第 13 卷第 2 期
1996 年 4 月

控制理论与应用
CONTROL THEORY AND APPLICATIONS

Vol. 13, No. 2
Apr., 1996

A New Fast Learning Algorithm for Feedforward Neural Networks Using U-D Factorization-Based Extended Kalman Filter *

ZHANG Youmin, DAI Guanzhong and ZHANG Hongcai

(Department of Automatic Control, Northwestern Polytechnical University · Xi'an, 710072, PRC)

Abstract: A New fast learning algorithm for training multilayer feedforward neural networks by using variable time-varying forgetting factor technique and U-D factorization-based fading memory extended Kalman filter is proposed in this paper. In comparison with BP and extended Kalman filter (EKF) based learning algorithm, the new algorithm can not only obviously improve the convergency rate, numerical stability, but also provide much more accurate learning results in fewer iterations with fewer hidden nodes. In addition, it is less affected by the choice of initial weights and initial covariance matrix as well as other setup parameters. The results of simulated computations of nonlinear dynamic system modelling and identification applications show that the new algorithm proposed here is an effective and efficient learning algorithm for feedforward neural networks.

Key words: feedforward neural networks; BP learning algorithm; extended Kalman filtering algorithm; U-D factorization; time-varying forgetting factor

1 Introduction

The classical method for training a multilayer feedforward network is the backpropagation (BP) algorithm[1]. Althouth it is successfully used in many cases, the BP algorithm suffers from a number of shortcomings. One such shortcoming is the slow convergency rate. Many iterations are required to train small networks for even the simplest problems. Furthermore, we have to tune the learning rate and the momentum term in a heuristic manner so that a quick convergence is obtained. An improper choice of these parameters may incur problems of stability or suffer from much slow convergence and may usually be local. Numerous heuristic schemes have been suggested to improve the speed of convergence of the BP algorithm[2~4]. These algorithms have considerably improved the convergence of BP algorithm and exhibit excellent performance. But these algorithms increase the storage and computational cost which may become unmanageable even for networks of moderate size, besides, the convergency rate of these algorithms needs further improving.

* This Project is Supported by National Natural Science Foundation of China (grants 69274015).

Manuscript received Jun. 16, 1994, revised Feb. 24, 1995.

It is the purpose of this paper to present a new alternative algorithm which is considerably faster than the BP and EKF algorithms, and has the advantage of being less affected by poor initial weights and setup parameters. In view of this, a new fast learning algorithm for traning multilayer feedforward neural networks by using U-D factorization-based fading memory extended Kalman filter is proposed in this paper. The performance comparison of new algorithm proposed in this paper with EKF and BP algorithms are given by two nonlinear system modelling and identification examples.

2 Feedforward Neural Networks

The neural networks considered in this paper are feedforward networks with one or more hidden layers between the inputs and outputs. Each layer consists of some computing units known as neurons. Fig. 1 shows the structure of a multilayer neural network.

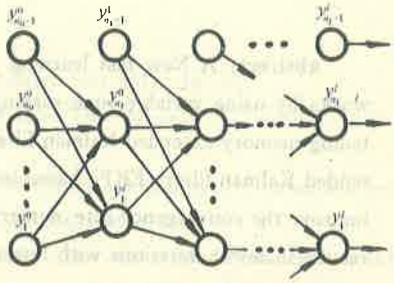

Fig. 1 Amultilayer feed forward neural network

The BP algorithm has become the standard algorithm used for training multilayer feedforward neural networks. However, in system modelling and identification applications the dynamic range of the output data may be greater than 1, the activation function of the output nodes is chosen to be linear, thus, we can acquire the modified BP algorithm, a detailed algorithm implementation can be acquired in paper[4].

