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 on network 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:

\[
J = \frac{1}{2 \cdot P \cdot y_r^2} \cdot \sum_{p=1}^{P} \left( Y(p) - \hat{Y}(p) \right)^2,
\]

(2.1)

where \( y_r = (\max_{p=1}^{P} Y(p) - \min_{p=1}^{P} Y(p)) \), \( \hat{Y} \) 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, a change in any parameter is
covered by the following equation:

\[
\Delta u(q) = \sum_{p=1}^{P} \Delta_p u(q),
\]

(2.2)
The parametric update equation is

\[ \nu(q + 1) = \nu(q) + \frac{\partial f}{\partial \nu}. \quad (2.3) \]

### 2.2. Lyapunov Method in Analysis of Stability

Consider a dynamic system, which satisfies

\[ \dot{x} = f(x, t), \quad x(t_0) = x_0, \quad x \in \mathbb{R}. \quad (2.4) \]

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

\[ \|x(t_0)\| < \delta \implies \|x(t)\| < \varepsilon, \quad \forall t \geq t_0. \quad (2.5) \]

**Lyapunov Stability Theorem**

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

1. 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 \( t \);
2. 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

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

Change of Lyapunov function is

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

from

\[ e(k + 1) = e(k) + \Delta e(k) \implies e^2(k + 1) = e^2(k) + \Delta^2 e(k) + 2 \cdot e(k) \cdot \Delta e(k). \quad (2.8) \]

Then

\[ \Delta V(k) = \Delta e(k) \cdot \left[e(k) + \frac{1}{2} \cdot \Delta e(k)\right]. \quad (2.9) \]
Difference of error is

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

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

$$\Delta v = -\eta \cdot \frac{\partial J}{\partial v}.$$  

(2.11)

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

$$\Delta V(k) = \left[ \frac{\partial e(k)}{\partial v} \right]^T \cdot \Delta v \cdot \left\{ e(k) + \frac{1}{2} \cdot \left[ \frac{\partial e(k)}{\partial v} \right]^T \cdot \Delta v \right\},$$

$$\Delta V(k) = \left[ \frac{\partial e(k)}{\partial v} \right]^T \cdot (-\eta) \cdot \frac{1}{P \cdot y_r^2} \cdot e(k) \cdot \frac{\partial \tilde{y}(k)}{\partial v}$$

$$\cdot \left\{ e(k) + \frac{1}{2} \cdot \left[ \frac{\partial e(k)}{\partial v} \right]^T \cdot (-\eta) \cdot \frac{1}{P \cdot y_r^2} \cdot e(k) \cdot \frac{\partial \tilde{y}(k)}{\partial v} \right\},$$

$$\Delta V(k) = e^2(k) \cdot \left\{ -\left[ \frac{\partial \tilde{y}(k)}{\partial v} \right]^T \cdot \eta \cdot \frac{1}{P \cdot y_r^2} \cdot \frac{\partial \tilde{y}(k)}{\partial v} \right. + \frac{1}{2} \cdot \left[ \frac{\partial \tilde{y}(k)}{\partial v} \right]^T \left[ \frac{\partial \tilde{y}(k)}{\partial v} \right]$$

$$\cdot \eta^2 \cdot \frac{1}{(P \cdot y_r^2)^2} \cdot \left( \frac{\partial \tilde{y}(k)}{\partial v} \right)^2 \right\}$$

$$\Delta V(k) = -e^2(k) \cdot \frac{1}{2} \cdot \frac{\eta}{P \cdot y_r^2} \cdot \left( \frac{\partial \tilde{y}(k)}{\partial v} \right)^2 \cdot \left\{ 2 - \frac{\eta}{P \cdot y_r^2} \cdot \left( \frac{\partial \tilde{y}(k)}{\partial v} \right)^2 \right\},$$

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

Therefore

$$\Delta V(k) = -\lambda \cdot e^2(k),$$

(2.13)

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

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 \( (1/2) \cdot \left( \eta / (P \cdot y_r^2) \right) \cdot (\partial \hat{y}(k) / \partial v)^2 > 0 \), then the convergence condition is limited to

\[
2 - \frac{\eta}{P \cdot y_r^2} \cdot \left( \frac{\partial \hat{y}(k)}{\partial v} \right)^2 > 0 \implies \frac{\eta}{P \cdot y_r^2} \cdot \left( \frac{\partial \hat{y}(k)}{\partial v} \right)^2 < 2 \implies \eta < \frac{(2 \cdot P \cdot y_r^2)}{(\partial \hat{y}(k) / \partial v)^2}.
\] (2.14)

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

\[
0 < \eta < \eta_{\text{Max}},
\] (2.15)

where \( \eta_{\text{Max}} = (2 \cdot P \cdot y_r^2) / \text{Max} (\partial \hat{y}(k) / \partial v)^2 \).

