View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Crossref

International Scholarly Research Network
ISRN Applied Mathematics
Volume 2011, Article ID 145801, 12 pages
doi:10.5402/2011/145801

Research Article

Lyapunov Stability Analysis of Gradient Descent-Learning Algorithm in Network Training

Ahmad Banakar

Mechanical Agriculture Department, Tarbiat Modares University, Tehran, P.O. Box 14115-336, Iran

Correspondence should be addressed to Ahmad Banakar, ah_banakar@modares.ac.ir

Received 17 March 2011; Accepted 13 May 2011

Academic Editors: J.-J. Ruckmann and L. Simoni

Copyright © 2011 Ahmad Banakar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Lyapunov stability theorem is applied to guarantee the convergence and stability of the learning algorithm for several networks. Gradient descent learning algorithm and its developed algorithms are one of the most useful learning algorithms in developing the networks. To guarantee the stability and convergence of the learning process, the upper bound of the learning rates should be investigated. Here, the Lyapunov stability theorem was developed and applied to several networks in order to guaranty the stability of the learning algorithm.

1. Introduction

Science has evolved from an attempt to understand and predict the behavior of the universe and the systems within it. Much of this owes to the development of suitable models, which agree with the observations. These models are either in a symbolic form which the humans use or in mathematical form that are found from physical laws. Most systems are causal, which can be categorized as either static, where the output depends on the current inputs, or dynamic, where the output depends on not only the current inputs but also past inputs and outputs. Many systems also possess unobservable inputs, which cannot be measured, but affect the system's output, that is, time series systems. These inputs are known as disturbances and aggravate the modeling process.

To cope with the complexity of dynamic systems, there have been significant developments in the field of artificial neural network during last three decades which have been applied for identification and modeling [1–5]. One major issue that instigates for proposing these different types of networks is to predict the dynamic behavior of many complex systems existing in nature. ANN is a powerful method in approximating a nonlinear system and mapping between input and output data [1]. Recently, wavelet neural networks (WNNs) have been introduced [6–10]. Such types of networks employ wavelets as the activation function in a hidden layer. Because of the ability of the localized analysis of wavelets collectively in their frequency and time domains and the learning ability of ANN, the WNN prompts a superior system model for complex and seismic applications. The majority of the applications of wavelet function are limited to a small dimension [11] although WNN can handle large-dimension problems as well [6]. Due to the dynamic behavior of recurrent network, they are suitable in dealing with the modeling of dynamic systems as compared with the static behavior of feed-forward network [12–19]. It has already been shown that recurrent networks are less sensitive to noise with relatively smaller network size and simpler structure. Their long-term prediction property makes them more powerful in dealing with dynamic systems. Recurrent networks are less sensitive to noise because the recurrent network could recognize and generate periodic waves in spite of the existence of a large amount of noise. This means that the network is able to regenerate the original periodic waves in the process of learning the teachers' signals with noises [2]. For unknown dynamic systems, the recurrent network results in a smaller-sized network as compared with the feed-forward network [12, 20]. For the time-series modeling, it generates a simpler structure [15–23] and gives long-term predictions [22, 24]. The recurrent network for system modeling learns and memorizes information in terms of embedded weights [21].

Different methods have been introduced for learning the parameters onnetwork based of the gradient descent. All learning methods like backpropagation-through-time [16, 17] or real-time recurrent learning algorithm [18] can be applied in order to adjust parameters of the feed-forward or recurrent networks. In [19], the quasi-Newton method was applied to improve the rate of convergence. In [9, 23], using the Lyapunov stability theorem, a mathematical way was introduced for calculating the upper bound of the learning rate for recurrent and feed-forward wavelet neural network based on the network parameters. Here, the Lyapunov stability theorem is developed and applied to several networks, and the learning procedure of the proposed networks is considered.

2. Methodology

2.1. Gradient-Descent Algorithm

The Gradient-descent (GD) learning can be achieved by minimizing the performance index J as follows:

$$J = \frac{1}{2 \cdot P \cdot y_r^2} \cdot \sum_{p=1}^{P} \left(Y(p) - \widehat{Y}(p) \right)^2, \tag{2.1}$$

where $y_r = (\max_{p=1}^{P} Y(p) - \min_{p=1}^{P} Y(p))$, $\widehat{Y}$ is the output of the known network, Y is the actual data, and P is the number of dataset. The reason for using a normalized mean square error is that it provides a universal platform for modeling evaluation irrespective of the application and target value specification while selecting an input to the model.

