

Sensitivity Analysis of Multilayer Perceptron*

Daniel S. Yeung, Xuequan Sun, and Xiaoqin Zeng

Department of Computing, The Hong Kong Polytechnic University

Abstract

Sensitivity analysis of neural network is useful for network design. Piché used a stochastic model to describe the Multilayer Perceptron (MLP), but it doesn't match the true MLP closely, and too severe limitations are imposed on both input and weight perturbations. This paper attempts to generalize Piché's stochastic model of MLP, and derive an universal expression of MLP's sensitivity for all sigmoidal activation functions, without any restriction on input and output perturbations. The effects of network design parameters such as the number of layer, the number of neuron per layer and the chosen activation function are analyzed, and they provide useful information for network design decision-making. Furthermore, we use our sensitivity expression to design MLP for a given application. It can help to design the network structure, as well as the training of MLP.

1 Introduction

The sensitivity analysis of neural network has been investigated for over 30 years. In 1962, Hoff used n -dimensional geometry approach to analyze the sensitivity of Adaline [Hoff, 1962]. Glanz [1965] simplified Hoff's results. After two decades, Winter [1989] derived an analytical expression for the probability of error in Madaline caused by weight perturbation. Stevenson continued Winter's work, and established the sensitivity of Madaline to weight error in 1990 [Stevenson, 1990a; Stevenson *et al.*, 1990b]. Cheng and Yeung [1999] use this geometrical arguments to analyze the sensitivity of Neocognitrons. However, their results are only fit for neurons with threshold activation function and binary inputs, and they are not applicable to other continuous activation functions. Yeung and Wang [1999] try to apply the method proposed in [Cheng and Yeung, 1999] to the sensitivity analysis of the MLP. Earlier in 1992, Choi and Choi [1992] present a thorough idea of sensitivity analysis of the MLP with differentiable activation functions. Although their analysis is systematic, the result is only applicable when the network

has been trained and the inputs are known, so it cannot be used for network design and construction.

Piché [1992;1995] uses statistical approach to relate the output error and the change of weights in Madaline with several activation functions such as linear, sigmoid and threshold. But Piché's stochastic model is a simplification of the true MLP, and his result holds true only when the input and weight perturbations are small enough.

This paper proposes an improved stochastic model which is better suited for the true MLP. We also present a function approximation method to derive an universal analytical expression of sensitivity for MLP with sigmoidal activation function. We assume no restriction on the amplitude of the input and weight perturbations. Using this function approximation method, one can use three parameters to characterize the activation function. So not only the effects of input and weight perturbations, the number of neuron and the number of layer, but also that of the activation function on the sensitivity of MLP can be analyzed by the universal analytical expression.

2 The Improved Stochastic Model of MLP

According to Piché's stochastic model, all neurons have the same activation function. All network inputs, weights, input perturbations and weight perturbations are respectively independent identically distributed random (iidr) variables with zero mean; inputs and input errors, inputs and weight errors, weights and inputs, weights and weight errors are statistically independent from each other. This model has the following weaknesses:

1. The bias input is omitted. This is a basic component of MLP;
2. The expectations of all network inputs are zeros. Many applications don't satisfy this condition;
3. All network inputs have the same variance. It is also difficult for many applications to satisfy this requirement;
4. The correlations between the inputs and their corresponding perturbations are not considered. Even if the inputs and their associative perturbations in the first layer are independent, there still exist correlations between the inputs and their corresponding perturbations in the successive layers of MLP.

* This research work is supported by a Hong Kong Polytechnic University research grant number G-S546 and a Hong Kong CERF number 5114/97E

Our improved stochastic model of MLP overcomes the above shortcomings in the following ways:

1. All network inputs are independent random variables with their own expectations and variances. The expectation of the bias input is 1, and its variance is 0;
2. All input perturbations are independent random variables with zero mean and their own variances. The variance of the bias input perturbation is 0. The correlations between network inputs and their corresponding perturbations are expressed by their correlation coefficients;
3. All weights and their perturbations are independent random variables with zero mean and their own variances. The correlations between weights and weight perturbations are also expressed by their correlation coefficients;
4. Except the correlations between inputs and input perturbations, weights and weight perturbations, all variables are independent from each other.

Our model is a better description of the true MLP.

