Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 1209-1213
doi:10.4028/www.scientific.net/AEF.6-7.1209

Sensitivity Analysis of Training Neural Networks by Orthogonal Weight Functions and Its Application in Intrusion Detection

Daiyuan Zhang[1, 2, 3]

[1]College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, China

[2]Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, China

[3]Institute of Computer Technology, Nanjing University of Posts and Telecommunications, Nanjing, China

E-mail: dyzhang@njupt.edu.cn, zhangdaiyuan2011@sina.com

Keywords: neural network, weight fuction, sensitivity, intrusion detection.

Abstract. The analysis of statistical sensitivity for training neural networks using a new kind of orthogonal weight functions (OWFs) is discussed in this paper. The weights obtained after training are orthogonal functions defined on the sets of input variables (input patterns). We design a kind of classifier for intrusion detection. By extracting some parameters using the sensitivity formula of OWFs neural networks given in this paper, the test data for intrusion detection are optimized. We show that the classifier of OWFs neural networks has the advantages of optimized architecture and high detection rate.

Introduction

To overcomes the drawbacks that usually suffered from early algorithms (BP, RBF), a new type of artificial neural network's training algorithm using cubic spline weight functions (CSWFs) has been proposed in [1].

Based on [1], a new algorithm using orthogonal weight functions (OWFs) is introduced in this paper for the implementation of weight functions for training neural networks, so it has the advantages of CSWFs [1].

It is well known that sensitivity refers to how a system output is influenced by its input perturbations, and sensitivity analysis is very closely related with the architecture of neural networks and the training algorithms.

If patterns are affected by noise, the output of the network will be changed. Sensitivity is usually used to analyze this kind of changes [2], [3]. Piché [4] used a statistical approach to relate the output error to the change of weights for an ensemble of Madalines, with several activation functions such as linear, sigmoid, and threshold. On the related literature, other sensitivity function has been defined, such as output sensitivity, trajectory sensitivity, function sensitivity, etc.Based on the analysis of sensitivity of neural network using OWFs, this paper proposes the theoretical error and approximation error affected by noise.

In order to analyze the sensitivity of orthogonal weight function neural networks, the coefficients of the weight functions are calculated and the sensitivity calculation formula is deducted while the orthogonal function is applied to the Legendre orthogonal polynomial.

Intrusion detection technology is the computer software or hardware systems which can find out the unauthorized network access and attacks through the analysis of user system data, then take the alarm, and other response measures.Nowadays the advanced intrusion detection techniques used in intrusion detection including neural networks, data mining, data fusion, computer immunology and genetic algorithms.

Detection method that based on neural networks determines the invasion by extracting the mode characteristics from the normal or abnormal behaviors of the user or the systems, and creates the outline of their behavioral characteristics, according to the normal or abnormal outline during the intrusion detection to judge the exception of audit data. It can constantly learning and adjust the Mode characteristics of the subject by training, in order to build a characteristics outline which is adaptive.

It is important that selects the appropriate neural network. In this paper, the OWFs neural network is used for intrusion detection.

Network's architecture and algorithm

The network's architecture is *m*-*n*, which is different from that used in BP or RBF networks. The network has two layers, one is input layer, and the other is output layer.

There are *m* points in input layer, and denoted by x_i, $i=1,2,\cdots,m$, or we say, the input dimension of the network is *m*. Each of the input vectors $\boldsymbol{x}$ is *m*-dimensional, and x_i is *i*-th element of the vector, $i=1,2,\cdots,m$; There are *n* points in output layer, and denoted by Add_j, $j=1,2,\cdots,n$, or we say, the output dimension of the network is *n*. The neuron Add_j is used as an adder. Note that each of the *m* inputs is connected to each of the neurons (adders). each of the output vectors $\boldsymbol{z}$ is *n*-dimensional, and z_j is the *j*-th element of the vector. $w_{ji}(x_i)$ is the theoretical weight function of input patterns x_i corresponding to *j*-th neuron; z_j is the theoretical output of *j*-th neuron; $s_{ji}(x_i)$ is the approximation of $w_{ji}(x_i)$ and z_{sj} is the approximation of z_j.

The mapping relations between output layer and input layer are

$$z_j=\sum_{i=1}^{m}\eta_{ji}z_j \tag{1}$$

$$\sum_{i=1}^{m}\eta_{ji}=1,\ j=1,2,\cdots,n \tag{2}$$

$$s_{ji}(x_i)=\mathrm{Ipt}\left(x_{ip},z_{jp}\right),\ p=0,1,\cdots,N+1 \tag{3}$$

$$z_{\mathrm{sj}}=\sum_{i=1}^{m}s_{ji}(x_i) \tag{4}$$

The one-variable function $s_{ji}(x_i)=\mathrm{Ipt}\left(x_{ip},z_{jp}\right)$ can be found by *N*+2 interpolating patterns, which describes the interpolating function by the knots $\left(x_{ip},z_{jp}\right)$, and is called weight function between the *j*th output point (neuron) and the *i*th input point (variable).

