Machine Intelligence & Robotic Control, Vol. 1, No. 1, 5–9 (1999)

Paper

Theoretical Investigation of the Faulty Behavior of Feedforward Neural Networks with Differentiable Activation Functions*

Tao Zhang† and Dongcheng Hu†

Abstract: Based on the statistical approach, a kind of fault-tolerance analysis method for neural networks is proposed in detail by taking large-scale feedforward neural network (LFNN) as an object. Firstly, a stochastic fault model for LFNN is built in view of link faults and error input faults, which appear frequently in the hardware implementation of neural networks. In addition, with this model the features of the fault propagation of LFNN are studied. Next, all neurons are divided into two types of fault characteristics, i.e., some of them have only link faults and others have both link faults and error input faults. Then their correct output probability formulas are given respectively described by two theorems and two inferences, which are proved in the paper. Finally, using the results of above fault analysis of neurons, the algorithm of the correct output probability for LFNN is presented. The computer simulation is also made to show the correctness of the proposed algorithm.

Keywords: Feedforward Neural Networks, Fault-Tolerance Analysis, Central Limit Theorem

1. Introduction

WITH the further study of the hardware implementation of neural networks and the appearance of various neural network hardware, the problem of faulty behavior and fault tolerance of neural network hardware is recently becoming more and more important. Since the faulty behavior of neural networks is lack of the theoretical analysis, especially for the large-scale neural networks, the fault tolerance of neural networks is judged blindly to a certain extend.

At present, robustness analysis has been studied in some published papers [1]–[3]. However, there are still few papers about theoretical fault-tolerance analysis. We can only find out several papers about simulation of fault-tolerant behavior [4], or about fault-tolerance design [5]. However, in [4], [5], there are short of sufficient theoretical fault-tolerance analysis for simulation and it has been not proved that these fault-tolerance design methods can improve effectively fault tolerance of neural networks in theory. Therefore, for above reasons, the purpose of this paper is to focus on the theoretical fault-tolerance analysis and to present an effective analysis method for LFNN.

The remainder of this paper is organized as follows. First of all, a stochastic fault model of LFNN is presented. Then the fault-tolerance analysis for neurons and LFNN is discussed in detail. Finally computer simulations are made for the verification of conclusions.

2. Stochastic Fault Model of LFNN

2. 1 Stochastic fault model of LFNN

LFNN discussed in this paper is composed of L layers. The first layer is the input layer and the L-th layer is the output layer. The L-2 mid-layers are the hidden layers. In addition, each neuron is connected with all neurons in the neighboring layers and there are no any couplings in the same layer. The input is transferred through the hidden layers to the output layer. The relationship among the neurons in two neighboring layers can be described by the following formula.

$$x_j^{(l)} = f\left(\sum_{i=1}^{N^{(l-1)}} w_{ij}^{(l)} x_i^{(l-1)} - \theta_j^{(l)}\right), \quad j = 2, \cdots, N^{(l)} \tag{1}$$

where $N^{(l-1)}$ is the neuron number of layer $l-1$ $(l = 2, \cdots, N)$; $x_j^{(l)}$ is the output of neuron j of layer l; and $w_{ij}^{(l)} (i = 1, \cdots, N^{(l-1)})$ is the weight between neuron i of layer $l-1$ and neuron j of layer l. The value of $w_{ij}^{(l)}$ is between $+M$ and $-M$ (M is constant and $M > 0$) and $\theta_j^{(l)}$ is the threshold. Additionally, differentiable activation functions are considered in this paper for LFNN. The typical function is Sigmoid function whose formula is given by

$$u = \tanh(v) = \frac{1-\exp(-v)}{1+\exp(-v)}. \tag{2}$$

Besides, for LFNN assume $N^{(l)} \gg 1$ and $\theta_j^{(l)}$ $(l = 1, \cdots, N)$ equal to 0 for simplification.