In the batch-learning scheme employing the P-data set, achange in any parameter is covered by the following equation:

$$\Delta \upsilon(q) = \sum_{p=1}^{P} \Delta_p \upsilon(q), \tag{2.2}$$

and the parametric update equation is

$$\upsilon(q+1) = \upsilon(q) + \frac{\partial J}{\partial \upsilon}. \tag{2.3}$$

2.2. Lyapunov Method in Analysis of Stability

Consider a dynamic system, which satisfies

$$\dot{x} = f(x,t), \qquad x(t_0) = x_0, \quad x \in R. \tag{2.4}$$

The equilibrium point $x^* = 0$ is stable (in the sense of Lyapunov) at $t = t_0$ if for any $\varepsilon > 0$ there exists a $\delta(t_0, \varepsilon) > 0$ such that

$$\|x(t_0)\| < \delta \Longrightarrow \|x(t)\| < \varepsilon, \quad \forall t \geq t_0. \tag{2.5}$$

Lyapunov Stability Theorem

Let $V(x,t)$ be a nonnegative function with the derivative $\dot{V}$ along the trajectories of the system. Then

(i) The origin of the system is locally stable (in the sense of Lyapunov) if $V(x,t)$ is locally positive definite and $-\dot{V}(x,t) \leq 0$ is locally in x and for all t;

(ii) The origin of the system is globally uniformly asymptotically stable if $V(x,t)$ is positive definite and excrescent and $-\dot{V}(x,t)$ is positive definite.

To approve stability analysis of the networks based on GD learning algorithm, we can define discreet function as

$$V(k) = E(k) = \frac{1}{2} \cdot [e(k)]^2. \tag{2.6}$$

Change of Lyapunov function is

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2} \cdot \left[e^2(k+1) - e^2(k)\right]. \tag{2.7}$$

from

$$e(k+1) = e(k) + \Delta e(k) \Longrightarrow e^2(k+1) = e^2(k) + \Delta^2 e(k) + 2 \cdot e(k) \cdot \Delta e(k). \tag{2.8}$$

Then

$$\Delta V(k) = \Delta e(k) \cdot \left[e(k) + \frac{1}{2} \cdot \Delta e(k)\right]. \tag{2.9}$$

Difference of error is

$$\Delta e(k) = e(k+1) - e(k) \approx \left[\frac{\partial e(k)}{\partial \upsilon}\right]^{T} \cdot \Delta\upsilon, \tag{2.10}$$

where υ is the learning parameter and $e(k) = \hat{y}(k) - y(k)$ is error between output of plant and present output of network

$$\Delta\upsilon = -\eta \cdot \frac{\partial J}{\partial \upsilon}. \tag{2.11}$$

By using (2.10) and (2.1) and putting them in (2.3),

$$\begin{aligned}
\Delta V(k) &= \left[\frac{\partial e(k)}{\partial \upsilon}\right]^{T} \cdot \Delta\upsilon \cdot \left\{ e(k) + \frac{1}{2} \cdot \left[\frac{\partial e(k)}{\partial \upsilon}\right]^{T} \cdot \Delta\upsilon \right\}, \\
\Delta V(k) &= \left[\frac{\partial e(k)}{\partial \upsilon}\right]^{T} \cdot \left(-\eta \cdot \frac{\partial E(k)}{\partial \upsilon}\right) \cdot \left\{ e(k) + \frac{1}{2} \cdot \left[\frac{\partial e(k)}{\partial \upsilon}\right]^{T} \cdot \left(-\eta \cdot \frac{\partial E(k)}{\partial \upsilon}\right) \right\}, \\
\Delta V(k) &= \left[\frac{\partial e(k)}{\partial \upsilon}\right]^{T} \cdot (-\eta) \cdot \frac{1}{P \cdot y_r^2} \cdot e(k) \cdot \frac{\partial \hat{y}(k)}{\partial \upsilon} \\
&\quad \cdot \left\{ e(k) + \frac{1}{2} \cdot \left[\frac{\partial e(k)}{\partial \upsilon}\right]^{T} \cdot (-\eta) \cdot \frac{1}{P \cdot y_r^2} \cdot e(k) \cdot \frac{\partial \hat{y}(k)}{\partial \upsilon} \right\}, \\
\Delta V(k) &= e^2(k) \cdot \left\{ -\left[\frac{\partial \hat{y}(k)}{\partial \upsilon}\right]^{T} \cdot \eta \cdot \frac{1}{P \cdot y_r^2} \cdot \frac{\partial \hat{y}(k)}{\partial \upsilon} + \frac{1}{2} \cdot \left[\frac{\partial \hat{y}(k)}{\partial \upsilon}\right]^{T} \right. \\
&\quad \left. \cdot \left[\frac{\partial \hat{y}(k)}{\partial \upsilon}\right]^{T} \cdot \eta^2 \cdot \frac{1}{(P \cdot y_r^2)^2} \cdot \left(\frac{\partial \hat{y}(k)}{\partial \upsilon}\right)^2 \right\} \\
\Delta V(k) &= -e^2(k) \cdot \frac{1}{2} \cdot \frac{\eta}{P \cdot y_r^2} \cdot \left(\frac{\partial \hat{y}(k)}{\partial \upsilon}\right)^2 \cdot \left\{ 2 - \frac{\eta}{P \cdot y_r^2} \cdot \left(\frac{\partial \hat{y}(k)}{\partial \upsilon}\right)^2 \right\},
\end{aligned} \tag{2.12}$$

where $y_r = (\max_{p=1}^{P} y(p) - \min_{p=1}^{P} y(p))$.

