ARTICLE IN PRESS

Neural Networks ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2016 Special Issue

Least square neural network model of the crude oil blending process

José de Jesús Rubio *

Sección de Estudios de Posgrado e Investigación, ESIME Azcapotzalco, Instituto Politécnico Nacional, Av. de las Granjas no. 682, Col. Santa Catarina, México D.F., 02250, Mexico

HIGHLIGHTS

- The recursive least square algorithm is employed for the big data learning of a neural network.
- Some important characteristics of the least square algorithm are analyzed as are the stability and local minimum avoidance.
- The proposed approach is utilized for the modeling of the crude oil blending process.

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Recursive least square
Neural networks
Stability
Big data learning
Fast learning
Crude oil blending

ABSTRACT

In this paper, the recursive least square algorithm is designed for the big data learning of a feedforward neural network. The proposed method as the combination of the recursive least square and feedforward neural network obtains four advantages over the alone algorithms: it requires less number of regressors, it is fast, it has the learning ability, and it is more compact. Stability, convergence, boundedness of parameters, and local minimum avoidance of the proposed technique are guaranteed. The introduced strategy is applied for the modeling of the crude oil blending process.

1. Introduction

In recent years, the recursive least square algorithm has been highly utilized in the big data learning, evolving intelligent systems, and stable intelligent systems.

Big data learning is the learning ability to solve real-world big data problems; some interesting works of this topic are mentioned in Kangin, Angelov, Iglesias, and Sanchis (2015), Kasabov (2014), Luitel and Venayagamoorthy (2014), Mackin, Roy, and Wallenius (2011), Molina, Venayagamoorthy, Liang, and Harley (2013), Roy (2015a), Roy, Mackin, and Mukhopadhyay (2013), Roy (2015b), Schliebs and Kasabov (2013), Xu et al. (2014), and Yuen, King, and Leung (2015). Evolving intelligent systems are learning methods whose structure is flexible to adapt to the environment; some interesting investigations of this issue are detailed in Ballini and Yager (2014), Bouchachia (2014), Bouchachia and Vanaret (2014), Leite, Costa, and Gomide (2013), Lughofer and Sayed-Mouchaweh (2015), Maciel, Gomide, and Ballini (2014), Ordoñez, Iglesias, de Toledo, and Ledezma (2013), Pratama, Anavatti, Er, and Lughofer (2015), Pratama, Anavatti, and Lu (2015), and Toubakh and Sayed-Mouchaweh (2016). The recursive least square algorithm forms a linear combination of regressors which are nonlinear functions of input variables; usually, a large number of regressors must be employed so as to sufficiently cover the space of plant dynamics, it is a problem because the regression matrix would become ill-conditioned due to strong correlation among regressors.

The backpropagation algorithm, also known as the gradient, considered in Luitel and Venayagamoorthy (2014), Molina et al. (2013), Ortega-Zamorano, Jerez, Urda Muñoz, Luque-Baena, and Franco (2015), Xu et al. (2014), and Yuen et al. (2015) is often utilized for the learning of a feedforward neural network, it has the problem of slow convergence because it uses a constant scalar gain as its learning speed.

In this research, the recursive least square algorithm is employed for the big data learning of a feedforward neural network. The combination of the recursive least square algorithm with the feedforward neural network has four advantages as a solution of the two aforementioned problems: (1) the proposed algorithm avoids the regression matrix problem because it only requires the number of regressors utilized by the neural network, (2) the introduced method is faster than the backpropagation because the first uses a time-varying matrix gain as its learning speed, while the second utilizes a constant scalar gain, (3) the proposed combination has the learning ability due to the neural network, (4) the suggested strategy is more compact than the feedforward neural network because the first uses a vector in the hidden layer, while the second utilizes a matrix.

* Tel.: +52 5557296000 64497.
E-mail addresses: jrubioa@ipn.mx, rubio.josedejesus@gmail.com.

http://dx.doi.org/10.1016/j.neunet.2016.02.006
0893-6080/

Moreover, the stable intelligent systems are characterized to be systems where the stability is guaranteed and the parameters are bounded; some interesting works of this theme are included in Ahn (2012), Ahn and Lim (2013), Cheng Lv, Yi, and Li (2015), Li and Rakkiyappan (2013), Lughofer (2011), Orozco-Tupacyupanqui, Nakano-Miyatake, and Perez-Meana (2015), Rakkiyappan, Chandrasekar, Lakshmanan, and Park (2014), Rakkiyappan, Zhu, and Chandrasekar (2014), Rubio, Angelov, and Pacheco (2011), Zhang, Zhu, and Zheng (2015), and Zhang, Zhu, and Zheng (2016). The stability of the recursive least square algorithm should be analyzed to avoid the unboundedness of some parameters known as the overfitting.

