Proceedings of the 29th Chinese Control Conference
July 29-31, 2010, Beijing, China

On Neural Network Training Algorithm Based on the Unscented Kalman Filter*

LI Hongli[1,2], WANG Jiang[1], CHE Yanqiu[1], WANG Haiyang[1], CHEN Yingyuan[1]
1. School of Electrical and Automation Eng., Tianjin University, Tianjin 300072, P. R. China
E-mail: jiangwang@tju.edu.cn
2. School of Electrical Engineering and Automation, Tianjin Polytechic University, Tianjin 300160, P. R. China
E-mail: lihongli@tjpu.edu.cn

Abstract: Neural network has been widely used for nonlinear mapping, time-series estimation and classification. The backpropagation algorithm is a landmark of network weights training. Although the vast weights update algorithms have been developed, they are often plagued by convergence to poor local optima and low learn velocity. The unscented Kalman filter is a nonlinear parameter estimation algorithm. By means of it, weights update can be realized. Higher training velocity and mapping accuracy of network can be obtained. The numerical simulation results show the effectiveness of the algorithm compared with the standard backpropagation.

Key Words: Neural Network, Unscented Kalman Filter, Extended Kalman Filter, Backpropagation

1 INTRODUCTION

Neural network(NN) is a nonlinear system composed of many neurons. the neural network can simulate human brain neural system to some extent. It plays an essential role in information processing, storage and searching. Thus NN has some intelligent functions, such as learning, memory and computation etc. It has been widely used in nonlinear mapping, time-series estimation and classification. an widely used network training algorithm is BP algorithm which enhanced training procedures [1]. However, BP is only a simple gradient method based on first-order derivative information of errors. Some enhanced training methods have been introduced by many researchers. The most famous of enhanced training method is the second-order derivative weight update procedure. All these methods exhibit superior capabilities in terms of training speed, mapping accuracy and generalization. The vast majority of these methods are batch update algorithms in essence. However, convergence to poor local optima and lower learning velocity confine the application of these algorithms. The classical Kalman filter(KF) [2], which is based on the state-space equation of linear dynamical systems, provides a recursive solution to the linear optimal filtering estimation problem. It has addressed the estimation of a state vector in a linear model of a dynamical system. If the model is nonlinear, the extended Kalman filter(EKF) [3] [4], based on a linearization procedure, can be used. The EKF, an effective second-order algorithm, can be applied to estimate the weights of NN. However, the EKF provides only an approximation to optimal nonlinear estimation. These approximations can induce large errors which may lead to suboptimal performance and sometimes make the system divergence. It has the underlying assumptions and flaws that can be addressed by using the unscented Kalman(UKF). The UKF [5] [6] is an alternative filter with performance superior to that of the EKF. General application areas of the UKF are state estimation, parameter estimation and dual estimation problems. The UKF has been applied to auto-control, signal filter, pattern recognition, brain information process and other aspects. In this paper the UKF is used to learn the weights of a NN. The simulation results show that NN based on the UKF has higher training velocity and mapping accuracy compared with the standard BP.

2 NEURAL NETWORK

NN is composed of many nonlinear nodes (called neurons), which are connected through the weights. All weights (including bias of all neurons) must be modified before NN is able to be used. There are two types of network architecture: the well-known forward network and the feedback network. Once trained, the forward network merely carries out a static mapping from input signals to outputs, such that the output is independent of the history in which input signals are presented. On the other hand, a trained feedback network provides a dynamic mapping, therefore the output is not only a function of the current input pattern, but also implicitly a function of the entire history of inputs through the time-delayed recurrent node activations [7] [8]. The forward NN can be used for nonlinear mapping, time-series estimation and classification. Whereas, the feedback NN can be used for optimization. There are vast majority of NN weights training methods among which the BP is the most famous. In the standard BP algorithm, weights are modified by utilizing first-order derivative information of errors. Weights are updated along the negative gradient of the cost function. Although some enhanced methods are based upon second-order derivative information, most of them are often plagued by convergence to poor local optima or lower learning velocity. In this paper a three layers forward network is discussed. The weights learning algorithm is based on the UKF. Figure 1 shows an example of a three layers forward network with one input layer, one hidden layer and one output layer. The NN is used to approximate a nonlinear function and solve a three classification problem.