Two types of faulty behaviors appearing frequently in the hardware implementation of LFNN will be concerned in this paper. One is link fault, and the other is error input fault. The so-called link fault refers to that links among the neurons have the stuck-at-M (or stuck-at-($-M$), or stuck-at-0) faults that can cause the corresponding weights and input to lose the functions. Since the fault-tolerance analysis for the stuck-at-($-M$) fault can be also made similarly by the following proposed method and the stuck-at-0 fault is a special case for LFNN, thus the stuck-at-M fault and the stuck-at-0 fault will be only considered in this paper. The so-called error input fault refers to that the error output caused by the link faults of the last layer is the input of each neuron in this layer. In order to reflect the influence of error input fault, we define its value equal to the opposite value of correct input.

* Received August 20, 1999; accepted August 28, 1999. This work was supported by China National Science Foundation and also by China National Education Foundation for Ph.D. training.

† Department of Automation, Tsinghua University, Beijing 100084, P.R. China. E-mail: zhangtao@hotmail.com; hudc@mail.tsinghua.edu.cn

Paper No. 1345–269X/99/010005-05

For studying the propagation and fault-tolerant statistical characteristics of faults, the stochastic fault model of LFNN is defined as follows. Assume the input of layer $l-1$, $x_j^{(l-1)}(j=1,\cdots,N^{(l-1)})$, is independent identically distributed (IID) with mean $\mu_x^{(l-1)}$ and variance $\left(\sigma_x^{(l-1)}\right)^2$; the weight of each neuron in every layer is also assumed IID with mean 0 and variance σ_w^2. The input and weights are mutually independent. The performance of all neurons in the same layer is identical and independent.

2.2 Fault propagation of LFNN

According to the stochastic fault model of LFNN, the propagation and quantitative variation of link faults and error input faults in LFNN can be assumed as follows:

1. The initial input of LFNN is correct;
2. Since the identical performance of each neuron in the same layer and the complete coupling, the probabilities of existing link faults of neurons are all P_f $(P_f \in (0,1))$;
3. There are only link faults in the first layer;
4. There are both link faults and error input faults in the hidden layers and the output layer. However, the probability of existing error input faults of each layer must be calculated with the following method.
   (a) Since the performances of each neuron in the same layer are identical and IID, the number of error input faults for layer l can be given approximately by the mean
   $$\bar{N}_e^{(l)} = \left(1-P_c^{(l-1)}\right)\left(N^{(l-1)}-N_f^{(l-1)}\right) \tag{3}$$
   where $P_c^{(l-1)}$ is the correct output probability of layer $l-1$ $(l=2,\cdots,L)$;
   (b) When there are link faults in the layer $l-1$, the output of layer l will be still IID with mean $\mu_x^{(l)}$ and variance $\left(\sigma_x^{(l)}\right)^2$ in accordance with theorem 1.

Theorem 1: The input of the neuron i $(i = 1,\cdots,N^{(l-1)})$ in the layer $l-1$ $(l=2,\cdots,N)$ of LFNN, $x_i^{(l-1)}$, is IID with mean $\mu_x^{(l-1)}$ and variance $\left(\sigma_x^{(l-1)}\right)^2$. Its weight is also IID with mean 0 and variance σ_w^2. The input and weights are mutually independent. If there are link faults in LFNN, the output of layer l will be still IID and its mean and variance respectively are:

$$\mu_x^{(l)} = \int_{-\infty}^{+\infty} \frac{1-AB(x)}{1+AB(x)}\cdot\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)dx \tag{4}$$

$$\begin{aligned}\left(\sigma_x^{(l)}\right)^2 &= \int_{-\infty}^{+\infty}\left(\frac{1-AB(x)}{1+AB(x)}\right)^2\cdot\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)dx \\ &\quad -\left\{\int_{-\infty}^{+\infty}\frac{1-AB(x)}{1+AB(x)}\cdot\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)dx\right\}^2\end{aligned} \tag{5}$$