The z_{sj} denotes the output values of the network's *j*th neuron, and the z_j indicates the target patterns of the *j*th point of the network.

In order to find $s_{ji}(x_i)$, the definition of best square approximation is firstly given below. Suppose that function $f(x)\subset C[a,b]$, $\varphi=\mathrm{span}\{\phi_0(x),\cdots,\phi_l(x)\}$ is a subset of $C[a,b]$. Suppose that the function $S^*(x)\in\varphi$ can satisfy the following formula

$$\left\|f(x)-S^*(x)\right\|_2^2=\inf_{S^*(x)\in\varphi}\int_a^b\rho(x)\left[f(x)-S^*(x)\right]^2\mathrm{d}x \tag{5}$$

The S^* is the approximation function in the subset of φ. If the approximation polynomial is expressed in the following

$$S^*(x)=a_0g_0(x)+a_1g_1(x)+\cdots+a_l\,\mathrm{g}_l(x) \tag{6}$$

The coefficients of (6) for the approximation polynomial can be expressed as the following

$$a_k=\frac{(f,g_k)}{(g_k,g_k)},\ k=0,1,\cdots,l \tag{7}$$

On the condition of multi-dimensional input and multi-dimensional output, the error of orthogonal weight function neural networks can be expressed as

$$J_j=\left\|z_j-z_{sj}\right\|_2^2=\left\|\sum_{i=1}^{m}w_{ji}(x_i)-\sum_{i=1}^{m}s_{ji}(x_i)\right\|_2^2\le m\cdot\max_{1\le i\le m}\left(\left(\frac{2M}{(l+1)!}\left(\frac{b-a}{4}\right)^{l+1}\right)^2\right) \tag{8}$$

$$J = \max_{1\le j\le n} J_j = \max_{1\le j\le n} \| z_j - z_{sj} \|_\infty \le m \cdot \max_{1\le j\le n}\left(\max_{1\le i\le m}\left(\frac{2M}{(l+1)!}\left(\frac{b-a}{4}\right)^{l+1}\right)^2\right) \tag{9}$$

where $M = \max_{1\le i\le m}\left| w_{ji}^{(l+1)}(x_i)\right|$.

The expressions (8) and (9) are important for some applications of generalization.

Sensitivity Analysis

Suppose that $\boldsymbol{x}^*$ is the vector of input patterns, and is the vector of perturbation, then the disturbed input patterns can be expressed as following

$$\boldsymbol{x} = \boldsymbol{x}^* + \boldsymbol{\Delta x} \tag{10}$$

In this paper, two kinds of sensitivity are discussed, i.e. the sensitivity of theoretical error and the sensitivity of approximation error. Below gives the theoretical error.

$$\left\| w(\boldsymbol{x}^* + \boldsymbol{\Delta x}) - w(\boldsymbol{x}^*)\right\| \le \left\| w(\boldsymbol{x}^* + \boldsymbol{\Delta x}) - s(\boldsymbol{x}^* + \boldsymbol{\Delta x})\right\| + \left\| s(\boldsymbol{x}^* + \boldsymbol{\Delta x}) - s(\boldsymbol{x}^*)\right\| + \left\| w(\boldsymbol{x}^*) - s(\boldsymbol{x}^*)\right\| \tag{11}$$

The theoretical error of a system can include model error, and approximation error. Model error can be expressed as $\left\| w(\boldsymbol{x}^* + \boldsymbol{\Delta x}) - s(\boldsymbol{x}^* + \boldsymbol{\Delta x})\right\|$ and $\left\| w(\boldsymbol{x}^*) - s(\boldsymbol{x}^*)\right\|$, and the approximation error is $\left\| s(\boldsymbol{x}^* + \boldsymbol{\Delta x}) - s(\boldsymbol{x}^*)\right\|$.