The backpropagation with variable learning steps, mentioned in Cheng Lv et al. (2015), Li and Rakkiyappan (2013), Lughofer (2011), Orozco-Tupacyupanqui et al. (2015), and Rubio et al. (2011) also is utilized for the learning of a feedforward neural network; even it is an efficient algorithm, it would be interesting to modify this algorithm to improve its performance by the changing of the time-varying scalar gain as its learning speed.

In this study, the stability of the recursive least square algorithm for the big data learning of a feedforward neural network is analyzed, the proposed stable algorithm has two advantages as a solution of the two above mentioned characteristics: (1) the introduced strategy avoids the overfitting because the stability, convergence, boundedness of parameters, and local minimum avoidance are guaranteed, (2) the suggested algorithm is faster than the backpropagation with variable learning steps because the first uses a time-varying matrix gain as its learning speed, while the second utilizes a time-varying scalar gain.

The paper is organized as follows. In Section 2, the feedforward neural network is presented. In Section 3, the feedforward neural network is linearized. In Section 4, the recursive least square algorithm of a feedforward neural network is designed. In Section 5, the stability, convergence, boundedness parameters, and local minimum avoidance of the before mentioned algorithm are assured. In Section 6, the proposed technique is summarized. In Section 7, the suggested method is applied for the modeling of the crude oil blending process. Section 8 presents conclusions and suggests future research directions.

2. Feedforward neural network

Consider the following unknown discrete-time nonlinear system:

$$y(k) = f\left[x(k)\right] \tag{1}$$

where $x(k) = [x_1(k), \ldots, x_i(k), \ldots, x_N(k)]^T = [y(k-1), \ldots, y(k-n), u(k-1), \ldots, u(k-m)]^T \in \Re^{N\times 1}$ ($N = n+m$) is the input vector, $u(k-1) \in \Re$ is the process input, $y(k) \in \Re$ is the process output, and f is an unknown nonlinear function, $f \in C^{\infty}$. The output of the feedforward neural network with one hidden layer can be expressed as follows:

$$\begin{aligned}
\widehat{y}(k) &= \widehat{v}(k)\phi(k) = \sum_{j=1}^{M} \widehat{v}_j(k)\phi_j(k) \\
\phi(k) &= \left[\phi_1(k), \ldots, \phi_j(k), \ldots, \phi_M(k)\right]^T \\
\phi_j(k) &= \tanh\left(\widehat{w}_j(k)\sum_{i=1}^{N} x_i(k)\right)
\end{aligned} \tag{2}$$

where $i = 1, \ldots, N$, $j = 1, \ldots, M$, $x(k) \in \Re^{N\times 1}$ is the input vector given by (1), $\widehat{y}(k) \in \Re$ is the output of the neural network, $\widehat{v}(k) \in \Re^{1\times M}$ and $\widehat{w}(k) \in \Re^{1\times M}$ are the output and hidden layer weights of the neural network, respectively, $\widehat{w}_j(k) \in \Re$, $x_i(k) \in \Re$, $\phi(k) \in \Re^{M\times 1}$, $\phi_j(k) \in \Re$, $\widehat{v}_j(k) \in \Re$. Fig. 1 shows the architecture of the feedforward neural network where one can see the input, hidden, and output layers.

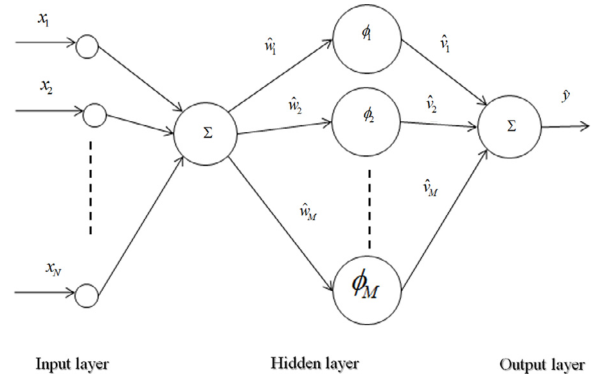

Fig. 1. Architecture of the neural network.