Therefore

$$\Delta V(k) = -\lambda \cdot e^2(k), \tag{2.13}$$

where $\lambda = (1/2) \cdot (\eta/(P \cdot y_r^2)) \cdot (\partial\hat{y}(k)/\partial\upsilon)^2 \cdot \{2 - (\eta/(P \cdot y_r^2)) \cdot (\partial\hat{y}(k)/\partial\upsilon)^2\}$.

From the Lyapunov stability theorem, the stability is guaranteed if $V(k)$ is positive and $V(k)$ is negative. From (2.6), $V(k)$ is already positive. The condition of stability depends on $V(k)$ being negative. Therefore, $\lambda > 0$ is considered for all models.

Because $(1/2) \cdot (\eta/(P \cdot y_r^2)) \cdot (\partial \widehat{y}(k)/\partial \upsilon)^2 > 0$, then the convergence condition is limited to

$$2 - \frac{\eta}{P \cdot y_r^2} \cdot \left(\frac{\partial \widehat{y}(k)}{\partial \upsilon}\right)^2 > 0 \Longrightarrow \frac{\eta}{P \cdot y_r^2} \cdot \left(\frac{\partial \widehat{y}(k)}{\partial \upsilon}\right)^2 < 2 \Longrightarrow \eta < \frac{\left(2 \cdot P \cdot y_r^2\right)}{\left(\partial \widehat{y}(k)/\partial \upsilon\right)^2}. \tag{2.14}$$

The maximum learning rate η changes in a fixed range. Since $2 \cdot P \cdot y_r^2$ does not depend on the model, the value of η_{Max} guarantees that the convergence can be found by minimizing the term of $|\partial \widehat{y}(k)/\partial \upsilon_I|$. Therefore,

$$0 < \eta < \eta_{\text{Max}}, \tag{2.15}$$

where $\eta_{\text{Max}} = (2 \cdot P \cdot y_r^2)/\operatorname{Max}(\partial \widehat{y}(k)/\partial \upsilon)^2$.

3. Experimental Results

In this section, the proposed stability analysis is applied for some networks. The selected networks are neurofuzzy (ANFIA) [25, 26], Wavelet neurofuzzy, and recurrent wavelet network.

3.1. *Example 1: Convergence Theorems of the TSK Neurofuzzy Model*

TSKmodel has a linear or nonlinear relationship of inputs $w^m(X)$ in the output space. The rules of TSK model are in the following way:

$$R^m : \text{if } \mathbf{x} \text{ is } \mathbf{A}^m \text{ then } y \text{ is } w^m(X). \tag{3.1}$$

A linear form of $w^m(X)$ in (3.1) is as follows:

$$w^m(X) = w_0^m + w_1^m x_1 + \cdots + w_n^m x_n. \tag{3.2}$$

By taking the Gaussian membership function and an equal number of fuzzy sets to the rules with respect to the inputs, the firing strength of rules (3.1) can be written as

$$\mu_{A^m}(\mathbf{x}) = \prod_{i=1}^{n} \exp\left(-\left(\frac{x_i - \overline{x}_{mi}}{\sigma_{mi}}\right)^2\right), \tag{3.3}$$

where $\overline{x}_{mi}$ and σ_{mi} are the center and standard deviation of the Gaussian membership functions, respectively. By applying the T-norm (product operator) of the membership functions of the premise parts of the rule and the weighted average gravity method for defuzzification, the output of the TSK model can be defined as

$$\widehat{Y} = \frac{\sum_{m=1}^{M} \mu_{A^m}(\mathbf{x}) \cdot w^m(\mathbf{x})}{\sum_{m=1}^{M} \mu_{A^m}(\mathbf{x})}. \tag{3.4}$$

Theorem 3.1. *The asymptotic learning convergence of TSK neurofuzzy is guaranteed if the learning rate for different learning parameters follows the upper bound as will be mentioned below:*

$$\begin{gathered} 0 < \eta_w < 2 \cdot P \cdot y_r^2, \\ 0 < \eta_\sigma < \frac{2 \cdot P \cdot y_r^2}{\max_m |w(\boldsymbol{X})|^2 \cdot \left(2/\sigma_{\min}^3\right)^2}, \\ 0 < \eta_{\overline{x}} < \frac{2 \cdot P \cdot y_r^2}{\max_m |w(\boldsymbol{X})|^2 \cdot \left(2/\sigma_{\min}^2\right)^2}. \end{gathered} \tag{3.5}$$

