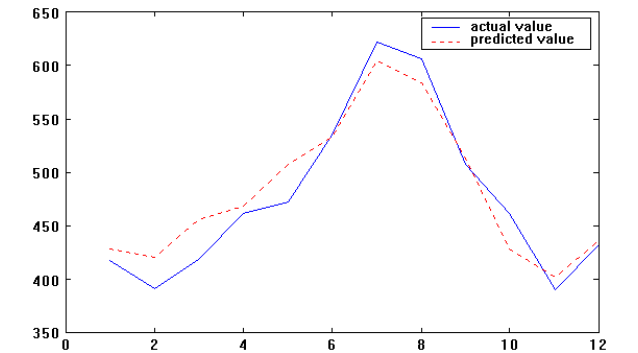


图 3 RBF 网络预测和实际值对比

用 BP 网络进行预测 ,选择双层隐层 ,允许误差为 0.001 ,第一层和第二层隐层均采用两个神经元 ,因为增加隐层中的神经元个数并不能提高预测精度^[3]。采用的历史数据长度为 16。实线为实际值 ,虚线为网络输出值。因为 BP 网络的初始化是随机的 ,因而每次运行结果都不相同 ,性能很不稳定 ,而 RBF 网络的每次输出结果都是相同的 ,这里选择 BP 网络最好的结果 ,如图 4 所示。

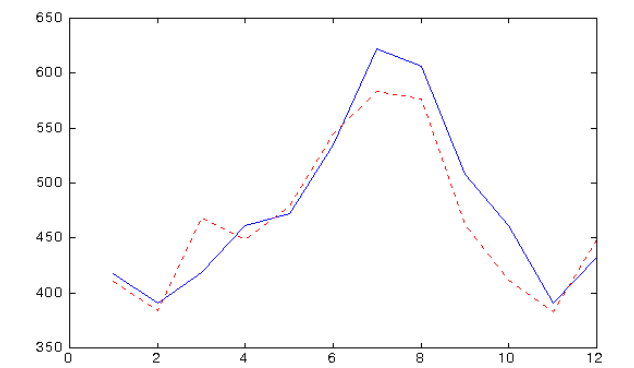


图 4 BP 网络预测和实际值对比

另外 ,尽管采用了自适应学习速率动量梯度下降反向传播算法对网络进行训练 ,BP 网络学习速度依然比较慢 ,以该实验为例 ,RBF 网络只用了 2 秒钟 ,BP 网络则用了 3 分钟左右。BP 网络的均方误差为 841.5604 ,而 RBF 网络的为 479.4058。

6 结论

(1)采用基于 OLS 算法的 RBF 神经网络 ,具有结构简单 ,学习速度快 ,预测精度高的特点 ,用于时间序列预测获得了十分满意的结果。

(2)由于 RBF 神经网络不存在局部极小问题 ,OLS 算法自

(上接 36 页)

同经典情况类似 , \mathcal{L}_w 逻辑规则可简化为 :

$$\frac{\begin{array}{l} \boxed{\leq i}(\phi_1 \rightarrow_t \phi_2) \\ \hline \begin{array}{l} \boxed{\geq 1-i+j-y} \phi_1 \\ \boxed{\leq i+y} \phi_2 \end{array} \quad \begin{array}{l} y \leq i \\ j \leq i \end{array} \end{array}}{\boxed{\leq 1-i+j} \phi_1} \quad \frac{\begin{array}{l} \boxed{\geq j}(\phi_1 \rightarrow_t \phi_2) \\ \hline \boxed{\leq 1-i+j} \phi_1 \end{array}}{\boxed{\geq y} \phi_2}$$

