

研究简报 ·

基于无先导卡尔曼滤波的 RBFN 训练算法研究

张海涛 李大字* 靳其兵 耿延睿

(北京化工大学信息科学与技术学院, 北京 100029)

摘要: 提出了应用无先导卡尔曼滤波器(UKF)来训练径向基神经网络(RBFN)的新方法。与广义卡尔曼滤波器(EKF)和双重卡尔曼滤波器(DEKF)对函数的一阶近似不同,UKF对非线性函数采用二阶近似展开,而且最重要的一点是不必求取系统的雅可比矩阵,从而大大减小计算量。本文对时间序列预测及分类问题进行了仿真,结果证实了该方法的有效性和快速性。

关键词: 径向基神经网络; 卡尔曼滤波器; 无先导卡尔曼滤波器

中图分类号: TP183

引言

径向基神经网络(RBFN)由Broomhead和Lowe于1988年首次提出,它的数学理论基础是多元函数的插值算法^[1]。RBFN主要应用于将 m 维的输入空间映射到 n 维的输出空间,也用于模式分类^[2]和一些信号控制过程^[3]。

卡尔曼滤波算法(KF)被应用于诸多领域,例如图像处理过程^[4],模糊离散时间动力系统^[5]等。扩展卡尔曼滤波器(EKF)也被广泛应用于工业领域^[6]。无先导卡尔曼滤波器由Julier等首次提出并且在理论上给出了证明^[7],在实际应用中也得到了良好的效果^[8]。

尽管KF,EKF和DEKF都已经被应用于RBFN的训练方面^[9-10],而且已经验证了它们相对于传统训练算法的优越性,但是由于上述三种滤波器算法都需要对非线性函数进行线性化,所以需要耗费很多的时间来进行计算处理。而UKF正是针对这一点做出了有效的改进^[11],而且在函数线性化方面具有很好的效果,并且在估计函数泰勒展开式的协方差方面误差相对较小。

1 基于 UKF 的 RBFN 学习方法

1.1 RBFN的非线性系统模型

因为RBFN可以完成由 m 维输入空间到 n 维输出空间的映射,由此网络可以写成如下的形式

$$y = f(x) \times w \quad (1)$$

其中 $f(\cdot)$ 为径向基函数, w 为输出层权值。实际上由于RBFN中心隐含层的激励函数为非线性的,所以整体网络也是一个非线性的网络。由此,RBFN的整体网络动作可以用离散时间非线性系统表达式表示出来^[9]

$$\begin{cases} x_k = x_{k-1} \\ y_k = h(x_k) \end{cases} \quad (2)$$

其中 $h(\cdot)$ 为状态 x_k 的非线性函数。为了使系统可以使用UKF,所以在系统中要加入模拟噪声信号。可将系统表达式重写为

$$\begin{cases} x_k = x_{k-1} + p_k \\ y_k = h(x_k) + q_k \end{cases} \quad (3)$$

其中 p 和 q 为系统噪声且均为高斯白噪声,协方差分别为 Q 和 R 。将系统状态变量 x 设定为RBFN的中心点和权值组合而成的向量

$$x_k = [w_0^T, w_1^T, \dots, w_c^T, v_2^T, \dots, v_c^T]^T \quad (4)$$

1.2 无先导卡尔曼滤波器算法

无先导卡尔曼滤波器算法步骤为

(1) 随机产生网络中心点位置和权值的大小;

(2) 近似计算系统状态向量:将公式(3)中离散系统的状态向量 x_k 做为滤波器的输入。假设 $M \times 1$ 维 x_k 的均值为 \bar{x}_k ,协方差为 P_k ,则滤波器状态向

收稿日期: 2006-09-18

基金项目: 教育部留学回国人员科研启动基金;北京市教委共建项目建设计划(XK100100435);北京化工大学青年教师基金(QN0625)

第一作者: 男,1982年生,硕士生

*通讯联系人

E-mail: lidz@mail.buct.edu.cn

量 $\bar{x}_{i,k}$ 可由如下公式计算得出

$$\bar{x}_{i,k} = \begin{cases} \bar{x}_k, & i=0 \\ \bar{x}_k + (\sqrt{(M+1)P_x})_i, & i=1, 2, \dots, M \\ \bar{x}_k - (\sqrt{(M+1)P_x})_{i-M}, & i=M+1, M+2, \dots, 2M \end{cases} \quad (5)$$