Proof. In equation (2.15) for neurofuzzy models can be written as

$$0 < \eta_v < \frac{2 \cdot P \cdot y_r^2}{\left|\partial \widehat{Y}_{\mathrm{NF}} / \partial v\right|_{\max}^2}. \tag{3.6}$$

Because $\beta_m = \mu_{A^m}(\mathbf{X}) / \sum_{m=1}^{M} \mu_{A^m}(\mathbf{X}) \leq 1$ for all m and since local models have same variables, that is, $\mathbf{X}$, therefore, from (3.7), (3.5) easily can be derived

$$\begin{gathered} \frac{\partial \widehat{Y}_{\mathrm{NF}}}{\partial w_{m0}} = \beta_m, \\ \frac{\partial \widehat{Y}_{\mathrm{NF}}}{\partial w_{mi}} = x_i \cdot \beta_m, \\ \frac{\partial \widehat{Y}_{\mathrm{NF}}}{\partial \overline{x}_{mi}} = w_m(\mathbf{X}') \cdot \frac{\beta_m}{\mu_{A^m}} \cdot (1 - \beta_m) \cdot \frac{2 \cdot (x_i - \overline{x}_{mi})}{\sigma_{mi}^2}, \\ \frac{\partial \widehat{Y}_{\mathrm{NF}}}{\partial \sigma_{mi}} = w_m(\mathbf{X}') \cdot \frac{\beta_m}{\mu_{A^m}} \cdot (1 - \beta_m) \cdot \frac{2 \cdot (x_i - \overline{x}_{mi})^2}{\sigma_{mi}^3}. \end{gathered} \tag{3.7}$$

□

3.2. Example 2: Convergence Theorems of Recurrent Wavelet Neuron Models

Each neuron model in the proposed recurrent neuron models is summation or multiplication of Sigmoid Activation Function (SAF) and Wavelet Activation Function (WAF) as shown in Figure 1. Morlet wavelet function is considered in the recurrent models. In the series of developing different recurrent networks and neuron models, the proposed neurons' model is used in a one-hidden-layer feed-forward neural network as shown in Figure 2.

The output of feed-forward network is given in the following equation:

$$\widehat{Y}_{\mathrm{WNN}} = \sum_{l=1}^{L} W_l \cdot y_l, \tag{3.8}$$

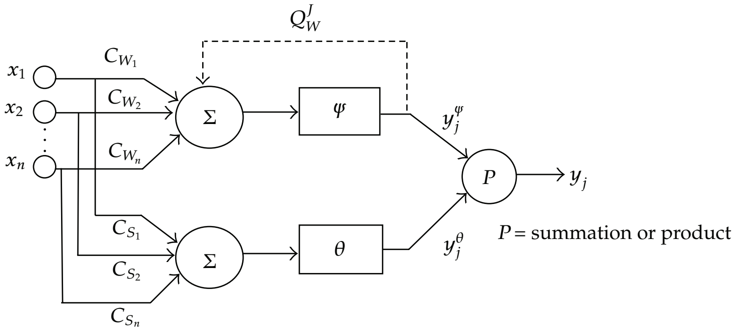


Figure 1: Summation/product recurrent sigmoid-wavelet neuron model.

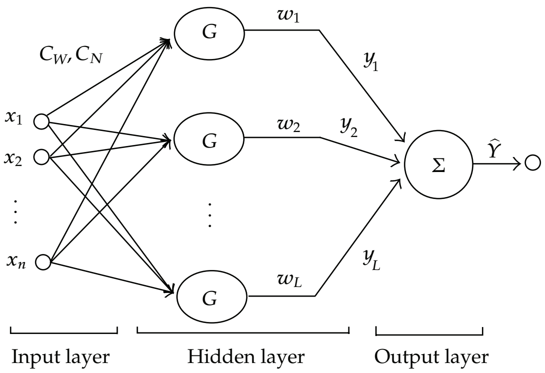


Figure 2: Feed-forward neural network.

where y_l is the output of S-W neurons, W_l is the weights between hidden neuron and output neurons, and L is the number of hidden neuron,

$$y_j(k) = y_j^{\theta}(k) + y_j^{\psi}(k). \tag{3.9}$$

The functions y_j^{θ} and y_j^{ψ} are output of SAF and WAF for jth S-W neuron, in the hidden layer, respectively. The functions y_j^{θ} and y_j^{ψ} are expressed as follow.

$$\begin{gathered} y_j^{\theta}(k) = \theta\left(\sum_{i=1}^{n} C_{S_i}^{j} \cdot x_i(k)\right), \\ y_j^{\psi}(k) = \psi\left(\sum_{i=1}^{n} C_{W_i}^{j} \cdot x_i(k)\right). \end{gathered} \tag{3.10}$$

x_i is ith input. C_S and C_W are weights to input signal for SAF and WAF, in each hidden neuron, respectively.