例 5.给出无穷值 $\mathcal{L}_{\text{ukasiewicz}}$ 逻辑中 ,公式 $P \rightarrow_{\mathcal{L}} (Q \rightarrow_{\mathcal{L}} P)$ 的 tableau 证明。

$$\begin{array}{l} \boxed{\leq c} P \rightarrow_{\mathcal{L}} (Q \rightarrow_{\mathcal{L}} P) \\ (1) \boxed{\geq 1-c+j-y} P \quad y \leq c < 1 \\ \boxed{\leq j+y} Q \rightarrow_{\mathcal{L}} P \quad (2) \quad j \leq c \\ (3) \boxed{\geq 1-j+k-z} Q \quad z \leq j < 1 \\ (5) \boxed{\leq k+z} P \quad (4) \quad k \leq j \end{array}$$

对应的 IP 问题是 :

$$\{c < 1\} \cup \begin{cases} (1) & P & -j & +c & \geq 1 \\ (2) & & -j & +c & \geq 0 \\ (3) & Q & +j & -k & \geq 1 \\ (4) & & j & -k & \geq 0 \\ (5) & \neg P & & +k & \geq 0 \end{cases}$$

表 1

序号	实际值	RBF 预测值	RBF 误差(%)	BP 预测值	BP 误差(%)
1	417	428.4775	2.75	410.7945	1.49
2	391	420.5212	7.55	383.5822	1.90
3	419	455.9659	8.82	468.7093	11.86
4	461	468.2080	1.56	448.2372	2.77
5	472	507.8900	7.60	478.5766	1.39
6	535	532.7150	0.42	544.9934	1.87
7	622	604.2774	2.84	583.6504	6.17
8	606	583.8442	3.66	576.1799	4.92
9	508	513.2552	1.03	461.9083	9.07
10	461	428.5272	7.04	411.8075	10.67
11	390	401.6139	2.98	382.5278	1.92
12	432	435.9920	0.92	448.3727	3.79

动构建近优网络 ,在很大程度上解决了过拟合的问题 ,提高了网络的泛化能力。

(3)该文在预测时对历史数据长度进行了优化 ,比繁琐的试凑更加合理。(收稿日期 2004 年 7 月)

参考文献

- 1.许东等著.基于 MATLAB6.X 系统分析与设计—神经网络[M].西安 :西安电子科技大学出版社 ,2002-09 :19~28
- 2.Chen S C F N Cowan P M Grant.Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks[J].IEEE Transactions on Neural Networks ,1991 2(2) 302~309
- 3.Renate Sitte Joaquin Sitte.Analysis of Predictive Ability of Time Delay Neural Networks Applied to S&P 500 Time Series[J].IEEE Transactions on Systems ,Man and Cybernetics—Part C :Applications and Reviews ,2000 30(4) 568~572
- 4.(美)Martin T Hagan 等著.戴葵等译.神经网络设计[M].北京 :机械工业出版社 ,2002 :198~258

7 小结

该文提出将 tableau 方法与整数规划相结合的 IP-tableau 方法 ,并提升到一阶逻辑、多值逻辑以及无穷值逻辑。将 tableau 转化为 IP 方法后 ,可以采用许多提高 IP 效率的方法。另外在多值逻辑中 ,随着 n 值的增加 ,IP 问题不会变得更加复杂。通过将 tableau 问题转换为规划问题 ,在推理效率方面有较大的提高。(收稿日期 2005 年 2 月)

参考文献

- 1.Zabel R.Proof theory of finity-valued logics[D].Ph D dissertation.Institut für Algebra und Diskrete Mathematic ,TU Wien ,1993
- 2.Fitting M.Types and Tableau[M].Berlin Springer Verlag ,2000
- 3.刘全 ,孙吉贵.非经典逻辑的语义 tableau 方法[J].计算机科学 ,2002 ; 29(5) :72~75
- 4.Beckert B ,Hähnle R ,Manyá F.Transformations between signed and classical clause logic[C].In :Proc of ISMVL 99 ,IEEE Press ,1999 : 248~255
- 5.刘全 ,孙吉贵.提高一阶多值逻辑 tableau 推理效率的布尔剪枝方法[J].计算机学报 ,2003 26(9) :1165~1170
- 6.Beckert B.Depth-first proof search without backtracking for free variable clausal tableaux[J].Journal of Logic and Computation ,2000 ; 10(6) 819~834