其中 $\sqrt{(M+1)P_x}$ 是一个标量, 决定了 $\bar{x}_{i,k}$ 相对于 \bar{x}_k 的扩散程度, 通常取 0.001 ~ 1 之间的常数, 为了有利于状态估计, l 通常置为 0;

(3) 预测均值和协方差, 计算公式如下

$$\bar{x}_{i,k|k-1} = \bar{x}_{i,k-1} \quad (6)$$

$$\bar{x}_{k|k-1} = \frac{1}{2M} \sum_{i=0}^{2M} \bar{x}_{i,k|k-1} \quad (7)$$

$$P_{k|k-1} = \frac{1}{2M} \sum_{i=0}^{2M} (\bar{x}_{i,k|k-1} - \bar{x}_{k|k-1})(\bar{x}_{i,k|k-1} - \bar{x}_{k|k-1})^T + Q_k \quad (8)$$

其中 $\frac{1}{2M} = 1 - \frac{1}{2}, i=0;$

$$= \frac{1}{2M}, i=1, 2, \dots, 2M \quad (9)$$

并且 $\bar{x}_{k|k-1}$ 为 \bar{x}_k 相对于 \bar{x}_{k-1} 的值, $P_{k|k-1}$ 为 P_k 相对于 P_{k-1} 的值。

(4) 应用经典卡尔曼计算公式更新系统参数

$$\bar{y}_k = h(\bar{x}_k) \quad (10)$$

$$P_{yy} = h(\bar{x}_k) P_{k|k-1} h(\bar{x}_k)^T + R_k \quad (11)$$

$$P_{xy} = P_{k|k-1} h(\bar{x}_k)^T \quad (12)$$

$$\bar{x}_k = \bar{x}_{k|k-1} + P_{xy} P_{yy}^{-1} (\bar{y}_k - \bar{y}_k) \quad (13)$$

$$P_k = P_{k|k-1} - P_{xy} P_{yy}^{-1} P_{xy}^T \quad (14)$$

1.3 系统协方差矩阵 Q 和 R 的选取

使用 UKF 来训练 RBFN, 当中心点数值和输出层权值都初始化之后, 剩下需要确定的参数就是系统的协方差矩阵 Q 和 R 了。这两个参数对于系统的稳定性是非常重要的, 尤其是对于非线性系统的稳定性。系统的协方差矩阵在计算中的作用相当于在梯度下降法中的动量因子, 所以它们的取值直接关系到整个系统的可靠性和稳定性。假设协方差矩阵为对角矩阵。具体计算方法为^[10]

$$\hat{\Delta}[k] = \hat{\Delta}[k-1] + \mu[k] \{ (y[k] - \bar{y}[k])^2 - \hat{\Delta}[k-1] \} \quad (15)$$

$$R = I \quad (16)$$

其中 y 是网络的目标输出, \bar{y} 为网络的实际输出值。I 为单位矩阵, $\hat{\Delta}$ 为实数 Δ 的回归估计。另外

$$\mu[k] = 1/k \quad (17)$$

1.4 网络训练步骤

(1) 初始化网络的中心点个数及数值、输出层权值、中心激励函数、系统协方差矩阵;

(2) 使用上述算法按照网络目标输出对中心点以及输出层权值进行调节;

(3) 如果达到设定误差, 则跳出, 否则返回 (2);

(4) 如何训练过程中误差突然增大, 则跳出循环, 返回 (1), 重新设定网络相关参数。

2 算法试验仿真

2.1 时间序列预测

选取实际工业函数作为目标函数

$$y_{k+1} = 0.3y_k + 0.6y_{k-1} + f(u_k) \quad (18)$$

其中

$$f(u_k) = 0.6\sin(u_k) + 0.3\sin(3u_k) + 0.1\sin(5u_k)$$

$$u_k = \begin{cases} \sin(2k/250), & k \leq 250 \\ 0.5\sin(2k/250) + 0.5\sin(2k/25), & 250 < k \leq 500 \end{cases}$$

图 1 给出了应用三种不同的训练算法所得到的训练结果, 图 2 为测试结果, 可以看出在设定最终误差相同的情况下, UKF 所用的迭代次数最少。

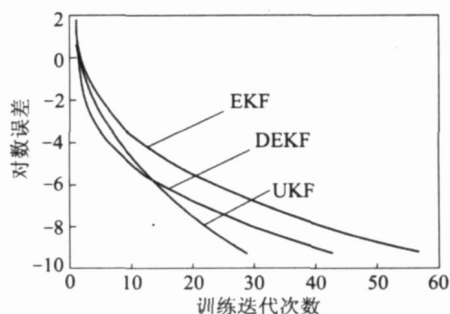


图 1 使用不同的滤波算法的训练结果

Fig. 1 Training results using different filter algorithms

2.2 分类问题

选取经典 Iris 数据作为分类问题的训练集。选取 150 组输入数据, 每一组包括 4 个输入数据, 输入空间为 4×150 的矩阵, 目标输出为

$$\begin{cases} 1 \sim 50 \text{ 组:} & (1 \ 0 \ 0) \\ 51 \sim 100 \text{ 组:} & (0 \ 1 \ 0) \\ 101 \sim 150 \text{ 组:} & (0 \ 0 \ 1) \end{cases}$$