Assume that vector of the *p*-th noise-free patterns input is $\boldsymbol{x}_p^* = \left(x_{1p}^*\ x_{2p}^*\ \ldots\ x_{mp}^*\right)^T$, where $p = 0,1,2,\ldots,N+1$. In this case, the theoretical output of the *j*-th neuron is

$$y_{jp}^* = \sum_{i=1}^{m} w_{ji}\left(x_{ip}^*\right) \quad (j = 1,2,\ldots n) \tag{12}$$

Assume that the noise vector of input patterns is $\Delta\boldsymbol{x}_p = \left(\Delta x_{1p}\ \Delta x_{2p}\ \ldots\ \Delta x_{mp}\right)^T$, the theoretical output caused by noise is

$$y_{jp} = \sum_{i=1}^{m} w_{ji}\left(x_{ip}\right) = \sum_{i=1}^{m} w_{ji}\left(x_{ip}^* + \Delta x_{ip}\right) \tag{13}$$

The output error of theoretical noise of *j*-th neuron is

$$\begin{aligned}\Delta y_{jp} = y_{jp} - y_{jp}^* &= \sum_{i=1}^{m} w_{ji}\left(x_{ip}^* + \Delta x_{ip}\right) - \sum_{i=1}^{m} w_{ji}\left(x_{ip}^*\right) = \sum_{i=1}^{m}\left(s_{ji}\left(x_{ip}^* + \Delta x_{ip}\right) - s_{ji}\left(x_{ip}^*\right)\right) \\ &+ \sum_{i=1}^{m}\left(w_{ji}\left(x_{ip}^* + \Delta x_{ip}\right) - s_{ji}\left(x_{ip}^* + \Delta x_{ip}\right)\right) - \sum_{i=1}^{m}\left(w_{ji}\left(x_{ip}^*\right) - s_{ji}\left(x_{ip}^*\right)\right)\end{aligned} \tag{14}$$

Suppose that the function is as following

$$f_{ji}(x) = w_{ji}(x) - s_{ji}(x) \tag{15}$$

Formula (14) can be transformed as follows

$$\Delta y_{jp} = y_{jp} - y_{jp}^* = \sum_{i=1}^{m} \cdot \Delta x_{ip}\left(\sum_{k=0}^{l} P'\left(k, x_{ip}^* + \varepsilon_1 \cdot \Delta x_{ip}\right) c_{jik} + f_{ji}'\left(x_{ip}^* + \varepsilon_2 \cdot \Delta x_{ip}\right)\right) \tag{16}$$

In (16), $P(k,x)$ is the value of the *k*-th item of orthogonal polynomials, $P' = \frac{\partial P}{\partial x}$, and $\varepsilon_1, \varepsilon_2 \in (0,1)$. When the perturbations of input patterns tends to zero, we have

$$\Delta y_{jp} = y_{jp} - y_{jp}^* = \sum_{i=1}^{m}\left(\Delta x_{ip} \cdot \left(P_{ip}' \cdot C_{ji} + f_{ji}'\left(x_{ip}^*\right)\right)\right) \tag{17}$$

where

$$\begin{cases} C_{ji} = \left(c_{ji0} \quad c_{ji1} \quad \cdots \quad c_{jil}\right)^T \\ P_{ip}' = \left(P'\left(0, x_{ip}^*\right) \quad P'\left(1, x_{ip}^*\right) \quad \cdots \quad P'\left(l, x_{ip}^*\right)\right) \end{cases} \tag{18}$$

The output perturbations can be expressed in the following

$$\Delta \boldsymbol{y}_p = \left(\sum_{i=1}^{m} \Delta x_{ip} \cdot \left(P'_{ip} \cdot C_{1i} + f'_{1i}\left(x^*_{ip}\right) \right) \quad \cdots \quad \sum_{i=1}^{m} \Delta x_{ip} \cdot \left(P'_{ip} \cdot C_{ni} + f'_{ni}\left(x^*_{ip}\right) \right) \right) \tag{19}$$

The definition of statistical sensitivity for weight function neural networks can be expressed as

$$S(\boldsymbol{X}) := \lim_{\sigma \to 0} \frac{\mathrm{sqrt}\left[\mathrm{var}\left[\Delta \boldsymbol{Y} \right] \right]}{\sigma} \tag{20}$$

Assume that the input variables are independent, we have

$$\mathrm{var}\left(\Delta \boldsymbol{y}_p \right) = \sum_{i=1}^{m} \mathrm{var}\left(\Delta x_{ip} \cdot \left(P'_{ip} \cdot C_{1i} + f'_{1i}\left(x^*_{ip}\right) \quad \cdots \quad P'_{ip} \cdot C_{ni} + f'_{ni}\left(x^*_{ip}\right) \right) \right) \tag{21}$$