3 U-D Factorization-Based Fading Memory Extended Kalman Filter Learning Algorithm

3.1 Extended Kalman Filter Based Learning Algorithm

A feedforward nerual network is trained by adjusting its weight according to an ongoing stream of input-output observation $\{u_d^l(t), y_d^l(t) : t = 1, \cdots, N\}$, the objective of this adaption is to learn a set of weights $\{w^1, w^2, \cdots, w^N\}$ such that, the NN will accurately predict future outputs. Although this training mechanism is generally referred to as supervised learning, it is equivalent to the process of parameter estimation. Then, training feedforward networks can be considered as a nonlinear estimation problem where the weight values are unknown and need to be estimated for the given set of input-output vectors. The approach is to concatenate all the network parameters into a vector $\theta(t)$ and to define an operator, NET($\cdot$), which performs the complete mapping function of a multilayer feedforward NN on the basis of this parameter vector and the input, $u(t) = y^0(t)$, according to

$$y = \mathrm{NET}(\theta, u) = f^l(w^l f^{l-1}(w^{l-1} \cdots f^1(w^1, y^0) \cdots)). \tag{1}$$

The parameter estimation problem is then defined as the determination of $\theta(p)$ which minimizes the sum of square prediction error of all prior observations embedded in the function:

$$J(p) = \frac{1}{2}\sum_{i=1}^{p}(y_d(t) - \mathrm{NET}(\theta(p), u(t))^2\lambda^{p-1} \quad (2)$$

where $y_d(t)$ is the observed output vector and p denotes the number of training samples. Assume that the nonlinearities of (1) are sufficiently smooth, and that the observation is of the form:

$$y_d(t) = \mathrm{NET}(\theta^*(t), y^0(t)) + \varepsilon(t) = y^1(t) + \varepsilon(t). \quad (3)$$

where $\theta^*(t)$ is the desired parameter vector that minimizes (2) and $\varepsilon(t)$ is the measurement noise with zero mean and a covariance matrix $R(t)$.

Then the extended Kalman filter can be applied to the estimation problem through the equations:

$$\bar{\theta}(t) = \bar{\theta}(t-1) + K(t)[y_d(t) - y^1(t))], \quad (4)$$

$$K(t) = P(t-1)H^T(t)[H(t)P(t-1)H^T(t) + \lambda(t)R(t)]^{-1}, \quad (5)$$

$$P(t) = [I - K(t)H(t)]P(t-1)/\lambda(t). \quad (6)$$

where $\bar{\theta}(t)$ is the estimated parameter vector, $P(t)$ is the error covariance matrix, $H(t)$ is the Hessian matrix containing the partial derivative of the network output with respect to each individual weight, and $\lambda(t)$ is a time-varying forgetting factor. When it is required to weight out the old data, it is required that $\lambda(t) < 1$. In order to improve the convergence and accuracy of learning algorithm, two kinds of forgetting factors are given in this paper as follwos:

Ⅰ) Assume $\lambda(t) = 1$ in equations (5) and (6), and chose the noise covariance matrix $R(t) = e^{-t/N}I$ instead of $R(t) = I$. Among a learning sample, $R(t)$ is set to be constant.

Ⅱ) Set $R(t) = I$, assume time-varying forgetting factor as follows:

$$\lambda(t) = \lambda_0\lambda(t-1) + (1-\lambda_0) \quad (7)$$

where λ_0 and the initial forgetting factor $\lambda(0)$ are design value.

3.2 U-D Factorization-Based fading memory EKF Learning Algorithm

In above fading memory extended Kalman filter, the computation of the covariance matrix $P(t)$ plays an important role. Its computational load is the major consideration of the algorithm. In addition, the limitation of the word length, the truncation errors of the algorithm may make the covariance matrix $P(t)$ to become non-symmetric and non-positive definite, and may even result in divergence of the estimates. Thus several covariance matrix factorization methods have been developed to improve the numerical properties for the computation of $P(t)$. The U-D factorization method is the most popular one[5]. This decomposition guarantees the positive-definiteness of covarivance matrix $P(t)$, and thus high estimation accuracy and robustness can be attained[5,6].