*This work is supported by the Key National Natural Science Foundation of China (Grant No. 50537030), the National Natural Science Foundation of China (Grant Nos. 50707020, 60901035 and 50907044) and the Postdoctoral Science Foundation of China (Grant Nos. 20070410576, 200801212, 20080430090, 20080430731 and 200902275).

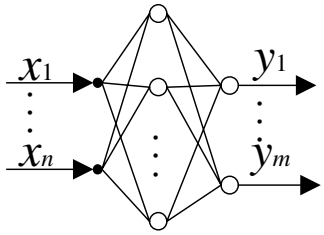

Fig. 1 The diagram of a feedforward network architecture

3 THE UKF FILTER

The state estimation problem can be described by Eq.(1). The first equation of Eq.(1) is the state equation while the second is the observation equation.

$$\begin{aligned} \boldsymbol{x}(t) =& \boldsymbol{f}\left(\boldsymbol{x}(t-\Delta t), \boldsymbol{u}(t)\right) + \boldsymbol{r}(t) \\ \boldsymbol{y}(t) =& \boldsymbol{h}\left(\boldsymbol{x}(t)\right) + \boldsymbol{e}(t) \end{aligned} \tag{1}$$

Where $\boldsymbol{x}(t)$ is the state vector to be estimated, $\boldsymbol{y}(t)$ is the measurement vector, $\boldsymbol{u}(t)$ is the control inputs. Suppose that the known data are the initial conditions $\boldsymbol{x}(0)$, the measurement $\boldsymbol{y}(t)$, and the control inputs $\boldsymbol{u}(t)$, If there are no exogenous signals, $\boldsymbol{u}(t)$ can be set to 0. It is desired to obtain estimation of the state vector $\boldsymbol{x}(t)$. $\boldsymbol{r}(t)$(process noise) and $\boldsymbol{e}(t)$(measurement noise) are assumed zero mean white Gaussian noise. $\boldsymbol{f}$ and $\boldsymbol{h}$ are the functions of the state and the observation vector, respectively. The famous Kalman filter addresses state or parameter estimation problem when $\boldsymbol{f}$ and $\boldsymbol{h}$ are linear. It provides a recursive solution to the linear optimal state estimation problem. The solution is recursive in that each updated estimate of the state is computed from the previous estimation and the new input data. If, however, the eqation is nonlinear, the Kalman filter has to be developed into the EKF based on the linearization of the nonlinear section(first-order Taylor series expansion) [9]. The problem about the EKF is that if nonlinearities can not be approximated well by linearized terms, most EKF estimates are biased and inconsistent. It provides only a first order approximation to optimal nonlinear estimation. Besides Jacobian or Hessian matrix calculations are necessary for the application of the EKF. The calculation of the above two matrix is very difficult for the complicated nonlinear system. A novel procedure(UKF) for dealing with estimation in strong nonlinear state space models has been proposed recently by Julier and Uhlman. The UKF, base on the unscented transformation(UT), can provide performance superior to that of the EKF. It achieves second-order accuracy for nonlinear systems with no Jacobian or Hessian matrix to be calculated.

3.1 UT

The UT is a method for calculating the statistics of a random variable which undergoes a nonlinear transformation [10]. It uses a minimal set of carefully chosen sample points to reprent the state distribution. These sample points completely capture the true mean and covariance of the Gauss random variable(GRV), and can achieve second-order estimation accuracy for any nonlinear system. These sample points is given by the so-called sigma points as shown in Eq.(2).