3 Definitions

First, some notations are introduced. The layers are indexed from 0 to L . The 0 layer is the input layer, and the L th layer is the output layer. The number of nodes in layer l is denoted by N^l . The function $g(\cdot)$ denotes the activation function. Neuron is the basic element of MLP, so one can first investigate the sensitivity of a single neuron, and then the whole MLP.

Suppose X^l , ΔX^l are the input and input perturbation vectors of neuron i in layer l , W_i^l , ΔW_i^l are its weight and weight perturbation vectors. $D(\cdot)$ denotes the variance.

Definition 1: The absolute sensitivity of the neuron to input and weight perturbations is given by

$$S_i^l \stackrel{\text{def}}{=} \sqrt{D(g((W_i^l + \Delta W_i^l) \cdot (X^l + \Delta X^l)) - g(W_i^l \cdot X^l))}.$$

Definition 2: The relative sensitivity of the neuron to input and weight perturbations is

$$\text{Rel}.S_i^l \stackrel{\text{def}}{=} \sqrt{\frac{D(g((W_i^l + \Delta W_i^l)^T \cdot (X^l + \Delta X^l)) - g(W_i^{lT} \cdot X^l))}{D(g(W_i^{lT} \cdot X^l))}}.$$

Since the outputs of MLP are the outputs of the neurons in the output layer, we can use the output neurons to represent the outputs of the MLP.

Definition 3: The absolute sensitivity of MLP to input and weight perturbations is

$$S \stackrel{\text{def}}{=} (S_1^L, S_2^L, \dots, S_{N_L}^L)^T.$$

Definition 4: The relative sensitivity of MLP to input and weight perturbations is

$$\text{Rel}.S \stackrel{\text{def}}{=} (\text{Rel}.S_1^L, \text{Rel}.S_2^L, \dots, \text{Rel}.S_{N_L}^L)^T.$$

4 Function Approximation Approach

If the sensitivity is directly calculated by using the above

definitions, it will be very difficult to obtain an analytical expression for the sensitivity unless we restrict the amplitudes of the input and weight perturbations. But this would severely limit the range of application of our sensitivity analysis. So a function approximation approach is proposed to overcome this problem. This is based on the idea that a given class of activation functions can be characterized by a general function expression using a set of common characteristics, i.e., parameters of this expression, which should satisfy the followings:

- 1) They can be used to approximate the activation function concerned, and
- 2) They should have the form convenient for sensitivity computation.

The threshold and sigmoidal activation functions are commonly used for the MLPs. It can be characterized by three factors: its maximum value, minimum value and obliquity, which are denoted by B , C and A respectively. Different values of B , C and A represent different activation functions. The main idea of the function approximation approach is to use another function $g_{A,B,C}(x)$ to approximate $g(x)$. The function $g_{A,B,C}(x)$ has the following form

$$g_{A,B,C}(x) = \frac{B+C}{2} + f_{A,B,C}(x), \quad (1)$$

where

$$f_{A,B,C}(x) = \begin{cases} \frac{B-C}{2}(1 - e^{-Ax^2}), & x \geq 0, \\ -\frac{B-C}{2}(1 - e^{-Ax^2}), & x < 0 \end{cases}, \quad (2)$$

The objective of function approximation is to determine the three factors A , B and C of $g(x)$, so that the distance between the two functions $g(x)$ and $g_{A,B,C}(x)$ is minimized.

The factors B and C are determined by

$$B = \lim_{x \rightarrow +\infty} g(x), \quad (3)$$

$$C = \lim_{x \rightarrow -\infty} g(x). \quad (4)$$

The factor A can be found by solving the following problem

$$\min_A \sqrt{\int_{-\infty}^{+\infty} (g(x) - g_{A,B,C}(x))^2 dx}. \quad (5)$$

Because $g(x)$ and $g_{A,B,C}(x)$ are antisymmetric about a certain point on the y-axis, problem (5) is equivalent to

$$\min_A \sqrt{\int_0^{+\infty} (g(x) - g_{A,B,C}(x))^2 dx}. \quad (6)$$

The commonly used sigmoidal activation function has the form $g(x) = \frac{B + Ce^{-\lambda x}}{1 + e^{-\lambda x}}$, we have $A = \frac{\sqrt{\pi}}{4} \lambda^2$.