图 3 给出了应用三种不同的滤波算法所得到的

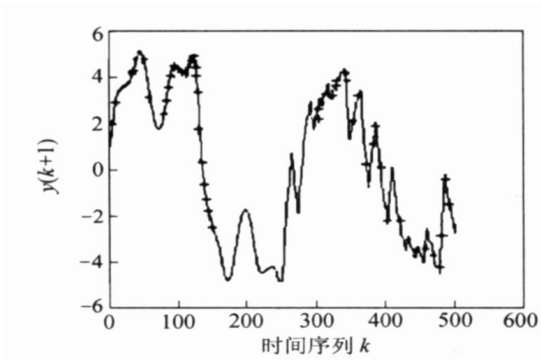


图 2 时间序列预测测试结果

Fig. 2 The test result of Chaotic time series prediction

分类结果误差曲线,其误差范围在 $[1.3 \times 10^{-4}, 2.0 \times 10^{-4}]$ 以内。表 1 给出了相应的网络参数的详细对比,可见 UKF 在中心点个数、迭代次数及误差方面均优于其他两种方法。

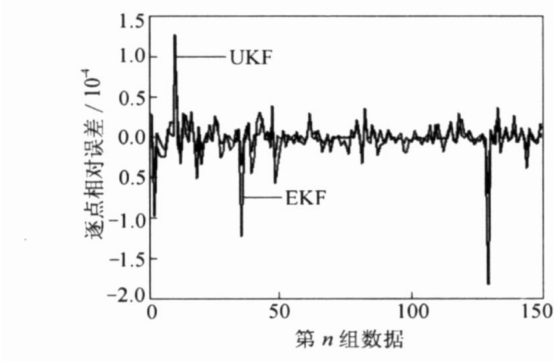


图 3 应用 UKF 和 EKF 对分类问题得训练误差

Fig. 3 Classification error of UKF and EKF

3 算法相关参数讨论

本文选取的中心点激励函数为逆多次函数

$$f(x) = [(x - c) + 1]^{1/(1-m)} \tag{19}$$

其中 c 为激励函数中心点, 1 为正常数,其值在 $0 \sim 1$ 之间; m 为正实数,且 $m \neq 1$;这两个参数对于网络的最终效果起到了至关重要的作用。下面对其做出详细分析。

表 1 分类问题网络参数比较

Table 1 Comparison of net parameters in classification			
滤波器	中心点	迭代次数	总体绝对误差
EKF	110	51	0.0311
DEKF	93	45	0.0111
UKF	71	22	0.0068

3.1 常数

当 α 值较大的时候,函数的整体趋势相对平

缓,反之则变得很“陡峭”。这是由于当 α 值小的时候,相当于对于输入空间的数据聚类半径变小,越是靠近中心点的数据对网络的影响就越大。但是一个激励函数所包含的具有相近数学特性的数据就会相对少,完成相同的任务所需要的中心节点数目会增加,网络收敛特性增强;当 α 小的时候情况则相反。图 4 给出了对于同一问题得不同 α 值的网络收敛特性。经验上, α 值在 0.5 附近最有利于网络整体的性能发挥。

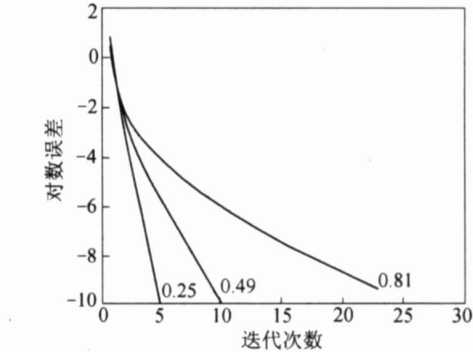


图 4 不同 α 值的训练结果

Fig. 4 Training results with different α

3.2 指数参数 m

当 m 值较大的时候函数图形较为“陡峭”。 m 的作用和 α 基本相似。但是 m 对于网络的影响要比 α 还要更大一些,图 5 给出了当 α 值为 1 的时候对于不同的 m 值所得到的不同的时间序列预测训练误差曲线。明显可以看出当 m 值变大的时候网络会发散。但是需要提到一点:由于 m 和 α 对于网络的性能有着类似的影响,所以它们的协调使用是非常重要的,既要保证整体运算速度,又要保证网络收敛。目前大多采用试凑的方法,对于一般问题 $\alpha = 0.5$ 和 $m = 3$ 就基本可以很好地完成网络任务。