where

$$\sigma_{wx}^2 = \sigma_w^2\sigma_x^2 + \sigma_w^2\mu_x^2, \quad A = \exp(-N_f M),$$

$$B(x) = \exp\left(-\sqrt{N^{(l-1)}-N_f^{(l-1)}}\,\sigma_{wx}^{(l)}x\right).$$

Proof: When there are link faults in LFNN, the output of any a neuron j $(j = 1,\cdots,N^{(l)})$ of layer l $(l=2,\cdots,N)$ is given by

$$x_j^{(l)} = \tanh\left(\sum_{i\in R} w_{ij}^{(l)}x_i^{(l-1)} + N_f M\right) \tag{6}$$

where R is the set of correct links and N_f is the number of link faults.

Defining

$$I_j^{(l)} = \sum_{i\in R} w_{ij}^{(l)}x_i^{(l-1)} \tag{7}$$

and

$$I_k^{(l)} = \sum_{i\in R} w_{ik}^{(l)}x_i^{(l-1)}, \quad j,k\in N^{(l)} \tag{8}$$

then we have

$$S = \frac{I_j^{(l)}}{\sqrt{N^{(l-1)}-N_f^{(l-1)}}\,\sigma_{wx}^{(l)}} \tag{9}$$

and

$$T = \frac{I_k^{(l)}}{\sqrt{N^{(l-1)}-N_f^{(l-1)}}\,\sigma_{wx}^{(l)}}. \tag{10}$$

When $N^{(l-1)} \gg 1$, both S and T tend to the normal distribution with mean 0 and variance 1 in the light of central limit theorem. The correlative coefficient between S and T can be calculated as

$$\begin{aligned}\rho(S,T) &= \frac{1}{\left(N^{(l-1)}-N_f^{(l-1)}\right)\left(\sigma_{wx}^{(l)}\right)^2} \\ &\quad \cdot\sum_{i\in R} E\left[w_{ij}^{(l)}w_{ik}^{(l)}\left(x_i^{(l-1)}\right)^2\right] = 0.\end{aligned} \tag{11}$$

Therefore, S and T are mutually IID. So, the output of layer l will be still IID and its mean and variance can be calculated as

$$\begin{aligned}E\left[x_j^{(l)}\right] &= \mu_x^{(l)} = E\left[\frac{1-A\exp\left(-\sum_{i\in R}w_{ij}^{(l)}x_i^{(l-1)}\right)}{1+A\exp\left(-\sum_{i\in R}w_{ij}^{(l)}x_i^{(l-1)}\right)}\right] \\ &= \int_{-\infty}^{+\infty}\frac{1-AB(x)}{1+AB(x)}\cdot\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)dx\end{aligned} \tag{12}$$

$$\begin{aligned}\mathrm{Var}\left[x_j^{(l)}\right] &= \left(\sigma_x^{(l)}\right)^2 = E\left[\left(x_j^{(l)}\right)^2\right] - E^2\left[x_j^{(l)}\right] \\ &= \int_{-\infty}^{+\infty}\left(\frac{1-AB(x)}{1+AB(x)}\right)^2\cdot\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)dx \\ &\quad -\left\{\int_{-\infty}^{+\infty}\frac{1-AB(x)}{1+AB(x)}\cdot\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)dx\right\}^2.\end{aligned} \tag{13}$$

■

3. Fault Analysis of Neurons

All neurons in LFNN are divided into two types of fault characteristics, i.e., some of them have only link faults and others have both link faults and error input faults. Then their formulas of correct output probability can be respectively given by the following two theorems and two inferences, whose proof will be shown in the appendix.