5 Sensitivity of a Single Neuron

According to the definitions of absolute and relative sensitivity, we need first derive the variances of the output and the output error of a neuron. The output of the unit j in

layer l is

$$y_i^l = g(W_i^{lT} \cdot X^l). \quad (7)$$

Its output error is

$$\Delta y_i^l = g((W_i^l + \Delta W_i^l)^T \cdot (X^l + \Delta X^l)) - g(W_i^{lT} \cdot X^l). \quad (8)$$

The covariances between inputs and input perturbations, weights and weight perturbations are $\sigma_{x_i \Delta x_i}^l = \rho_{x_i \Delta x_i}^l \sigma_{x_i}^l \sigma_{\Delta x_i}^l$ and $\sigma_{w_{ij} \Delta w_{ij}}^l = \rho_{w_{ij} \Delta w_{ij}}^l \sigma_{w_{ij}}^l \sigma_{\Delta w_{ij}}^l$, where ρ denotes the correlation coefficient.

Let σ^2 and μ represent variance and expectation. We have

$$E(W_i^{lT} \cdot X^l) = E(\Delta W_i^{lT} \cdot X^l + W_i^{lT} \cdot \Delta X^l + \Delta W_i^{lT} \cdot \Delta X^l) = 0, \quad (9)$$

$$D(W_i^{lT} \cdot X^l) = \sum_{j=1}^{N^{l-1}} (\sigma_{x_j}^{l2} + \mu_{x_j}^{l2}) \sigma_{w_{ij}}^{l2} + \sigma_{w_{i0}}^{l2}, \quad (10)$$

$$D(\Delta W_i^{lT} \cdot X^l + W_i^{lT} \cdot \Delta X^l + \Delta W_i^{lT} \cdot \Delta X^l), \quad (11)$$

$$= \sum_{j=1}^{N^{l-1}} ((\sigma_{x_j}^{l2} + \mu_{x_j}^{l2}) \sigma_{\Delta w_{ij}}^{l2} + \sigma_{\Delta x_j}^{l2} \sigma_{w_{ij}}^{l2} + \sigma_{\Delta x_j}^{l2} \sigma_{\Delta w_{ij}}^{l2}) \quad (12)$$

$$+ \sum_{j=1}^{N^{l-1}} (2\sigma_{x_j, \Delta x_j}^l \sigma_{w_{ij}, \Delta w_{ij}}^l + 2\sigma_{x_j, \Delta x_j}^l \sigma_{\Delta w_{ij}}^{l2} + 2\sigma_{\Delta x_j}^{l2} \sigma_{w_{ij}, \Delta w_{ij}}^l) + \sigma_{\Delta w_{i0}}^{l2}$$

Let

$$\sigma_R = \sqrt{D(W_i^{lT} \cdot X^l)}, \quad (13)$$

$$\sigma_P = \sqrt{D(\Delta W_i^{lT} \cdot X^l + W_i^{lT} \cdot \Delta X^l + \Delta W_i^{lT} \cdot \Delta X^l)}, \quad (14)$$

$$U = \frac{W_i^{lT} \cdot X^l}{\sigma_R}, \quad (15)$$

$$V = \frac{\Delta W_i^{lT} \cdot X^l + W_i^{lT} \cdot \Delta X^l + \Delta W_i^{lT} \cdot \Delta X^l}{\sigma_P}. \quad (16)$$

According to the Lindeberg Central Limit Theorem, when N^{l-1} is large enough, u and v are normally distributed with mean 0 and variance 1, i.e., $u \sim N(0,1)$ and $v \sim N(0,1)$. The correlation coefficient between u and v is

$$\rho_{U,V} = \frac{\sum_{j=1}^{N^{l-1}} (\sigma_{w_{ij}, \Delta w_{ij}}^l (\sigma_{x_j}^{l2} + \mu_{x_j}^{l2}) + \sigma_{x_j, \Delta x_j}^l \sigma_{w_{ij}}^{l2} + \sigma_{x_j, \Delta x_j}^l \sigma_{w_{ij}, \Delta w_{ij}}^l) + \sigma_{w_{i0}, \Delta w_{i0}}^l}{\sigma_R \sigma_P}. \quad (17)$$

Let

$$V = \frac{-\rho_{U,V}}{\sqrt{1-\rho_{U,V}^2}} U + \frac{1}{\sqrt{1-\rho_{U,V}^2}} V', \quad (18)$$

we get $V \sim N(0,1)$, and $\rho_{U,V} = 0$.