Let

$$\mathrm{var}\left(\Delta x_{1p} \right) = \cdots = \mathrm{var}\left(\Delta x_{mp} \right) = \sigma^2 \tag{22}$$

the formula (21) can be expressed in the following

$$\mathrm{var}\left(\Delta \boldsymbol{y}_{\boldsymbol{p}} \right) = \sigma^2 \cdot \sum_{i=1}^{m} \mathrm{var}\left(P'_{ip} \cdot C_{1i} + f'_{1i}\left(x^*_{ip}\right) \quad \cdots \quad P'_{ip} \cdot C_{ni} + f'_{ni}\left(x^*_{ip}\right) \right) \tag{23}$$

The theoretical sensitivity is

$$S\left(\boldsymbol{x}_p \right) = \mathrm{sqrt}\left[\sum_{i=1}^{m} \mathrm{var}\left(P'_{ip} \cdot C_{1i} + f'_{1i}\left(x^*_{ip}\right) \quad \cdots \quad P'_{ip} \cdot C_{ni} + f'_{ni}\left(x^*_{ip}\right) \right) \right] \tag{24}$$

We can calculate the sensitivity values based on theoretical error by (24). And the sensitivity of approximation error is

$$S(\boldsymbol{x}_p) = \mathrm{sqrt}\left[\sum_{i=1}^{m} \mathrm{var}\left(P'_{ip} \cdot C_{1i} \quad \cdots \quad P'_{ip} C_{ni} \right) \right] \tag{25}$$

Example

From the results of sensitivity analysis given in this paper, we see that, when the disturbance of input patterns is increased, the corresponding error of the network's output will also be increased. By this principle we can remove some undetected patterns for higher detection rate.

The example given below describes the intrusion detection by orthogonal weight function neural networks. The network's architecture used in this example is 24-1, which means that there are 24-dimensional input nodes and 1-dimensional output node. This example adopts 540 data patterns, including back Ipsweep, Satan attack, attack and normal data stream, and some remaining undetected data.

We select a group of data as learning patterns, calculate the perturbation of other data patterns (test patterns), and then according to the values of sensitivity, remove those intrusion detection data with large values of perturbation. The simulation results are as follows.

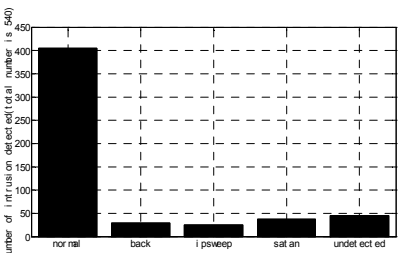

Fig. 1 Intrusion detection experiment with original data

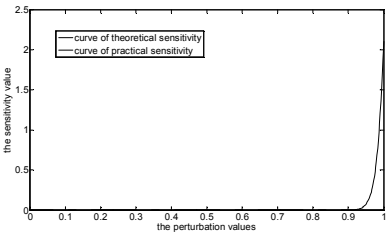

Fig. 2 Curve of sensitivity

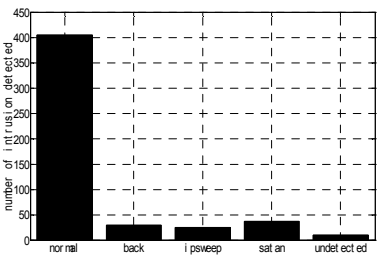

Fig. 3 Intrusion detection experiment with optimized data

Fig. 1 shows the classification results of the histogram with 540 intrusion detection data sets.

Fig. 2 shows that actual sensitivity curve of 540 invasion of input disturbance, as can be seen, in range of [0.95, 1], numerical value of sensitivity has a sudden sharp change, which means that the invasion of network input disturbance exerts an enormous influence. Because intrusion detection information input perturbation is accomplished by normalization of, to [0.95, 1] for boundaries, the input information of the original data can be deduced, and then these information will be removed.

Using the sensitivity analysis given in this paper, we can improve the detection efficiency, see Fig. 3.

Acknowledgment

This work was supported by the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001).

I thank T. Guo, my graduate student, for her simulation examples given in this paper.

References

[1] D. Y. Zhang, New theories and methods on neural networks. Beijing: Tsinghua university press, 2006 (in Chinese).

[2] J.Y Choi and C. Choi. Sensitivity analysis of multilayer perception with differentiable activation functions. IEEE Trans. Neural Netw., 1992,3(1): 101-107.

[3] M. Stevenson, R.Winter, and B. Widrow. Sensitivity of feedforward neural networks to weight errors. IEEE Trans. Neural Netw., 1990,1(1): 71-80.

[4] S. W. Piché. The selection of weight accuracies for Madalines. IEEE Trans. Neural Netw., 1995,6(2): 432-445.