3.1 Fault tolerance of neurons with link faults

Theorem 2: If each neuron of LFNN has N_f stuck-at-M faults on its N links, its correct output probability is

$$P_c = \frac{1}{2\pi\sqrt{(N-N_f)N_f}\sigma_{wx}^2} \cdot \iint_D \frac{1}{xy}\exp\left(-\frac{(\ln x)^2}{2(N-N_f)\sigma_{wx}^2} - \frac{(\ln y)^2}{2N_f\sigma_{wx}^2}\right)dxdy \tag{14}$$

where $\sigma_{wx}^2 = \sigma_w^2\sigma_x^2 + \sigma_w^2\mu_x^2$, $D: \left|\frac{2x(y-A)}{(1+Ax)(1+xy)}\right| < \delta$, in which $x, y > 0$, $\delta > 0$ with upper bound, and $A = \exp(-N_f M)$.

Inference 1: If each neuron of LFNN has N_f stuck-at-0 faults on its N links, its correct output probability is

$$P_c = \frac{1}{2\pi\sqrt{(N-N_f)N_f}\sigma_{wx}^2} \cdot \iint_D \frac{1}{xy}\exp\left(-\frac{(\ln x)^2}{2(N-N_f)\sigma_{wx}^2} - \frac{(\ln y)^2}{2N_f\sigma_{wx}^2}\right)dxdy \tag{15}$$

where $\sigma_{wx}^2 = \sigma_w^2\sigma_x^2 + \sigma_w^2\mu_x^2$, $D: \left|\frac{2x(y-1)}{(1+x)(1+xy)}\right| < \delta$, $x, y > 0$, $\delta > 0$ with upper bound.

3.2 Fault tolerance of neurons with link faults and error input faults

Theorem 3: If each neuron of LFNN has N_f stuck-at-M faults on its N links and N_e error input faults on its rest links which have no link faults, its correct output probability is

$$P_c = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{(N-N_f-N_e)N_eN_f}\sigma_{wx}^3} \cdot \iiint_\Omega \frac{1}{xyz}\exp\left(-\frac{(\ln x)^2}{2(N-N_f-N_e)\sigma_{wx}^2} - \frac{(\ln y)^2}{2N_e\sigma_{wx}^2} - \frac{(\ln z)^2}{2N_f\sigma_{wx}^2}\right)dxdydz \tag{16}$$

where $\sigma_{wx}^2 = \sigma_w^2\sigma_x^2 + \sigma_w^2\mu_x^2$, $\Omega: \left|\frac{2x(y^2z-A)}{(y+Ax)(1+xyz)}\right| < \delta$, in which $x, y, z > 0$, $\delta > 0$ with upper bound, and $A = \exp(-N_f M)$.

Inference 2: If each neuron of LFNN has N_f stuck-at-0 faults on its N links and N_e error input faults on its rest links which have no link faults, its correct output probability is

$$P_c = \frac{1}{(2\pi)^{\frac{3}{2}}\sqrt{(N-N_f-N_e)N_eN_f}\sigma_{wx}^3} \cdot \iiint_\Omega \frac{1}{xyz}\exp\left(-\frac{(\ln x)^2}{2(N-N_f-N_e)\sigma_{wx}^2} - \frac{(\ln y)^2}{2N_e\sigma_{wx}^2} - \frac{(\ln z)^2}{2N_f\sigma_{wx}^2}\right)dxdydz \tag{17}$$

where $\sigma_{wx}^2 = \sigma_w^2\sigma_x^2 + \sigma_w^2\mu_x^2$, $\Omega: \left|\frac{2x(y^2z-1)}{(y+x)(1+xyz)}\right| < \delta$, in which $x, y, z > 0$, $\delta > 0$ with upper bound.

4. Fault-Tolerance Analysis of LFNN

With the fault propagation characteristics of LFNN and the fault analysis of neurons, the algorithm of fault-tolerance analysis for LFNN can be given as the following process.