**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

TSK model 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:

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

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

\[
w^m(X) = w_0^m + w_1^m x_1 + \cdots + w_n^m x_n.
\] (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

\[
\mu_{A^m}(x) = \prod_{i=1}^{n} \exp \left( -\left( \frac{x_i - \bar{x}_{mi}}{\sigma_{mi}} \right)^2 \right),
\] (3.3)

where \( \bar{x}_{mi} \) and \( \sigma_{mi} \) 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

\[
\hat{y} = \frac{\sum_{m=1}^{M} \mu_{A^m}(x) \cdot w^m(x)}{\sum_{m=1}^{M} \mu_{A^m}(x)}.
\] (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{align*}
0 < \eta_w &< 2 \cdot P \cdot y_r^2, \\
0 < \eta_\sigma &< \frac{2 \cdot P \cdot y_r^2}{\max_m |w(X)|^2 \cdot \left(\frac{2}{\sigma_{\min}^3}\right)^2}, \\
0 < \eta_x &< \frac{2 \cdot P \cdot y_r^2}{\max_m |w(X)|^2 \cdot \left(\frac{2}{\sigma_{\min}^2}\right)^2}.
\end{align*}

(3.5)

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

\begin{equation}
0 < \eta_\nu < \frac{2 \cdot P \cdot y_r^2}{\left|\frac{\partial \hat{Y}_{NF}}{\partial \nu}\right|_{\max}}.
\end{equation}

(3.6)

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

\begin{align*}
\frac{\partial \hat{Y}_{NF}}{\partial w_{m0}} &= \beta_m, \\
\frac{\partial \hat{Y}_{NF}}{\partial w_{mi}} &= \beta_m \cdot x_i, \\
\frac{\partial \hat{Y}_{NF}}{\partial \sigma_{mi}} &= \beta_m \cdot \frac{2 \cdot (x_i - \bar{x}_{mi})}{\sigma_{mi}^3}.
\end{align*}

(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:

\[\hat{Y}_{WNN} = \sum_{l=1}^{L} W_l \cdot y_l,\]

(3.8)
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,

$$y_j(k) = y_j^\theta(k) + y_j^\psi(k).$$  \hfill (3.9)

The functions $y_j^\theta$ and $y_j^\psi$ are output of SAF and WAF for $j$th S-W neuron, in the hidden layer, respectively. The functions $y_j^\theta$ and $y_j^\psi$ are expressed as follow.

$$y_j^\theta(k) = \theta\left(\sum_{i=1}^{n} C_j S_i \cdot x_i(k)\right),$$

$$y_j^\psi(k) = \psi\left(\sum_{i=1}^{n} C_j W_i \cdot x_i(k)\right).$$  \hfill (3.10)

$x_i$ is $i$th 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 \( y \in \mathfrak{R} \).

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

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

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

\( (a) \) Summation Sigmoid-Recurrent Wavelet

Suppose \( Z = \sum_{i=1}^{n} C^j_i \cdot x_i(k) \) and \( S = \sum_{i=1}^{n} C^j_{Wi} \cdot x_i(k) + Q^j_W \cdot y^j_W(k-1) \).

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

\[
\frac{\partial \tilde{y}}{\partial C^j_{Wi}} = y_i < \left| y^j_W + y^j_{c} \right| < 1 + 1 = 2. \tag{3.11}
\]

Therefore \( 0 < \eta_{\mathcal{W}} < (2 \cdot P \cdot y^2_r)/2^2 = (P \cdot y^2_r)/2 \).

Differential of output of the model for another learning parameter is

\[
\frac{\partial \tilde{y}(k)}{\partial C^j_i} = x_i(k) \cdot W^j_i \cdot \psi'(\sum_{i=1}^{n} C^j_{Wi} \cdot x_i(k) + Q^j_W \cdot y^j_W(k-1))
\]

\[
< 1 \cdot \frac{2}{a} \cdot \frac{a}{a} \cdot e^{-(S-b)/a^2} \cdot \cos\left(\frac{5}{a} \cdot S - \frac{b}{a}\right) - e^{-((S-b)/a)^2} \cdot \frac{5}{a} \cdot \sin\left(\frac{5}{a} \cdot S - \frac{b}{a}\right) \tag{3.12}
\]