We introduce two notations

$$\alpha = \sigma_R + \sigma_P \rho_{U,V}, \quad (19)$$

$$\beta = \sigma_P \sqrt{1-\rho_{U,V}^2}. \quad (20)$$

Then

$$W_i^{lT} \cdot X^l = \sigma_R U, \quad (21)$$

$$(W_i^l + \Delta W_i^l)^T \cdot (X^l + \Delta X^l) = \alpha U + \beta V. \quad (22)$$

The distribution function of u , and the joint distribution function of u and v may be expressed as

$$f_U(u) = \frac{1}{2\pi} e^{-\frac{1}{2}u^2}, \quad (23)$$

$$f(u, v) = \frac{1}{2\pi} e^{-\frac{1}{2}(u^2+v^2)}. \quad (24)$$

Now, the expectations of output and output error are

$$E(y_j^l) = \int_{-\infty}^{+\infty} g(\alpha u) \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du, \quad (25)$$

$$E(\Delta y_j^l) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (g(\alpha u + \alpha \beta v) - g(\alpha u)) \cdot \frac{1}{2\pi} e^{-\frac{1}{2}(u^2+v^2)} dudv \quad (26)$$

The variances of output and output error are

$$D(y_j^l) = \int_{-\infty}^{+\infty} g^2(\alpha u) \cdot \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} du, \quad (27)$$

$$D(\Delta y_j^l) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (g(\alpha u + \alpha \beta v) - g(\alpha u))^2 \cdot \frac{1}{2\pi} e^{-\frac{1}{2}(u^2+v^2)} dudv. \quad (28)$$

Using our function approximation approach and the following polar coordinate transformation

$$\begin{cases} u = r \cos \theta \\ v = r \sin \theta \end{cases}, r \geq 0, \theta \in [0, 2\pi], \quad (29)$$

Formulae (25), (26), (27) and (28) become

$$E y_i^l = \frac{B+C}{2}, \quad (30)$$

$$E \Delta y_i^l = 0, \quad (31)$$

$$D y_i^l = \frac{(B-C)^2}{4} \left(1 - \frac{2}{\sqrt{2A\sigma_R^2+1}} + \frac{1}{\sqrt{4A\sigma_R^2+1}} \right), \quad (32)$$

$$D \Delta y_i^l = \frac{(B-C)^2}{4} P_i^l. \quad (33)$$

According to Definitions 1 and 2,

$$S_i^l = \frac{B-C}{2} \sqrt{P_i^l}, \quad (34)$$

$$\text{Rel} S_i^l = \sqrt{\frac{P_i^l}{(1 - \frac{2}{\sqrt{2A\sigma_R^2+1}} + \frac{1}{\sqrt{4A\sigma_R^2+1}})}}, \quad (35)$$

where, let $T =$

$$\begin{aligned} & \frac{[A(\alpha^2 + \beta^2 + \sigma_R^2) + 1] \lg(\frac{1}{2} \arctg(\frac{\alpha^2 + \sigma_R^2 - \beta^2}{2\alpha\beta}) + \arctg(\frac{\beta}{\alpha})) - A\sqrt{(\alpha^2 + \sigma_R^2 - \beta^2)^2 + 4\alpha^2\beta^2}}{(\arctg(\frac{[A(\alpha^2 + \beta^2 + \sigma_R^2) + 1] \lg(\frac{1}{2} \arctg(\frac{\alpha^2 + \sigma_R^2 - \beta^2}{2\alpha\beta}) + \arctg(\frac{\beta}{\alpha})) - A\sqrt{(\alpha^2 + \sigma_R^2 - \beta^2)^2 + 4\alpha^2\beta^2}}{\sqrt{4A^2\sigma_R^2\beta^2 + 2A\alpha^2 + 2A\beta^2 + 2A\sigma_R^2 + 1}})} \\ & - \arctg(\frac{[A(\alpha^2 + \beta^2 + \sigma_R^2) + 1] \lg(\frac{1}{2} \arctg(\frac{\alpha^2 + \sigma_R^2 - \beta^2}{2\alpha\beta}) + \arctg(\frac{\beta}{\alpha})) - A\sqrt{(\alpha^2 + \sigma_R^2 - \beta^2)^2 + 4\alpha^2\beta^2}}{\sqrt{4A^2\sigma_R^2\beta^2 + 2A\alpha^2 + 2A\beta^2 + 2A\sigma_R^2 + 1}})}, \\ & P_i^l = \frac{4}{\pi} \arctg(\frac{\beta}{\alpha}) + \frac{1}{\sqrt{4A\sigma_R^2+1}} + \frac{1}{\sqrt{4A(\alpha^2 + \beta^2)+1}} \\ & - \frac{2}{\sqrt{4A^2\sigma_R^2\beta^2 + 2A\alpha^2 + 2A\beta^2 + 2A\sigma_R^2 + 1}} \\ & - \frac{4}{\pi\sqrt{2A\sigma_R^2+1}} \arctg(\frac{\beta\sqrt{2A\sigma_R^2+1}}{\alpha}) \\ & - \frac{4}{\pi\sqrt{2A\alpha^2 + 2A\beta^2 + 1}} \arctg(\frac{\beta\sqrt{2A\alpha^2 + 2A\beta^2 + 1}}{\alpha}) \end{aligned}$$