By using the U-D factorization technique, and suppose feedforward network has single output, that is, $H(t)$ is a vector, the above fading memory EKF algorithm Eqs. (4) to (6) can be described as follows:

Step 1 Compute

$$f = U^{\mathrm{T}}(t-1)H(t), g = D(t-1)f, \quad \alpha_0 = \lambda(t)R(t);$$

Step 2 For $j = 1, \cdots, n_g$, Compute steps 3~5;

Step 3 Compute

$$\begin{cases} \alpha_j = \alpha_{j-1} + f_j g_j, \\ D(t)_{jj} = \alpha_{j-1} D(t-1)_{jj}/\alpha_j \lambda(t), \\ v_j = g_j, \\ \mu_j = -f_j/\alpha_{j-1}. \end{cases} \tag{8}$$

Step 4 For $i = 1, \cdots, j-1$, do step 5 (if $j = 1$, then skip step 5).

Step 5 Compute

$$\begin{cases} U(t)_{ij} = U(t-1)_{ij} + v_i \mu_i, \\ v_i = v_i + U(t-1)_{ij} v, \end{cases} \tag{9}$$

Step 6 $\bar{K}(t) = [v_1, \cdots, v_n]^{\mathrm{T}}, \quad K(t) = \bar{K}(t)/\alpha_{n_g}$. (10)

Step 7 $\bar{\theta}(t) = \bar{\theta}(t-1) + K(t)[y_d(t) - y^1(t)]$. (11)

where, if the root mean square error (RMSE) between desired and actual output is larger than some expected value (for example, 10^{-3}), $\lambda(t)$ is computed by equation (7), otherwise, set $\lambda(t) = 1.0$. This technique called "variable" time-varying forgetting factor can obviously smooth out weight changes, and the convergence and accuracy of learning algorithm can be improved greatly.

4 Simulation Examples and Results

In this section, simulation results from two nonlinear plant identification problems are presented, and the performances of new algorithm proposed in this paper (EKF-UD), extended Kalman filter (EKF) and BP algorithm are compared.

Example 1 Consider the nonlinear system as follows:

$$y(t) = 2.0\frac{u(t)}{1 + u^2(t)} \tag{12}$$

where $y(t) \in [-1, 1]$ when $u(t) \in [-10, 10]$. The EKF-UD, EKF and BP algorithms are employed to train a neural network with an architecture of 1—10—1 and consisting of total of 30 adjustable weights to approximate (12). Table 1 shows the root mean square error (RMSE) of three learning algorithms with different iteration. For EKF-UD and EKF algorithms, case 1 denotes the situation that there is no forgetting factor, that is to say the forgetting factor is equal to 1, case 2 represents the first forgetting factor and case 3 represents the second forgetting factor. Fig. 2 illustrates the RMSE and output error curve via different algorithms for different hidden nodes, initial parameter and forgetting factor. It is clear that the learning accuracy and convergence of new algorithm proposed in this paper are much better than EKF and BP algorithms, especially for case 3, the convergency and accuracy are the best.

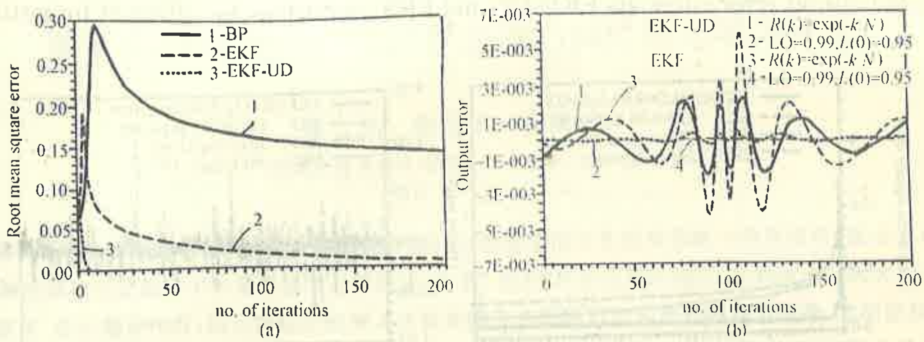

Fig. 2 Comparison of RMSE and output error versus training iteration of three algorithms for example 1