$$\begin{aligned} \boldsymbol{\chi}_0(t-\Delta t|t-\Delta t) &= \hat{\boldsymbol{x}}(t-\Delta t|t-\Delta t) \\ \boldsymbol{\chi}_i(t-\Delta t|t-\Delta t) &= \hat{\boldsymbol{x}}(t-\Delta t|t-\Delta t) \\ &\quad + \left[\sqrt{(n+k)P(t-\Delta t|t-\Delta t)}\right]_i \\ \boldsymbol{\chi}_{i+n}(t-\Delta t|t-\Delta t) &= \hat{\boldsymbol{x}}(t-\Delta t|t-\Delta t) \\ &\quad - \left[\sqrt{(n+k)P(t-\Delta t|t-\Delta t)}\right]_i \end{aligned} \tag{2}$$

where $i = 1, 2..., n$, n is the dimension of $\boldsymbol{x}(t)$ and $\left[\sqrt{(\cdot)}\right]_i$ is either the ith row or column of the matrix square root. k is a scaling parameter, which is set to zero in this paper. There are altogether $2n+1$ points that have to be computed.

3.2 UKF Algorithm

The UKF is a straightforward extension of the UT to the recursive estimation in Eq.(1). Along with Eq.(2), the UKF equation can be expressed as the following recursion [3, 11].

$$\begin{aligned} \boldsymbol{\chi}_i(t|t-\Delta t) =& \boldsymbol{f}\left(\boldsymbol{\chi}_i(t-\Delta t|t-\Delta t), \boldsymbol{u}\right) \\ \boldsymbol{y}_i(t|t-\Delta t) =& \boldsymbol{h}\left(\boldsymbol{\chi}_i(t|t-\Delta t)\right) \end{aligned} \tag{3}$$

$$\begin{aligned} \hat{\boldsymbol{x}}(t|t-\Delta t) &= \sum_{i=0}^{2n} W_i \boldsymbol{\chi}_i(t|t-\Delta t) \\ \hat{\boldsymbol{y}}(t|t-\Delta t) &= \sum_{i=0}^{2n} W_i \boldsymbol{y}_i(t|t-\Delta t) \end{aligned} \tag{4}$$

$$W_0 = \frac{k}{n+k}, W_i = \frac{1}{2(n+k)} \tag{5}$$

$$\begin{aligned} P(t|t-\Delta t) =& \sum_{i=0}^{2n} W_i \left[\boldsymbol{\chi}_i(t|t-\Delta t) - \hat{\boldsymbol{x}}(t|t-\Delta t)\right] \\ & \left[\boldsymbol{\chi}_i(t|t-\Delta t) - \hat{\boldsymbol{x}}(t|t-\Delta t)\right]^T \\ P_{yy}(t|t-\Delta t) =& \sum_{i=0}^{2n} W_i \left[\boldsymbol{y}_i(t|t-\Delta t) - \hat{\boldsymbol{y}}(t|t-\Delta t)\right] \\ & \left[\boldsymbol{y}_i(t|t-\Delta t) - \hat{\boldsymbol{y}}(t|t-\Delta t)\right]^T \\ P_{xy}(t|t-\Delta t) =& \sum_{i=0}^{2n} W_i \left[\boldsymbol{\chi}_i(t|t-\Delta t) - \hat{\boldsymbol{x}}(t|t-\Delta t)\right] \\ & \left[\boldsymbol{y}_i(t|t-\Delta t) - \hat{\boldsymbol{y}}(t|t-\Delta t)\right]^T \end{aligned} \tag{6}$$

$$\begin{aligned} K(t) =& P_{xy}(t|t-\Delta t) P_{yy}^{-1}(t|t-\Delta t) \\ \hat{\boldsymbol{x}}(t|t) =& \hat{\boldsymbol{x}}(t|t-\Delta t) + K(t)\left[\boldsymbol{y}(t) - \boldsymbol{h}\left(\hat{\boldsymbol{x}}(t|t-\Delta t)\right)\right] \\ P(t|t) =& P(t|t-\Delta t) - K(t) P_{yy}(t|t-\Delta t) K^T(t) \end{aligned} \tag{7}$$

where K is the called Kalman gain matrix, the carets indicate the mean of the corresponding density function, and P is the covariance matrix of the vector of estimation errors. Moreover, the notation $\boldsymbol{z}(t|t-\Delta t)$ indicates the value of the quantity $\boldsymbol{z}$ at time t using information taken up to time $t-\Delta t$. Likewise, $\boldsymbol{z}(t|t)$ indicates the value of $\boldsymbol{z}$ computed at time t using the information available up to and including time t. A second-order accuracy estimation of the state vector $\boldsymbol{x}(t)$ can be achieved by using the recursive Eq.(2)-Eq.(7). The UKF has been applied to state estimation, parameter estimation, and dual estimation [12, 13].