Therefore, \( 0 < \eta_{\mathcal{C}_w} < (2 \cdot P \cdot y^2_r)/7^2 = (2 \cdot P \cdot y^2_r)/49 \)

\[
\frac{\partial \tilde{y}(k)}{\partial C^j_i} = x_i(k) \cdot W^j_i \cdot \theta'(\sum_{i=1}^{n} C^j_{Si} \cdot x_i(k))
\]

\[
< 1 \cdot \frac{2}{a} \cdot 1 + \frac{5}{a} \cdot 1 \}
\]

Therefore \( 0 < \eta_{\mathcal{C}_s} < (2 \cdot P \cdot y^2_r)/1^2 = 2 \cdot P \cdot y^2_r \)

\[
\frac{\partial \tilde{y}(k)}{\partial Q^j_W} = W^j_i \cdot y^j_W(k-1) - \psi'(\sum_{i=1}^{n} C^j_{Wi} \cdot x_i(k) + Q^j_W \cdot y^j_W(k-1))
\]

\[
< 1 \cdot \frac{2}{a} \cdot \frac{a}{a} \cdot e^{-(S-b)/a^2} \cdot \cos\left(\frac{5}{a} \cdot S - \frac{b}{a}\right) - e^{-((S-b)/a)^2} \cdot \frac{5}{a} \cdot \sin\left(\frac{5}{a} \cdot S - \frac{b}{a}\right) \tag{3.13}
\]

Therefore, \( 0 < \eta_{\mathcal{Q}_W} < (2 \cdot P \cdot y^2_r)/7^2 = (2 \cdot P \cdot y^2_r)/49 \).
From facts 3 and 4 suppose \( Z = \sum_{i=1}^{n} C_{i}^{j} \cdot x_{i}(k) \) and \( S = \sum_{i=1}^{n} C_{Wi}^{j} \cdot x_{i}(k) + Q_{Wi}^{j} \cdot y_{\theta}^{i}(k - 1) \).

For parameter \( W \) in all networks:

\[
\frac{\partial \hat{y}}{\partial W_{i}} = y_{j} = y_{\psi}^{i} \cdot y_{\theta}^{i} < 1 \cdot 1 < 1. \tag{3.15}
\]

Therefore, \( 0 < \eta_{W} < (2 \cdot P \cdot y_{\tau}^{2}) / 1 < 2 \cdot P \cdot y_{\tau}^{2} \)

\[
\frac{\partial \hat{y}(k)}{\partial C_{Wi}^{j}} = x_{i}(k) \cdot W_{i} \cdot \theta \left( \sum_{i=1}^{n} C_{Si}^{j} \cdot x_{i}(k) \right) \cdot q' \left( \sum_{i=1}^{n} C_{Wi}^{j} \cdot x_{i}(k) + Q_{Wi}^{j} \cdot y_{\psi}^{i}(k - 1) \right) \]

\[
< 1 \cdot 1 \cdot 1 \cdot \left| -\frac{2}{a} \cdot \frac{S - b}{a} \cdot e^{-(S-b)/a^2} \cdot \cos \left( 5 \frac{S - b}{a} \right) - e^{-(S-b)/a^2} \cdot \frac{5}{a} \cdot \sin \left( 5 \frac{S - b}{a} \right) \right| \]

\[
< \left\{ \frac{2}{a_{\text{min}}} \cdot 1 \cdot 1 + \frac{5}{a_{\text{min}}} \cdot 1 \right\} < 7. \tag{3.16}
\]

Therefore, \( 0 < \eta_{CS} < (2 \cdot P \cdot y_{\tau}^{2}) / (7)^{2} = (2 \cdot P \cdot y_{\tau}^{2}) / 49 \)

\[
\frac{\partial \hat{y}(k)}{\partial C_{Si}^{j}} = x_{i}(k) \cdot W_{j} \cdot \theta' \left( \sum_{i=1}^{n} C_{Si}^{j} \cdot x_{i}(k) \right) \cdot q' \left( \sum_{i=1}^{n} C_{Wi}^{j} \cdot x_{i}(k) + Q_{Wi}^{j} \cdot y_{\psi}^{i}(k - 1) \right) \]

\[
< 1 \cdot 1 \cdot 1 \cdot \left| -\frac{2}{a} \cdot \frac{S - b}{a} \cdot e^{-(S-b)/a^2} \cdot \cos \left( 5 \frac{S - b}{a} \right) - e^{-(S-b)/a^2} \cdot \frac{5}{a} \cdot \sin \left( 5 \frac{S - b}{a} \right) \right| \]

\[
< \left\{ \frac{2}{a_{\text{min}}} \cdot 1 \cdot 1 + \frac{5}{a_{\text{min}}} \cdot 1 \right\} < 7. \tag{3.17}
\]