Table 1 Comparison of RMSE versus training iteration of three algorithms

Iteration	BP	EKF			EKF-UD		
		CASE 1	CASE 2	CASE 3	CASE 1	CASE 2	CASE 3
10	2.941E−1	3.174E−2	7.997E−2	1.382E−3	1.004E−2	9.633E−3	2.874E−4
20	2.493E−1	1.856E−2	5.619E−2	2.395E−4	5.953E−3	5.384E−3	7.738E−5
100	1.585E−1	6.495E−3	1.672E−2	9.079E−5	2.776E−3	2.214E−3	1.745E−6
200	1.363E−1	4.800E−3	5.322E−3	8.177E−5	2.278E−3	1.074E−3	5.629E−7

In order to compare the convergency rate and computing requirement of different algorithms with same learning accuracy, we suppose RMSE of desired and actual output be equal to 10^{-2}, 10^{-3}, 10^{-4} and 10^{-5} respectively, the required iteration number for EKF-UD, EKF and BP are listed in Table 2

Table 2 Comparison of the iteration number of three algorithms

RMSE	BP	EKF			EKF−UD		
		CASE 1	CASE 2	CASE 3	CASE 1	CASE 2	CASE 3
10^{-2}	—	48	33	6	11	10	4
10^{-3}	—	—	280	12	—	210	5
10^{-4}	—	—	—	79	—	587	16
10^{-5}	—	—	—	—	—	1267	26

where "—" denotes that the corresponding algorithm hasn't reached the expected RMSE within 200000 iterations for BP and 2000 iterations for EKF, EKF-UD respectivelly.

Example 2 Consider a nonlinear dynamic system as follows:

$$y(t) = (0.8 - 0.5e^{-y^2(t-1)})y(t-1) - (0.3 + 0.9e^{-y^2(t-1)})y(t-2) + u(t-1) + 0.2u(t-2) + 0.1u(t-1)u(t-2) + \varepsilon(t). \tag{13}$$

where $\varepsilon(t)$ is gaussian white noise sequence with zero mean and variance 0.04, and the system input $u(t)$ is an independent sequence of uniform distribution with zero mean and variance 1.0. The architecture of neural network is 4—5—1. Fig. 3 illustrates the RMSE curve and output error curve via EKF-UD and EKF algorithms for different forgetting factors.

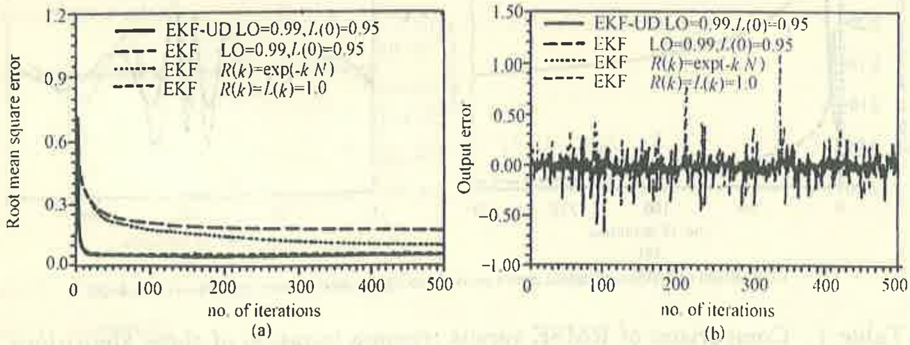

Fig. 3 Comparison of RMSE and output error versus training iteration of two algorithms for example 2

The results of simulated computations of nonlinear static and dynamic system modelling and identification applications show that the new EKF-UD algorithm proposed here is an effective and efficient learning algorithm for feedforward neural networks.

5 Conclusions

A new fast learning algorithm for training multilayer feedforward NN by using U-D factorization-based fading memory extended Kalman filter is proposed in this paper. Two kinds of time-varying forgetting factors are presented. By using U-D factorization and variable time-varying forgetting factor technique, the new algorithm can greatly improve the convergency rate, numerical stability of BP and EKF algorithms, and provide much more accurate results in fewer iterations with fewer hidden nodes. In addition, it is less affected by the choice of initial weights and initial covariance matrix. It is shown that the new algorithm is an effective and efficient learning algorithm for feedforward neural network.