Step 1: When $l = 1$, calculate probability $P_c^{(l)}$ according to Theorem 2 (when $M = 0$, calculate it according to Inference 1);

Step 2: Setting $l = l + 1$,

$$\bar{N}_e^{(l)} = \left(1 - P_c^{(l-1)}\right)\left(N^{(l-1)} - N_f^{(l-1)}\right);$$

Step 3: Calculate the mean and variance of the input of layer l according to Theorem 1;

Step 4: Calculate $P_c^{(l)}$ according to Theorem 3 (when $M = 0$, calculate it according to Inference 2);

Step 5: If $l = L$, then enter into next step, otherwise turn to step 2;

Step 6: $P_c^{(l)}$ is just the correct output probability.

5. Simulations

Although the results of fault-tolerance analysis of LFNN can be given by the above analysis, it is still hard to obtain the numerical results of fault-tolerance analysis for LFNN with Sigmoid activation function. In order to verify the correctness of the algorithm proposed in this paper, the numerical results worked out by two approximate methods are used to make comparison with the computer Monte Carlo simulation.

5.1 Linear activation function

We chose the linear activation function to take the place of Sigmoid activation function and made similarly fault-tolerance analysis for LFNN by the proposed method in this paper. With the comparison between the theoretical analysis and the computer simulation for three-layer LFNN, the correctness of proposed algorithm in this paper can be distinctly shown by **Figs. 1** and **2**. Note here that the programming language of computer simulation was VC++ 5.0 and the environment of computer simulation was on the personal computer with Pentium I-233 MMX processor.

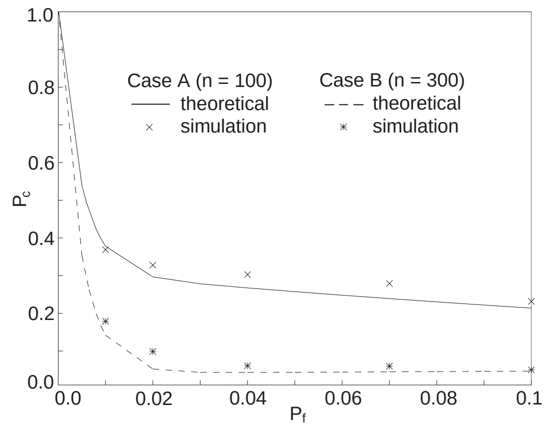

Fig. 1 Fault tolerance of three-layer LFNN with stuck-at-M link faults

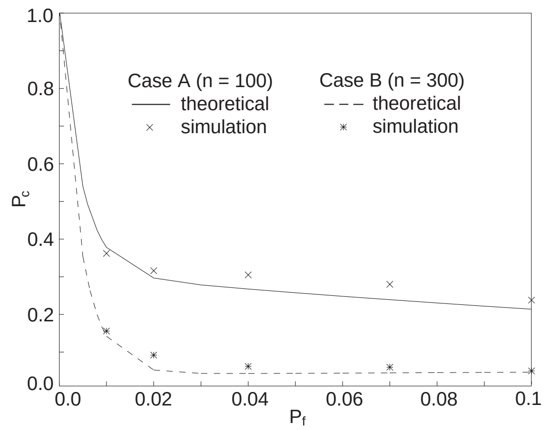

Fig. 2 Fault tolerance of three-layer LFNN with stuck-at-0 link faults

5.2 Chebyshev inequality method

For the fault-tolerance analysis of LFNN with Sigmoid activation function, we also chose Chebyshev inequality method, as another approximate method, to calculate approximate numerical results of fault-tolerance analysis of LFNN for the verification of the proposed algorithm.