To prove convergence of the recurrent networks, these facts are needed:

Fact 1: let $g(y) = ye^{(-y^2)}$. Then $|g(y)| < 1,$ for all $y \in \Re$.

Fact 2: let $f(y) = y^2 e^{(-y^2)}$. Then $|f(y)| < 1,$ for all $y \in \Re$.

Fact 3: let $\theta(y) = 1/(1+e^{-y})$be a sigmoid function. Then $|\theta(y)| < 1,$ for all $y \in \Re$

Fact 4: let $\psi_{a,b}(y) = e^{-((y-b)/a)^2} \cos(5((y-b)/a))$ be a Morlet wavelet function. Then $|\psi_{a,b}(y)| < 1,$ for all $y, a, b \in \Re$.

(a) Summation Sigmoid-Recurrent Wavelet

Suppose $Z = \sum_{i=1}^{n} C_{S_i}^{j} \cdot x_i(k)$ and $S = \sum_{i=1}^{n} C_{W_i}^{j} \cdot x_i(k) + Q_W^j \cdot y_\psi^j(k-1)$.

From the facts 3 and 4: For parameter W in all models

$$\frac{\partial \hat{y}}{\partial W_j} = y_j < \left| y_\psi^j + y_\theta^j \right| < 1 + 1 = 2. \tag{3.11}$$

Therefore $0 < \eta_W < (2 \cdot P \cdot y_r^2)/2^2 = (P \cdot y_r^2)/2$.

Differential of output of the model for another learning parameter is

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial C_{W_i}^{j}} &= x_i(k) \cdot W^j \cdot \psi' \left(\sum_{i=1}^{n} C_{W_i}^{j} \cdot x_i(k) + Q_W^j \cdot y_\psi^j(k-1) \right) \\ &< 1 \cdot 1 \cdot \left| \frac{-2}{a} \cdot \frac{S-b}{a} \cdot e^{-((S-b)/a)^2} \cdot \cos\left(5 \frac{S-b}{a}\right) - e^{-((S-b)/a)^2} \cdot \frac{5}{a} \cdot \sin\left(5 \frac{S-b}{a}\right) \right| \\ &< \left\{ \frac{2}{a_{\min}} \cdot 1 \cdot 1 + \frac{5}{a_{\min}} \cdot 1 \right\} < 7. \end{aligned} \tag{3.12}$$

Therefore, $0 < \eta_{C_W} < (2 \cdot P \cdot y_r^2)/7^2 = (2 \cdot P \cdot y_r^2)/49$

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial C_{S_i}^{j}} &= x_i(k) \cdot W^i \cdot \theta' \left(\sum_{i=1}^{n} C_{S_i}^{j} \cdot x_i(k) \right) \\ &< 1 \cdot 1 \cdot \theta(z) \cdot (1 - \theta(z)) < 1 \cdot 1 = 1. \end{aligned} \tag{3.13}$$

Therefore $0 < \eta_{C_S} < (2 \cdot P \cdot y_r^2)/1^2 = 2 \cdot P \cdot y_r^2$

$$\begin{aligned} \frac{\partial \hat{y}(k)}{\partial Q_W^j} &= W^j \cdot y_\psi^j(k-1) \cdot \psi' \left(\sum_{i=1}^{n} C_{W_i}^{j} \cdot x_i(k) + Q_W^j \cdot y_\psi^j(k-1) \right) \\ &< 1 \cdot 1 \cdot \left| \frac{-2}{a} \cdot \frac{S-b}{a} \cdot e^{-((S-b)/a)^2} \cdot \cos\left(5 \frac{S-b}{a}\right) - e^{-((S-b)/a)^2} \cdot \frac{5}{a} \cdot \sin\left(5 \frac{S-b}{a}\right) \right| \\ &< \left\{ \frac{2}{a_{\min}} \cdot 1 \cdot 1 + \frac{5}{a_{\min}} \cdot 1 \right\} < 7. \end{aligned} \tag{3.14}$$

Therefore, $0 < \eta_{Q_W} < (2 \cdot P \cdot y_r^2)/7^2 = (2 \cdot P \cdot y_r^2)/49$.

(b) Multiplication Sigmoid-Recurrent Wavelet

From facts 3 and 4 suppose $Z = \sum_{i=1}^{n} C_{S_i}^{j} \cdot x_i(k)$ and $S = \sum_{i=1}^{n} C_{W_i}^{j} \cdot x_i(k) + Q_W^j \cdot y_\varphi^j(k-1)$.