4 NN TRAINING ALGORITHM BASED ON THE UKF

NN's behavior can be described by the following equation.

$$\begin{aligned} \boldsymbol{W}(t) =& \boldsymbol{W}(t-\Delta t) + \boldsymbol{r}(t) \\ \boldsymbol{y}(t) =& \boldsymbol{G}\left(\boldsymbol{W}(t)\right) + \boldsymbol{e}(t) \end{aligned} \tag{8}$$

Where $\boldsymbol{W}(t)$ is the weights of NN, $\boldsymbol{y}(t)$ is the desired output, $\boldsymbol{r}(t)$(process noise) and $\boldsymbol{e}(t)$(measurement noise) are zero mean white Gaussian noise. The first equation can be regarded as the process equation. Whereas, the second equation is regarded as a nonlinear observation equation. Accordingly, $\boldsymbol{W}(t)$ can be regarded as the states to be estimated. $\boldsymbol{y}(t)$ is the observation on $\boldsymbol{W}(t)$. So the UKF can be used to address the NN weights training problem. The weights update algorithm based on the UKF is an incremental training method, which can avoid the poor local minima to some extent.

A three layers NN is trained in this paper using both the UKF and BP algorithm. In the UKF weights updating algorithm, all the weights(including bias) are arrayed in a column vector as Eq.(9). Then the UKF is applied to estimate the weights vector.

$$\begin{aligned} \boldsymbol{W} =& [\omega_{11}^{(1)}, \omega_{12}^{(1)}, \cdots, \omega_{1k}^{(1)}, \\ & \omega_{21}^{(1)}, \omega_{22}^{(1)}, \cdots, \omega_{2k}^{(1)}, \\ & \cdots, \\ & \omega_{n1}^{(1)}, \omega_{n2}^{(1)}, \cdots, \omega_{nk}^{(1)}, \\ & \omega_{11}^{(2)}, \omega_{12}^{(2)}, \cdots, \omega_{1m}^{(2)}, \\ & \omega_{21}^{(2)}, \omega_{22}^{(2)}, \cdots, \omega_{2m}^{(2)}, \\ & \cdots, \\ & \omega_{k1}^{(2)}, \omega_{k2}^{(2)}, \cdots, \omega_{km}^{(2)}, \cdots,]^T \end{aligned} \tag{9}$$

Where m is the number of inputs, k is the number of hidden layer neurons, and n is the number of output neurons. The superscript (1) of ω denotes that ω is the weight between the input layer and the hidden layer. The superscript (2) denotes the weight between the hidden layer and the output layer. Whereas, the subscript ij of ω refers to the weight between the ith neurons of the front layer and the jth neurons of the next layer.

5 ALGORITHM SIMULATION

NN based on the UKF can be used for nonlinear mapping, time-series estimation and classification, etc. In this section we present two simulation examples of NN. The first example is about a nonlinear mapping, in which NN is used to approximate a nonlinear polynomial. In the second example NN solves a three classification problem. The initial values of the weights are the random numbers between -1 and 1. The initial value of the covariance matrix P is identity matrix.

5.1 NONLINEAR MAPPING

$$f(x) = 1.1(1 - x + 2x^2)e^{(-\frac{x^2}{2})} \tag{10}$$

An important application of NN is the nonlinear mapping in which NN is used to approximate a nonlinear polynomial.