The definition of Chebyshev inequality method is given in [6]. Assume a random variable X has limited mean μ and variance σ^2. For any real number $\epsilon > 0$, $P\{|X-\mu| \geq \epsilon\} \leq \sigma^2/\epsilon^2$ which has another form given by $P\{|X-\mu| < \epsilon\} \geq 1-\sigma^2/\epsilon^2$. The importance of this formula is that the bound of probability for the deviation of random variable is easily obtained as long as the mean and variance of stochastic variable are known. With this method, the approximate numerical results of fault-tolerance analysis of LFNN with Sigmoid activation function, in the form of the bound of probability, can be worked out. The comparison between the theoretical analysis and the computer simulation was made as follows:

5.2.1 Theoretical analysis Here, three-layer *LFNN* was taken as an example. If define $N^{(l)} = 50$ $(l = 1, 2, 3)$, $P_f^{(l)} = 0.04$, and $\sigma_w^{(1)} = \frac{1}{5}$, $\sigma_x^{(1)} = \frac{1}{3}$, $\mu_x^{(1)} = \frac{1}{70}$, $\mu_w^{(1)} = 0.2556625$ for the first layer, and $\sigma_w^{(l)} = \frac{1}{3}$, $\sigma_x^{(l)} = 0.4896808$, $\mu_x^{(l)} = 0.398942$, $\mu_w^{(l)} = 1.8991412$ $(l = 2, 3)$ for the second and third layers. When $\epsilon = 0.45$, the lower bound of correct output probability of three-layer LFNN P_c was $P_c \geq 0.9536753$.

5.2.2 Simulation Simulation was also made on the personal computer with Intel Pentium I by the programming language VC++ 5.0 according to the computer Monte Carlo method. The conditions were the same as 5.2.1. The simulation time was 2,000. The final result was $P_c = 0.9695$ which is greater than the theoretical result. It shows that the theoretical analysis with Chebyshev inequality method is correct.

6. Conclusions

In this paper, the fault tolerance of LFNN has been discussed. From the theoretical analysis and computer simulation, we could draw out the following conclusions.

1. Although someone thinks that the larger is the scale of neural networks, the better is the fault tolerance of neural networks, according to the results in this paper, expanding the scale is not good for improving the fault tolerance of neural networks, but harmful. Therefore, only by relying on the correct analysis of fault tolerance of neural networks can we work out ANN hardware of neural networks with the high reliability;
2. A new algorithm for the fault-tolerance analysis of LFNN presented in this paper, based on the analysis of faults of neurons in each layer, can be also used for other kinds of neural networks.

Further explanation is omitted here, due to the space limitation.

7. Appendix

7.1 Proof of Theorem 2

If neuron has no faults, the output of neuron is obtained by

$$G = f\left(\sum_{i=1}^{N} w_i x_i\right) = \tanh\left(\sum_{i\in R} w_i x_i + \sum_{i\in F} w_i x_i\right) \tag{18}$$

where N is the input number, R is the set of correct links, and F is the set of error links. If neuron has N_f stuck-at-M link faults, the output of neuron is given by

$$\widetilde{G} = \frac{1-\exp\left\{-\left(\sum_{i\in R} w_i x_i + N_f M\right)\right\}}{1+\exp\left\{-\left(\sum_{i\in R} w_i x_i + N_f M\right)\right\}}. \tag{19}$$

Therefore the output difference is $\Delta G = \widetilde{G} - G$. Setting

$$X = \exp\left(-\sum_{i\in R} w_i x_i\right), \quad Y = \exp\left(-\sum_{i\in F} w_i x_i\right),$$

then we have

$$\Delta G = \frac{2X(Y-A)}{(1+AX)(1+XY)}.$$

Since the input of neuron x_i $(i = 1, \cdots, N)$ is IID with mean μ_x and variance σ_x^2, its weight w_i $(i = 1, \cdots, N)$ is IID with mean 0 and variance σ_w^2 and they are mutually independent, then $w_i x_i$ $(i = 1, \cdots, N)$ is also IID with mean 0 and variance $\sigma_w^2\sigma_x^2 + \sigma_w^2\mu_x^2$. Then according to central limit theorem, $U = \sum_{i \in R} w_i x_i / \sqrt{N - N_f}\sigma_{wx}$ tend to the normal distribution with mean 0 and variance 1 on the condition of $N \gg 1$. So, the probability density function of X is

$$f_X(x) = \frac{1}{\sqrt{2\pi(N-N_f)}\sigma_{wx} x} \exp\left(-\frac{(\ln x)^2}{2(N-N_f)\sigma_{wx}^2}\right). \quad (20)$$