For parameter W in all networks:

$$\frac{\partial \hat{y}}{\partial W^j} = y_j = y_\varphi^j \cdot y_\theta^j < 1 \cdot 1 < 1. \tag{3.15}$$

Therefore, $0 < \eta_W < (2 \cdot P \cdot y_r^2)/1 < 2 \cdot P \cdot y_r^2$

$$\begin{aligned}\frac{\partial \hat{y}(k)}{\partial C_{W_i}^j} &= x_i(k) \cdot W^j \cdot \theta\left(\sum_{i=1}^{n} C_{S_i}^j \cdot x_i(k)\right) \cdot \varphi'\left(\sum_{i=1}^{n} C_{W_i}^j \cdot x_i(k) + Q_W^j \cdot y_\varphi^j(k-1)\right)\\ &< 1 \cdot 1 \cdot 1 \cdot \left|\frac{-2}{a} \cdot \frac{S-b}{a} \cdot e^{-((S-b)/a)^2} \cdot \cos\left(5\frac{S-b}{a}\right) - e^{-((S-b)/a)^2} \cdot \frac{5}{a} \cdot \sin\left(5\frac{S-b}{a}\right)\right|\\ &< \left\{\frac{2}{a_{\min}} \cdot 1 \cdot 1 + \frac{5}{a_{\min}} \cdot 1\right\} < 7.\end{aligned} \tag{3.16}$$

Therefore, $0 < \eta_{C_W} < (2 \cdot P \cdot y_r^2)/(7)^2 = (2 \cdot P \cdot y_r^2)/49$

$$\begin{aligned}\frac{\partial \hat{y}(k)}{\partial C_{S_i}^j} &= x_i(k) \cdot W^j \cdot \theta'\left(\sum_{i=1}^{n} C_{S_i}^j \cdot x_i(k)\right) \cdot \varphi\left(\sum_{i=1}^{n} C_{W_i}^j \cdot x_i(k) + Q_W^j \cdot y_\varphi^j(k-1)\right)\\ &< 1 \cdot 1 \cdot \theta(Z) \cdot (1 - \theta(Z)) \cdot 1 < 1 \cdot 1 < 1.\end{aligned} \tag{3.17}$$

Therefore, $0 < \eta_{C_S} < (2 \cdot P \cdot y_r^2)/(1)^2 = 2 \cdot P \cdot y_r^2$

$$\begin{aligned}\frac{\partial \hat{y}(k)}{\partial Q_W^j} &= W^j \cdot y_\varphi^j(k-1) \cdot \theta\left(\sum_{i=1}^{n} C_{S_i}^j \cdot x_i(k)\right) \cdot \varphi'\left(\sum_{i=1}^{n} C_{W_i}^j \cdot x_i(k) + Q_W^j \cdot y_\varphi^j(k-1)\right)\\ &< 1 \cdot 1 \cdot 1 \cdot \left|\frac{-2}{a} \cdot \frac{S-b}{a} \cdot e^{-((S-b)/a)^2} \cdot \cos\left(5\frac{S-b}{a}\right) - e^{-((S-b)/a)^2} \cdot \frac{5}{a} \cdot \sin\left(5\frac{S-b}{a}\right)\right|\\ &< \left\{\frac{2}{a_{\min}} \cdot 1 \cdot 1 + \frac{5}{a_{\min}} \cdot 1\right\} < 7.\end{aligned} \tag{3.18}$$

Therefore, $0 < \eta_{Q_W} < (2 \cdot P \cdot y_r^2)/(7)^2 = (2 \cdot P \cdot y_r^2)/49$

3.3. Example 3: Convergence Theorems of the Wavelet Nuro-Fuzzy (WNF) Model

The consequent part of each fuzzy rule corresponds to a sub-WNN consisting of wavelet with the specified dilation value, where, in the TSK fuzzy model, a linear function of inputs is used while $w^m(X) = \widehat{Y}_{\mathrm{WNN}_m}$. Figure 1 shows the proposed WNN model which uses a combination of sigmoid and wavelet activation functions as a hidden neuron (Figure 2 without recurrent part) in the consequent part of each fuzzy rule.

Theorem 3.2. *The asymptotic learning convergence is guaranteed if the learning rate for different learning parameters follows the upper bound as will be mentioned below:*

$$\begin{gathered}
0 < \eta_{\sigma} < \frac{2 \cdot P \cdot y_r^2}{\left|\widehat{Y}_{\mathrm{WNN}}\right|_{\max}^2 \cdot \left(2/\sigma_{\min}^3\right)^2}, \\
0 < \eta_{\overline{x}} < \frac{2 \cdot P \cdot y_r^2}{\left|\widehat{Y}_{\mathrm{WNN}}\right|_{\max}^2 \cdot \left(2/\sigma_{\min}^2\right)^2}, \\
0 < \eta_{w} < \frac{2 \cdot P \cdot y_r^2}{\left|\partial \widehat{Y}_{\mathrm{WNN}}/\partial w\right|_{\max}^2}, \\
0 < \eta_{C_S} < \frac{2 \cdot P \cdot y_r^2}{\left|\partial \widehat{Y}_{\mathrm{WNN}}/\partial C_S\right|_{\max}^2}, \\
0 < \eta_{C_W} < \frac{2 \cdot P \cdot y_r^2}{\left|\partial \widehat{Y}_{\mathrm{WNN}}/\partial C_W\right|_{\max}^2},
\end{gathered} \tag{3.19}$$