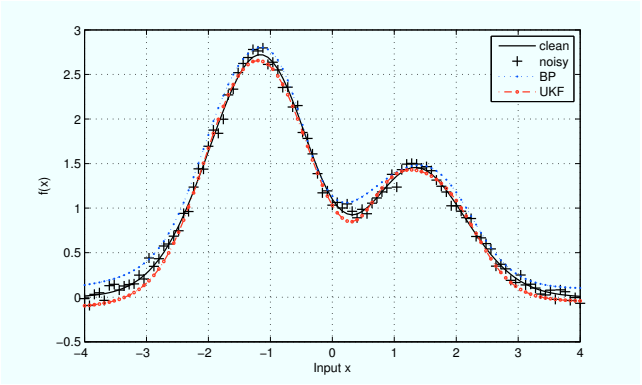

Fig. 2 The Hermit polynomial approximation

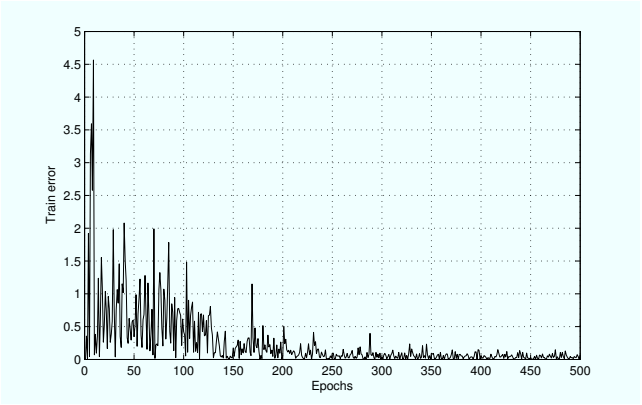

Fig. 3 The training curve of NN based on the UKF

Eq.(10) is a typical nonlinear polynomial, Hermit polynomial [14]. The equation is approximated by a three layers NN. Both the input number and the output neuron number are one. The number of the hidden neurons are five. Sigmoidal function is used as the hidden layer neurons activation function. The output layer neurons activation function is linear.

The solid line in Fig.2 shows the curve of Hermit polynomial. The $'+'$ points in Fig.2 is the training samples corrupted by additive zero mean white Gaussian noise. $'\circ'$ points in Fig.2 is the outputs of the trained network based on the UKF. Whereas,$'\cdot'$ is the outputs of the trained network based on the BP. The approximation performance of the network based on the UKF is more superior than that based on the BP. Fig.3 shows that the error is very low after only 200 epochs in the UKF algorithm. Whereas the error remain very high until 10000 epochs in the BP algorithm. Therefore, the training method based on the UKF is substantially more effective in terms of number of training epochs than standard BP for nonlinear mapping problems.

5.2 CLASSIFICATION

The following example is a typical three classification problem as shown in Fig.5. $'+'$ is the first class(inside the triangle), $'\square'$ is the second class(inside the rectangle), and $'\circ'$ is the third class(others). The input number is two, the output neuron number are three and the number of the hidden neurons are five. Sigmoidal function is used as the hidden layer neurons and the output layer neurons activation function. The target outputs of the three class are $[1, 0, 0]$,