Similarly, the probability density function of Y is

$$f_Y(y) = \frac{1}{\sqrt{2\pi N_f}\sigma_{wx} y} \exp\left(-\frac{(\ln y)^2}{2N_f\sigma_{wx}^2}\right). \quad (21)$$

Since $\sum_{i \in R} w_i x_i$ and $\sum_{i \in F} w_i x_i$ $(i = 1, \cdots, N)$ are mutually independent, $\exp(-\sum_{i \in R} w_i x_i)$ and $\exp\left(-\sum_{i \in F} w_i x_i\right)$ are continuous functions, so they are both Borel measurable function. Therefore X and Y are mutually independent.

The correct output probability of neuron is

$$\begin{aligned} P_c &= P\{|\Delta G| < \delta\} \\ &= P\left\{\left|\frac{2X(Y-A)}{(1+AX)(1+XY)}\right| < \delta\right\} \\ &= \iint_D f_X(x) f_Y(y) dx dy \\ &= \frac{1}{2\pi\sqrt{(N-N_f)N_f}\sigma_{wx}^2} \\ &\quad \cdot \iint_D \frac{1}{xy} \exp\left(-\frac{(\ln x)^2}{2(N-N_f)\sigma_{wx}^2} - \frac{(\ln y)^2}{2N_f\sigma_{wx}^2}\right) dx dy \end{aligned} \quad (22)$$

where $\sigma_{wx}^2 = \sigma_w^2 \sigma_x^2 + \sigma_w^2 \mu_x^2$, D : $\left|\frac{2x(y-A)}{(1+Ax)(1+xy)}\right| < \delta$, in which $x, y > 0$, $\delta > 0$ with upper bound, and $A = \exp(-N_f M)$.

Referring to the proof of Theorem 2 can prove Theorem 3, Inference 1, and Inference 2 in this paper. Their proofs will be omitted here because of the limitation of space.

References

[1] P. Kerlirzin and P. Refregier, "Theoretical investigation of the robustness of multilayer perceptrons," *IEEE Trans. on Neural Networks*, vol. 6, no. 3, pp. 560–571, 1995.

[2] K. N. Wang and A. N. Michel, "Robustness and perturbation analysis of a class of neural networks," *Neural Networks*, vol. 7, no. 2, pp. 251–259, 1994.

[3] L. T. Yang, D. C. Hu, and Y. P. Luo, "Error response and robustness of a class of multilayered perceptrons with threshold functions,"*Journal of Electronics*, vol. 16, no. 2, pp. 179–186, 1999.

[4] C. H. Seguin and R. D. Clay, "Fault tolerance in artificial neural networks," in *Proc. of IJCNN*, San Diego, USA, 1990, vol. 1, pp. 703–808.

[5] D. S. Phatak and I. Koren, "Complete and partial fault tolerance of feedforward neural nets," *IEEE Trans. on Neural Networks*, vol. 6, no. 2, pp. 446–456, 1995.

[6] Fudan University, *Probability Theory*. Publishing House of Higher Education, 1987.

Biographies

Tao Zhang received B.E. and M.E. degrees in the Department of Automation from Tsinghua University in 1993 and 1995 respectively, and a D.E. degree in the Department of Automation from Tsinghua University in 1999.

From September in 1995, he is an assistant professor in the Department of Automation, Tsinghua University.

His research interests are in the fault-tolerance analysis and design, reliability estimation, computer control systems and neural networks, applications of communication networks and so on.

Dongcheng Hu graduated from the Department of Electrical Engineering, Tsinghua University in 1970. He has the academic background of automatic control and is now a vice president of Tsinghua University. He is the council member of the Chinese Society of Automation and the senior member of the Chinese Society of Electronics.

His research interests cover automatic testing, fault diagnosis and reliability.