where η_w, η_{C_N}, or η_{C_W} and η_σ or $\eta_{\overline{x}}$ are the parameters' learning rates of the consequent and the premise parts of the fuzzy rules. C_S and C_W are weights to inputs, signal for sigmoid and wavelet activation functions of local WNNs, in each hidden neuron, respectively. $\overline{x}_m$ and σ_m are the center and standard deviation of the Gaussian membership functions of rule number m in WNF model, respectively.

Proof. In equation (2.15) for WNF models can be written as

$$\begin{gathered}
0 < \eta_{v} < \frac{2 \cdot P \cdot y_r^2}{\left|\partial \widehat{Y}_{\mathrm{WNF}}/\partial v\right|_{\max}^2}, \\
\frac{\partial \widehat{Y}_{\mathrm{WNF}}}{\partial w} = \beta_m \cdot \frac{\partial \widehat{Y}_{\mathrm{WNN}_m}}{\partial w}, \\
\frac{\partial \widehat{Y}_{\mathrm{WNF}}}{\partial C_N} = \beta_m \cdot \frac{\partial \widehat{Y}_{\mathrm{WNN}_m}}{\partial C_N}, \\
\frac{\partial \widehat{Y}_{\mathrm{WNF}}}{\partial C_W} = \beta_m \cdot \frac{\partial \widehat{Y}_{\mathrm{WNN}_m}}{\partial C_W}.
\end{gathered} \tag{3.20}$$

Because $\beta_m = \mu_{A^m}(\mathbf{X}) / \sum_{m=1}^{M} \mu_{A^m}(\mathbf{X}) \leq 1$ for all m, therefore (3.13) to (3.15) are easily derived.

From (2.15) and (3.4) for parameters σ or $\overline{x}$,there is

$$\begin{aligned}
\frac{\partial \widehat{Y}_{\mathrm{WNF}}}{\partial \sigma} &= \widehat{Y}_{\mathrm{WNN}_m} \cdot \frac{\beta_m}{\mu_{A^m}} \cdot \left(1-\beta_m\right) \cdot \frac{2\cdot\left(x_i-\overline{x}_{mi}\right)^2}{\sigma_{mi}^3} \\
&= \widehat{Y}_{\mathrm{WNN}_m} \cdot \frac{\left(1-\beta_m\right)}{\sum_{m=1}^{M}\mu_{A^m}} \cdot \frac{2\cdot\left(x_i-\overline{x}_{mi}\right)^2}{\sigma_{mi}^3}, \\
\frac{\partial \widehat{Y}_{\mathrm{WNF}}}{\partial \overline{x}} &= \widehat{Y}_{\mathrm{WNN}_m} \cdot \frac{\beta_m}{\mu_{A^m}} \cdot \left(1-\beta_m\right) \cdot \frac{2\cdot\left(x_i-\overline{x}_{mi}\right)}{\sigma_{mi}^2} \\
&= \widehat{Y}_{\mathrm{WNN}_m} \cdot \frac{\left(1-\beta_m\right)}{\sum_{m=1}^{M}\mu_{A^m}} \cdot \frac{2\cdot\left(x_i-\overline{x}_{mi}\right)}{\sigma_{mi}^2}
\end{aligned} \tag{3.21}$$

and therefore (3.19) arederived. □

4. Conclusion

In this paper, a developed Lyapunov stability theorem was applied to guarantee the convergence of the gradient-descent learning algorithm in network training. The experimental examples showed that the upper bound of the learning parameter could be easily considered using this theorem. So, an adaptive learning algorithm can guaranty the fast and stable learning procedure.

References

[1] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.

[2] S. Z. Qin, H. T. Su, and T. J. McAvoy, "Comparison of four neural net learning methods for dynamic system identification," *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 122–130, 1992.

[3] T. Yabuta and T. Yamada, "Learning control using neural networks," in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA '91)*, pp. 740–745, Sacramento, Calif, USA, April 1991.

[4] P. Frasconi, M. Gori, and G. Soda, "Local feedback multilayered networks," *Neural Computation*, vol. 7, no. 1, pp. 120–130, 1992.

[5] J. C. Patra, R. N. Pal, B. N. Chatterji, and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 29, no. 2, pp. 254–262, 1999.

[6] Q. Zhang and A. Benveniste, "Wavelet networks," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 889–898, 1992.

[7] J. Zhang, G. G. Walter, Y. Miao, and W. Lee, "Wavelet neural networks for function learning," *IEEE Transactions on Signal Processing*, vol. 43, no. 6, pp. 1485–1497, 1995.