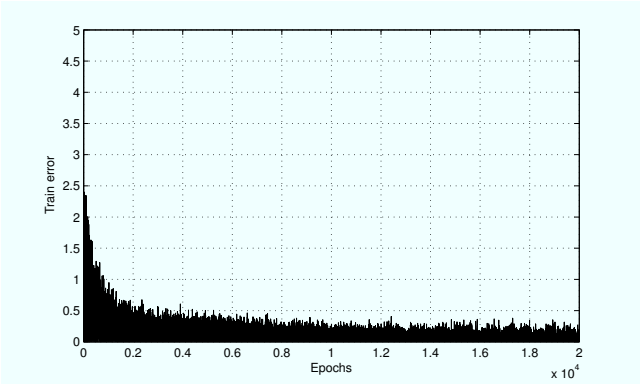

Fig. 4 The training curve of NN based on the BP

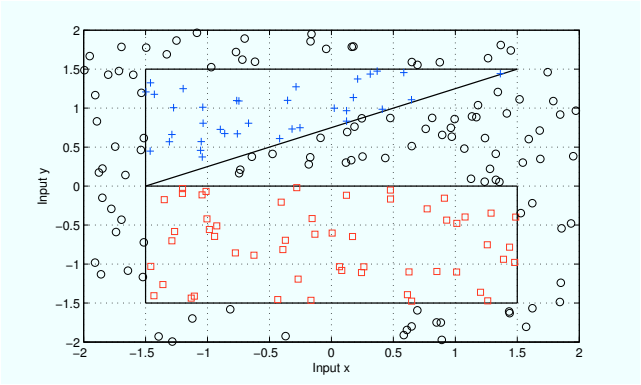

Fig. 5 Three classification problem

$[0,1,0]$, and $[0,0,1]$, respectively. The training samples is 200 random numbers in $(-2,2)\times(-2,2)$. After the NN has been trained, using the UKF, 5000 random test samples in $(-2,2)\times(-2,2)$ is used to test the network. The classification accuracy rate is 96.34%.

6 CONCLUSIONS

The NN based on the UKF is discussed in this paper. The UKF weights update algorithm is compared with the BP. The UKF weights update algorithm has a higher approximation accuracy and convergence velocity than the BP. Higher classification accuracy of the network is also obtained. The weight update algorithm based on the UKF is an incremental update algorithm which is an effective means for escaping from the poor local minima. The numerical simulation results show the effectiveness of the algorithm compared with the standard BP.

REFERENCES

[1] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning representations of back-propagation errors[J]. *Nature*, 1986, 323: 533-536.

[2] R.E. Kalman. A new approach to linear filtering and prediction problems[J]. *Transactions of the ASME, Ser. D, Journal of Basic Engineering*, 1960, 82: 34-45.

[3] S.J. Julier, J.K. Uhlmann. A new extension of the Kalman filter to nonlinear systems[C]// *Proceedings of AeroSense: 11th Int Symposium Aerospace/Defense Sensing,Simulation and Controls*. 1997:54-65.

[4] G.V. Puskorius, L.A. Feldkamp, and L.I. Davis, Jr. Dynamic neural network methods applied to on-vehicle idle speed control[C]// *Proceedings of the IEEE*, 1996,84:1407-1420.

[5] S.J. Julier, J.K. Uhlmann, and H. Durrant-Whyte. A new approach for filtering nonlinear systems[C]// *Proceedings of the American Control Conference*. 1995: 1628-1632.

[6] E.A.Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation[C]// *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC), IEEE,*. 2000.

[7] E.W. Saad, D.V. Prokhorov, and D.C. Wunsch III. Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks[J]. *IEEE Transactions on Neural Networks*, 1998, 9: 1456C1470.

[8] K.-C. Jim, C.L. Giles, and B.G. Horne. An analysis of noise in recurrent neural networks: convergence and generalization[J]. *IEEE Transactions on Neural Networks*, 1996, 7: 1424C1438.

[9] S. Singhal and L. Wu. Training multilayer perceptrons with the extended Kalman filter[C]// *Advances in Neural Information Processing Systems 1*, 1989: 133-140.

[10] S. Haykin. *Kalman Filtering and Neural Networks*[M]. New York: Wiley, Chap. 7, 2002.

[11] A. Sitz, U. Schwarz, J. Kurths, and H. U. Voss. Estimation of parameters and unobserved components for nonlinear systems from noisy time series[J]. *PHYSICAL REVIEW E*, 2002, 66: 016210.

[12] Bin Deng, Jiang Wang,and Yenqiu Che. A combined method to estimate parameters of neuron from a heavily noise-corrupted time series of active potential[J]. *CHAOS*, 2009,19:015105.

[13] Zheng Li, Joseph E. O'Doherty, Timothy L. Hanson, Mikhail A. Lebedev, Craig S. Henriquez, Miguel A. L. Nicolelis. Unscented Kalman Filter for Brain-Machine Interfaces[J]. *PLoS one*, 2009,4(7)e6243: 1-18.

[14] David J. C. MacKay. Bayesian interpolation[J]. *Neural Computation*, 1992, 4(3): 415-447.