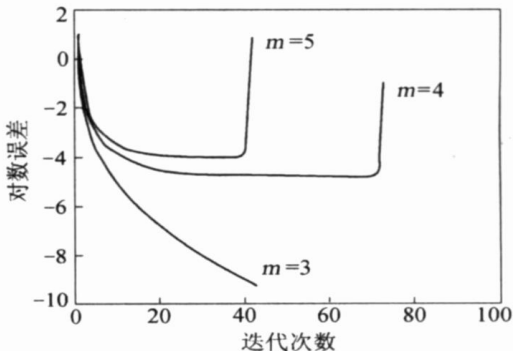


图 5 不同 m 值的训练结果

Fig. 5 Training results with different m

4 结束语

UKF 是一种优越的优化算法,这是因为它不必计算系统的雅克比矩阵的值,且所需要的样本点要相对少,这在一定程度上进一步提高了计算效率。通过将 RBFN 离散模型与 UKF 算法的结合,使得 RBF 训练比传统训练算法更加快捷,更重要的是提高了训练精度和容错能力。仿真结果也证明了 UKF 在 RBFN 的训练方面的有效性和快速性。

参考文献:

- [1] SCHWENKER F, KESTLER H A, PALM G, et al. Three learning phase for radial basis function networks [J]. *Neural Networks*, 2001, 14(4-5): 439-458.
- [2] SARIMVEIS H, DOGANIS P, ALEXANDRIDIS A, et al. A classification technique based on radial basis function neural networks[J]. *Advances in Engineering Software*, 2006, 37(4): 218-221.
- [3] KARA YIANNIN. Reformulated radial basis neural networks trained by gradient descent[J]. *Neural Networks*, 1999, 3(5): 657-671.
- [4] PIOVOSO M, PHILLIP A, LAPLANTE L, et al. Kalman filter recipes for real-time image processing[J]. *Real-Time Imaging*, 2003, 9(6): 433-439.
- [5] SIMON D. Kalman filtering for fuzzy discrete time dynamic systems [J]. *Applied Soft Computing Journal*, 2003, 3(3): 191-207.
- [6] BOJE E, PETRICK M. Application of the extended Kalman filter to a lysine hydrochlorination process[J]. *Control Engineering Practice*, 2000, 8(3): 291-297.
- [7] JULIER S, HLMANN J. A new approach for filtering nonlinear systems [C]. Piscataway, NJ, USA: IEEE Press, 1995: 1628-1632.
- [8] LI Peihua, ZHANG Tianwen, MA Bo. Unscented kalman filter for visual curve tracking[J]. *Image and Vision Computing*, 2004, 22(2): 157-164.
- [9] SIMON D. Training radial basis neural networks with the extended kalman filter[J]. *Neurocomputing*, 2002, 48(1-4): 455-475.
- [10] IULIAN B C. RBF networks training using a dual extended kalman filter[J]. *Neurocomputing*, 2002, 48(1-4): 609-622.
- [11] ROMANENKO A, CASTRO JOSE A A M. The unscented filter as an alternative to the EKF for nonlinear state estimation: a simulation case study[J]. *Computers and Chemical Engineering*, 2004, 28(3): 347-355.

Training radial basis neural networks with the unscented Kalman filter

ZHANG HaiTao LI DaZi JIN QiBing GEN G YanRui

(College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China)

Abstract: A new method is proposed for training radial basis function networks (RBFN) using the unscented Kalman filter (UKF). In contrast to the extended Kalman filter (EKF) and the dual extended Kalman filter (DEKF), which extend the nonlinear functions using a first order approximation, the UKF uses a second order approximation. The most important consequence is that the algorithm does not require the Jacobi matrix of the system to be calculated, thus reducing the calculation complexity and resulting in considerable savings in time. Simulation results in the fields of chaotic time series prediction and classification problems demonstrate both the validity and faster speed of the proposed method.

Key words: RBFN; Kalman filter; unscented Kalman filter