[8] T. I. Boubez and R. L. Peskin, "Wavelet neural networks and receptive field partitioning," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1544–1549, San Francisco, Calif, USA, March 1993.

[9] A. Banakar and M. F. Azeem, "Artificial wavelet neural network and its application in neuro-fuzzy models," *Applied Soft Computing*, vol. 8, no. 4, pp. 1463–1485, 2008.

[10] Q. Zhang, "Using wavelet network in nonparametric estimation," *IEEE Transactions on Neural Networks*, vol. 8, no. 2, pp. 227–236, 1997.

[11] A. Benveniste, B. Juditsky, B. Delyon, Q. Zhang, and P. Y. Glorennec, "Wavelets in identification," in *Proceedings of the 10th IFAC Symposium on System Identification, (SYSID '94)*, Copenhagen, Denmark, July 1994.

[12] X. D. Li, J. K. L. Ho, and T. W. S. Chow, "Approximation of dynamical time-variant systems by continuous-time recurrent neural networks," *IEEE Transactions on Circuits and Systems*, vol. 52, no. 10, pp. 656–660, 2005.

[13] B. Srinivasan, U. R. Prasad, and N. J. Rao, "Back propagation through adjoints for the identification of nonlinear dynamic systems using recurrent neural models," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 213–228, 1994.

[14] P. Frasconi and M. Gori, "Computational capabilities of local-feedback recurrent networks acting as finite-state machines," *IEEE Transactions on Neural Networks*, vol. 7, no. 6, pp. 1520–1525, 1996.

[15] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, pp. 79–87, 1991.

[16] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing I*, D. E. Rumelhart and J. L. McClelland, Eds., pp. 675–695, MIT Press, Cambridge, UK, 1986.

[17] P. Werbos, "Generalization of backpropagation with application to a recurrent gas Markov model," *Neural Networks*, vol. 1, pp. 339–356, 1988.

[18] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent networks," *Neural Networks*, vol. 1, pp. 270–280, 1989.

[19] R. J. Williams and D. Zipser, "Mechanical system modeling using recurrent neural networks via quasi-Newton learning methods," *Applied Mathematical Modeling*, vol. 19, no. 7, pp. 421–428, 1995.

[20] C.-F. Juang, "A TSK-type recurrent fuzzy network for dynamic systems processing by neural network and genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 155–170, 2002.

[21] C.-H. Lee and C.-C. Teng, "Identification and control of dynamic systems using recurrent fuzzy neural networks," *IEEE Transactions on Fuzzy Systems*, vol. 8, no. 4, pp. 349–366, 2000.

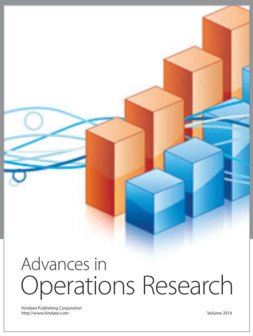
[22] P. A. Mastorocostas and J. B. Theocharis, "A recurrent fuzzy-neural model for dynamic system identification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, no. 2, pp. 176–190, 2002.

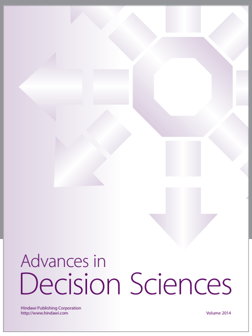
[23] S. J. Yoo, Y. H. Choi, and J. B. Park, "Generalized predictive control based on self-recurrent wavelet neural network for stable path tracking of mobile robots: adaptive learning rates approach," *IEEE Transactions on Circuits and Systems*, vol. 53, no. 6, pp. 1381–1394, 2006.

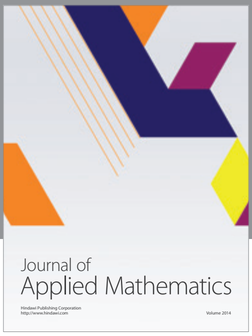
[24] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis, and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Transactions on Energy Conversion*, vol. 21, no. 1, pp. 273–284, 2006.

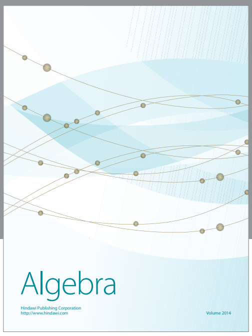
[25] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.

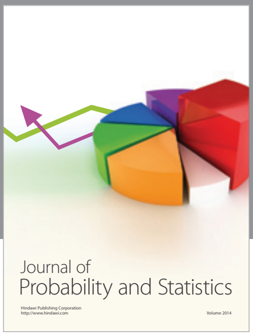
[26] J. S. R. Jang, "ANFIS: adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993.











Hindawi

Submit your manuscripts at
http://www.hindawi.com