3. Linearization of the neural network

The linearization of the feedforward neural network is required for the recursive least square algorithm design and for the stability analysis.

According to the Stone–Weierstrass theorem, the unknown nonlinear function f of (1) is approximated as follows:

$$\begin{aligned}
y(k) &= v_*\phi_{*k} + \epsilon_f = \sum_{j=1}^{M} v_{j*}\phi_{*j}(k) + \epsilon_f \\
\phi_{*k} &= \left[\phi_{*1}(k), \ldots, \phi_{*j}(k), \ldots, \phi_{*M}(k)\right]^T \\
\phi_{*j}(k) &= \tanh\left(w_{j*}\sum_{i=1}^{N} x_i(k)\right)
\end{aligned} \tag{3}$$

where $\phi_*(k) \in \Re^{M\times 1}$, $\epsilon_f = y(k) - v_*\phi_*(k) \in \Re$ is the modeling error, $\phi_{*j}(k) \in \Re$, $v_{j*} \in \Re$, $w_{j*} \in \Re$, $v_{j*} \in \Re$ and $w_{j*} \in \Re$ are the optimal parameters that can minimize the modeling error ϵ_f. In the case of two independent variables, a function has a Taylor series as follows:

$$\begin{aligned}
f(\omega_1, \omega_2) &= f(\omega_{1^0}, \omega_{2^0}) + \left(\omega_1 - \omega_{1^0}\right)\frac{\partial f(\omega_1, \omega_2)}{\partial \omega_1} \\
&\quad + \left(\omega_2 - \omega_{2^0}\right)\frac{\partial f(\omega_1, \omega_2)}{\partial \omega_2} + r_f
\end{aligned} \tag{4}$$

where $r_f \in \Re$ is the remainder of the Taylor series. ω_1 and ω_2 correspond to $\widehat{w}_j(k) \in \Re$ and $\widehat{v}_j(k) \in \Re$, ω_{1^0} and ω_{2^0} correspond to $w_{j*} \in \Re$ and $v_{j*} \in \Re$, define $\widetilde{w}_j(k) = \widehat{w}_j(k) - w_{j*} \in \Re$ and $\widetilde{v}_j(k) = \widehat{v}_j(k) - v_{j*} \in \Re$; therefore, the Taylor series is applied to linearize (2) as follows:

$$\begin{aligned}
\widehat{v}(k)\phi(k) &= v_*\phi_*(k) + \sum_{j=1}^{M} \widetilde{w}_j(k)\frac{\partial \widehat{v}(k)\phi(k)}{\partial \widehat{w}_j(k)} \\
&\quad + \sum_{j=1}^{M} \widetilde{v}_j(k)\frac{\partial \widehat{v}(k)\phi(k)}{\partial \widehat{v}_j(k)} + r_f
\end{aligned} \tag{5}$$

where $\frac{\partial \widehat{v}(k)\phi(k)}{\partial \widehat{w}_j(k)} \in \Re$ and $\frac{\partial \widehat{v}(k)\phi(k)}{\partial \widehat{v}_j(k)} \in \Re$, please note that $\widehat{v}(k)\phi(k) = \sum_{j=1}^{M} \widehat{v}_j(k)\phi_j(k)$ and $v_*\phi_*(k) = \sum_{j=1}^{M} v_{j*}\phi_{*j}(k)$. As all the parameters are scalars, the Taylor series is fully applicable. Considering (2) and using the chain rule, it gives:

$$\begin{aligned}
\frac{\partial \widehat{v}_k\phi(k)}{\partial \widehat{w}_j(k)} &= \widehat{v}_j(k)\frac{\partial \phi(k)}{\partial \widehat{w}_j(k)} = \widehat{v}_j(k)\frac{\partial \tanh\left(\widehat{w}_j(k)\sum_{i=1}^{N} x_i(k)\right)}{\partial \widehat{w}_j(k)} \\
&= \sigma_j(k)
\end{aligned} \tag{6}$$

Please cite this article in press as: Rubio, J. d. J., Least square neural network model of the crude oil blending process. Neural Networks (2016), http://dx.doi.org/10.1016/j.neunet.2016.02.006