Therefore, \( 0 < \eta_{Q_{Wi}} < (2 \cdot P \cdot y_{\tau}^{2}) / (7)^{2} = (2 \cdot P \cdot y_{\tau}^{2}) / 49 \)

\[
\frac{\partial \hat{y}(k)}{\partial Q_{Wi}^{j}} = W_{i} \cdot y_{\psi}^{i}(k - 1) \cdot \theta \left( \sum_{i=1}^{n} C_{Si}^{j} \cdot x_{i}(k) \right) \cdot q' \left( \sum_{i=1}^{n} C_{Wi}^{j} \cdot x_{i}(k) + Q_{Wi}^{j} \cdot y_{\psi}^{i}(k - 1) \right) \]

\[
< 1 \cdot 1 \cdot 1 \cdot \left| -\frac{2}{a} \cdot \frac{S - b}{a} \cdot e^{-(S-b)/a^2} \cdot \cos \left( 5 \frac{S - b}{a} \right) - e^{-(S-b)/a^2} \cdot \frac{5}{a} \cdot \sin \left( 5 \frac{S - b}{a} \right) \right| \]

\[
< \left\{ \frac{2}{a_{\text{min}}} \cdot 1 \cdot 1 + \frac{5}{a_{\text{min}}} \cdot 1 \right\} < 7. \tag{3.18}
\]

**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 \( w^m(X) = \tilde{Y}_{WNN,m} \). 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:

\[
0 < \eta_\omega < \frac{2 \cdot P \cdot y_r^2}{\left| \tilde{Y}_{WNN} \right|_{\text{max}}^2 \cdot (2/\sigma_{\min})^2},
\]

\[
0 < \eta_\pi < \frac{2 \cdot P \cdot y_r^2}{\left| \partial \tilde{Y}_{WNN}/\partial \pi \right|_{\text{max}}^2},
\]

\[
0 < \eta_c_s < \frac{2 \cdot P \cdot y_r^2}{\left| \partial \tilde{Y}_{WNN}/\partial C_S \right|_{\text{max}}^2},
\]

\[
0 < \eta_c_w < \frac{2 \cdot P \cdot y_r^2}{\left| \partial \tilde{Y}_{WNN}/\partial C_W \right|_{\text{max}}^2}.
\]

where \( \eta_\omega, \eta_c_s, \) or \( \eta_c_w \) and \( \eta_\pi \) 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. \( x_m \) and \( \sigma_m \) 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

\[
0 < \eta_\omega < \frac{2 \cdot P \cdot y_r^2}{\left| \partial \tilde{Y}_{WNF}/\partial \omega \right|_{\text{max}}^2},
\]

\[
\frac{\partial \tilde{Y}_{WNF}}{\partial \omega} = \beta_m \cdot \frac{\partial \tilde{Y}_{WNN_m}}{\partial \omega},
\]

\[
\frac{\partial \tilde{Y}_{WNF}}{\partial C_N} = \beta_m \cdot \frac{\partial \tilde{Y}_{WNN_m}}{\partial C_N},
\]

\[
\frac{\partial \tilde{Y}_{WNF}}{\partial C_W} = \beta_m \cdot \frac{\partial \tilde{Y}_{WNN_m}}{\partial C_W}.
\]

Because \( \beta_m = \mu_{A_m}(X)/\sum_{m=1}^{M} \mu_{A_m}(X) \leq 1 \) for all \( m \), therefore (3.13) to (3.15) are easily derived.
From (2.15) and (3.4) for parameters $\sigma$ or $x$, there is

\[
\frac{\partial \hat{Y}_{\text{WNN}}}{\partial \sigma} = \hat{Y}_{\text{WNN}} \cdot \frac{\beta_m}{\mu_{Am}} \cdot (1 - \beta_m) \cdot \frac{2 \cdot (x_i - \bar{x}_{mi})^2}{\sigma_{mi}^3},
\]

\[
\frac{\partial \hat{Y}_{\text{WNN}}}{\partial x} = \hat{Y}_{\text{WNN}} \cdot \frac{1}{\sum_{m=1}^{M} \mu_{Am}} \cdot \frac{2 \cdot (x_i - \bar{x}_{mi})}{\sigma_{mi}^2},
\]

and therefore (3.19) are derived.

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


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