Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2007) 2460–2466
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Nonlinear system identification with recurrent neural networks and dead-zone Kalman filter algorithm

José de Jesús Rubio[a], Wen Yu[b,*]

[a]Departamento de Electronica, Sección de Instrumentación, UAM-Azcapotzalco, México D.F., México
[b]Departamento de Control Automatico, CINVESTAV-IPN, Av.IPN 2508, México DF 07360, Mexico

Received 2 August 2006; received in revised form 4 September 2006; accepted 23 September 2006
Communicated by K. Li
Available online 20 October 2006

Abstract

Compared to normal learning algorithms, for example backpropagation, Kalman filter-based algorithm has some better properties, such as faster convergence, although this algorithm is more complex and sensitive to the nature of noises. In this paper, extended Kalman filter is applied to train state-space recurrent neural networks for nonlinear system identification. In order to improve robustness of Kalman filter algorithm dead-zone robust modification is applied to Kalman filter. Lyapunov method is used to prove that the Kalman filter training is stable.

Keywords: Identification; Neural networks; Kalman filter; Stability

1. Introduction

Recent results show that neural network technique seems to be very effective in identifying a broad category of complex nonlinear systems when complete model information cannot be obtained. Neural networks can be classified as feedforward and recurrent ones [7]. Feedforward networks, for example multilayer perceptrons (MLP), are implemented for the approximation of nonlinear functions in the right-hand side of dynamic model equations. The main drawback of these neural networks is that the weights' updating do not utilize information on the local data structure and the function approximation is sensitive to the training data [14]. Since recurrent networks incorporate feedback, they have powerful representation capability and can successfully overcome disadvantages of feedforward networks [11]. Even though backpropagation (BP) has been widely used as a practical training method for neural networks, the limitations are that it may converge very slowly, there exists local minima problem, and the training process is sensitive to measurement noise. The stability of modified BP algorithm is proved in [22].

Gradient-like learning laws are relatively slow. In order to solve this problem, many descendent methods in the identification and filter theory have been proposed to estimate the weights of neural networks. For example, the extended Kalman filter is applied to train neural networks in [1,8,12,18,20,21], they can give solutions of least-square problems. Most of them use static neural networks, sometimes the output layer must be linear and the hidden layer weights are chosen randomly [3]. A faster convergence with the extended Kalman filter is reached, because it has fewer interactions [8]. However, the computational complexity in each interaction is increased, it requires a large amount of memory. Decoupling technique is used to decrease computational burden [15], the decoupled Kalman filter with diagonal matrix P is similar to gradient algorithm [7], but the learning rate is a time-varying matrix.

There are not so many stability analyses for Kalman filter training, in spite of reported successful Kalman filter applications. Guo [6] analyzed convergence and stability properties of the Kalman filter for linear stochastic time-varying regression models. Reif et al. [16] proved that the estimation error of Kalman filter remains bounded if the

*Corresponding author. Tel.: +52 55 50613734; fax: +52 55 57477089.
E-mail address: yuw@ctrl.cinvestav.mx (W. Yu).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.09.004

system satisfies a detectability condition. Reif and Unbehauen [17] stated that the Kalman filter is exponentially stable when the covariance P is bounded and filter error is small enough, these conditions are very hard. Ghaoui [4] presented a new approach to finite-horizon guaranteed state prediction for discrete-time systems affected by bounded noise and unknown-but-bounded parameter uncertainty. There are only a few published results on stability analysis of neural networks training with Kalman filter. Nishiyama and Suzuki [13] used H_∞-learning to improved the robustness of Kalman filter training. By using results on stochastic stability of Kalman filter, Alessandri et al. [1] analyzed the convergence of the weights of neural networks with the assumption of the covariance P_k being bounded. The lack of robustness in Kalman filter with respect to noise was demonstrated in [13]. Several robust modification techniques were proposed for the least-square algorithm [5] which is the special cases of Kalman filter.

In this paper the extended Kalman filter is modified with dead-zone technique, and is applied for state-space recurrent neural networks training. Both hidden layers and output layers can be updated. Stability analysis of identification error with the Kalman filter algorithm is given by the Lyapunov stability technique. A simple simulation gives the effectiveness of the suggested algorithm.

2. Recurrent neural networks training with extended Kalman filter

Consider the following unknown discrete-time nonlinear system:

$$x(k+1) = f[x(k), u(k)], \tag{1}$$

where $u(k) \in \Re^m$ is the input vector, $|u(k)|^2 \leqslant \overline{u}$, $x(k) \in \Re^n$ is a state vector, $u(k)$ and $x(k)$ are known. f is an unknown general nonlinear smooth function $f \in C^\infty$. We use the following state-space recurrent neural network to identify the nonlinear plant (1):

$$\hat{x}(k+1) = A\hat{x}(k) + V_{1,k}\sigma[W_{1,k}x(k)] + V_{2,k}\phi[W_{2,k}x(k)]u(k), \tag{2}$$

where $\hat{x}(k) \in \Re^n$ represents the internal state of the neural network . The matrix $A \in \Re^{n\times n}$ is a stable matrix. The weights in output layer are $V_{1,k}, V_{2,k} \in R^{n\times m}$, the weights in hidden layer are $W_{1,k}, W_{2,k} \in R^{m\times n}$, σ is m-dimension vector function $\sigma = [\sigma_1 \cdots \sigma_m]^{\mathrm{T}}$, $\phi(\cdot)$ is $\Re^{m\times m}$ diagonal matrix,