References

[1] Rumelhart, D. E. and McClelland, J.. Parallel Distributed Processing. Vol. 1, Cambridge, MA: MIT Press, 1986

[2] Singhal, S. and Wu, L.. Training Feed-Forward Networks with the Extended Kalman Algorithm. Proc. Int. Conf. on ASSP, 1989, 1187—1190

[3] Scalero, R. S. and Tepedelenlioglu, N.. A Fast New Algorithm for Training Feedforward Neural Networks. IEEE Trans. on Signal Processing, 1992, 40(1): 202—210

[4] Zhang, Y. M., Dai, G. Z. and Zhang, H. C.. A Extended Kalman Filtering Learning Algorithm for Feedforward Neural Networks. Information and Control, 1994, 23(2): 113—118(In Chinese)

[5] Bierman, G. J., Factorization Methods for Discrete Sequential Estimation. New York: Academic Press, 1977

[6] Ljung, L. and Soderstrom, T.. Theory and Practice of Recursive Indentification. Cambridge, MA: MIT press, 1983

基于U-D分解推广卡尔曼滤波的神经网络学习算法

张友民 戴冠中 张洪才
(西北工业大学自动控制系·西安,710072)

摘要:本文针对前馈神经网络BP算法所存在的收敛速度慢且常遇局部极小值等缺陷,提出一种基于U-D分解的渐消记忆推广卡尔曼滤波学习新算法.与BP和EKF学习算法相比,新算法不仅大大加快了学习收敛速度、数值稳定性好,而且需较少的学习次数和隐节点数即可达到更好的学习效果,对初始权值,初始方差阵等参数的选取不敏感,便于工程应用.非线性系统建模与辨识的仿真计算表明,该算法是提高网络学习速度、改善学习效果的一种非常有效的方法.

关键词:前馈神经网络;BP学习算法;推广卡尔曼滤波;U-D分解;时变遗忘因子

本文作者简介

张友民 见本刊1996年第1期第114页.
戴冠中 见本刊1996年第1期第114页.
张洪才 见本刊1996年第1期第114页.

(上接第204页)

GPS实验室开发的“汉字输入输出型便携机”目前国内外尚无类似产品报道,在94年春季广交会上受到沿海国家参观人士的注意,意欲修改为该国文字的输入;在94年“中国大学生第二届应用科技发明大奖赛”中获大奖.“海图型船用导航仪”(Ⅰ型产品,现已出Ⅱ型)于1995年6月在南京获“95中国高新技术、新产品博览会”(国家科委主办)金奖,并获得金杯.

和国内进口或组装的生产单位相比,中国科学技术大学GPS实验室的所有样品(及产品)全是独立自主设计、软件模块化,可以随时按照用户的特殊要求增加功能,改变配置,特别适合中国国情.和国外产品相比,在各项技术指标均不低于国外同等产品外,我们的汉字输入输出功能、中国地区电子地图的生成、软件开发的成本与周期、汇编级的液晶显示屏弹出式窗口操作界面与动画,目前也是国内外其他产品所不具备的.但是,我们有自己的不足之处:作为成熟的成品生产,我们的工艺水平(包括外壳工艺与底板加工)与国外产品相比尚有明显差距,需要先进工艺的支持.

GPS实验室的近期目标是相继推出现有的各类型产品,不断完善改型,提高质量,保持GPS技术开发的国内领先水平,并继续开拓GPS应用新领域.96年的研制重点将转向差分GPS产品,建立地区性的单边带差分GPS修正伪距信号通讯网.

GPS实验室根据对国外情况的了解及国内需求的分析,考虑到GPS作用的重大及应用领域的广阔,可以断言今后五年内GPS产品在中国市场上的前景将是“仅受人们的想象力制约”的.以每100个城市人口拥有1台GPS(包括车用交通管理与单机使用)计算,市场就是300亿元人民币.目前中国GPS所处的形势,很象六年前的电话和三年前的BB机,这两个市场分别被AT&T公司及Motorola公司为主占领了.但GPS涉及到中国地图和中国用户的操作方式,这一更大的市场只可能由中国的GPS研制开发单位加上外国公司的支持(资金与管理)共同占领.中国科学技术大学GPS实验室以科大的技术力量为依托,希望在今后五年内在中国GPS产业的发展过程中起到一个先锋的作用.

(下转第247页)