$$\sigma[W_{1,k}x(k)] = \left[\sigma_1\left(\sum_{j=1}^{n} w_{1,1,j}x_j\right), \sigma_2\left(\sum_{j=1}^{n} w_{1,2,j}x_j\right), \ldots, \sigma_m\left(\sum_{j=1}^{n} w_{1,m,j}x_j\right)\right]^{\mathrm{T}},$$

$$\phi[W_{2,k}x(k)]u(k) = \left[\phi_1\left(\sum_{j=1}^{n} w_{2,1,j}x_j\right)u_1, \phi_2\left(\sum_{j=1}^{n} w_{2,2,j}x_j\right) \times u_2, \ldots, \phi_m\left(\sum_{j=1}^{n} w_{2,m,j}x_j\right)u_m\right]^{\mathrm{T}}, \tag{3}$$

where σ_i and ϕ_i are Sigmoid functions. According to the Stone–Weierstrass theorem and density properties of recurrent neural networks [11], the unknown nonlinear system (1) can be written in the following form:

$$x(k+1) = Ax(k) + V_{1,k}\sigma[W_{1,k}x(k)] + V_{2,k}\phi[W_{2,k}x(k)]u(k) - \eta(k), \tag{4}$$

where $\eta(k) = f[x(k), u(k)] - Ax(k) - V_{1,k}\sigma[W_{1,k}x(k)] - V_{2,k}\phi[W_{2,k}x(k)]u(k)$ is modeling error with respect to the weights $V_{1,k}$, $V_{2,k}$, $W_{1,k}$ and $W_{2,k}$; they are time-varying weights which will be updated by identification error. By [11] we know that the term $\eta(k)$ can be made arbitrarily small by simply selecting appropriate the number of neurons in the hidden layer (in this paper, it is m). In the case of two independent variables, a smooth function f has the following Taylor series expansion:

$$f = \sum_{k=0}^{l-1} \frac{1}{k!}\left[(x_1 - x_1^0)\frac{\partial}{\partial x_1} + (x_2 - x_2^0)\frac{\partial}{\partial x_2}\right]_0^k f + \varepsilon, \tag{5}$$

where ε is the remainder of the Taylor formula. If we let x_1 and x_2 correspond to $W_{1,k}x(k)$ and $V_{1,k}$, x_1^0, x_2^0 correspond to $W_1^0x(k)$ and V_1^0, and define $\widetilde{W}_{1,k} = W_{1,k} - W_1^0$, $\widetilde{V}_{1,k} = V_{1,k} - V_1^0$, then we have

$$V_{1,k}\sigma[W_{1,k}x(k)] = V_1^0\sigma[W_1^0x(k)] + \Theta_{1,k}B_{1,k} + \varepsilon_1, \tag{6}$$

where V_1^0, V_2^0, W_1^0 and W_2^0 are set of known initial constant weights, $B_{1,k} = [\sigma, \sigma' V_{1,k}^{\mathrm{T}}x]^{\mathrm{T}} \in R^{2m\times 1}$, $\Theta_{1,k} = [V_{1,k}, W_{1,k}^{\mathrm{T}}]^{\mathrm{T}} \in R^{n\times 2m}$, σ' is the derivative of nonlinear activation function $\sigma(\cdot)$ with respect to $W_{1,k}x(k)$, the definition of (3), $\sigma' \in \Re^{m\times m}$. Similarly

$$V_{2,k}\phi[W_{2,k}x(k)]u(k) = V_2^0\phi[W_2^0x(k)]u(k) + \Theta_{2,k}B_{2,k} + \varepsilon_2, \tag{7}$$

where $B_{2,k} = [\phi u, \phi' \operatorname{diag}(u)V_{2,k}^{\mathrm{T}}x(k)]^{\mathrm{T}}$, $\Theta_{2,k} = [V_{2,k}, W_{2,k}^{\mathrm{T}}]^{\mathrm{T}}$. We define the modeling error $\zeta(k) = \varepsilon_1 + \varepsilon_2 - \eta(k)$; substituting (6) and (7) into (4) we have

$$y(k) = B_k^{\mathrm{T}}\Theta_k + \zeta(k), \tag{8}$$

where

$$\Theta_k = \begin{bmatrix}\Theta_{1,k}\\ \Theta_{2,k}\end{bmatrix} = [V_{1,k}, W_{1,k}^{\mathrm{T}}, V_{2,k}, W_{2,k}^{\mathrm{T}}]^{\mathrm{T}},$$

$$B_k = \begin{bmatrix}B_{1,k}\\ B_{2,k}\end{bmatrix} = [\sigma, \sigma' V_{1,k}^{\mathrm{T}}x, \phi u, \phi' \operatorname{diag}(u)V_{2,k}^{\mathrm{T}}x(k)]^{\mathrm{T}},$$

the output $y(k)$ is

$$y(k) = x(k+1) - Ax(k) - V_1^0\sigma[W_1^0x(k)] - V_2^0\phi[W_2^0x(k)]u(k).$$