$$+ \frac{4T}{\pi \sqrt{4A^2\sigma_R^2\beta^2 + 2A\alpha^2 + 2A\beta^2 + 2A\sigma_R^2 + 1}} \quad (36)$$

For each neuron in a given layer, its inputs and the associated errors are just the outputs and their associated errors of the former layer. So $y_j^{l-1} = x_j^l$, $\Delta y_j^{l-1} = \Delta x_j^l$. Now the covariance between y_j^{l-1} and Δy_j^{l-1} can be calculated by

$$\sigma_{x_i \Delta x_i}^{l+1} = \sigma_{y_i \Delta y_i}^l = \frac{(B-C)^2}{4} Q_i^l, \quad (37)$$

where

$$\begin{aligned} Q_i^l = & -\frac{2}{\pi} \arctg\left(\frac{\beta}{\alpha}\right) + \frac{1}{\sqrt{2A\sigma_R^2 + 1}} - \frac{1}{\sqrt{4A\sigma_R^2 + 1}} - \frac{1}{\sqrt{2A(\alpha^2 + \beta^2) + 1}} \\ & + \frac{1}{\sqrt{4A^2\sigma_R^2\beta^2 + 2A\alpha^2 + 2A\beta^2 + 2A\sigma_R^2 + 1}} \\ & + \frac{2}{\pi \sqrt{2A\sigma_R^2 + 1}} \arctg\left(\frac{\beta \sqrt{2A\sigma_R^2 + 1}}{\alpha}\right) \\ & + \frac{2}{\pi \sqrt{2A\alpha^2 + 2A\beta^2 + 1}} \arctg\left(\frac{\beta \sqrt{2A\alpha^2 + 2A\beta^2 + 1}}{\alpha}\right) \\ & - \frac{2T}{\pi \sqrt{4A^2\sigma_R^2\beta^2 + 2A\alpha^2 + 2A\beta^2 + 2A\sigma_R^2 + 1}}. \end{aligned} \quad (38)$$

All formulae are derived exactly except the approximation of the activation function. So there is no need for us to restrict the input and weight perturbations to be very small. The network sensitivity expressions are expressed in terms of A , B and C , i.e., the activation function itself. This enables us to investigate the effect of a given activation function on the network sensitivity.

6 Sensitivity of MLP

Since the sensitivity of a MLP is defined to be the sensitivity vector of its output neurons, it will be necessary to calculate the sensitivity of neurons from the input layer to the output layer successively.

The algorithm to calculate the sensitivity of the network is given by

- 1) Based on the application requirement, determine the expectations and variances of network inputs $\mu_{x_i}^0, \sigma_{x_i}^{0^2}$, input perturbation ratios $\frac{\sigma_{\Delta x_i}^{0^2}}{\sigma_{x_i}^{0^2}}$, the correlation coefficients between inputs and input perturbations $\rho_{x_i \Delta x_i}^0$, the variances of weights $\sigma_{w_{ij}}^{l^2}$, weight perturbation ratios $\frac{\sigma_{\Delta w_{ij}}^{l^2}}{\sigma_{w_{ij}}^{l^2}}$ and the correlation coefficients between weights and weight perturbations $\rho_{w_{ij} \Delta w_{ij}}^l$;
- 2) Using function approximation approach to determine A , B and C of the activation function;
- 3) Let $l=1$, $\sigma_x^{l^2} = \sigma_x^2$, $\sigma_{\Delta x}^{l^2} = \sigma_{\Delta x}^2$;

- 4) For all neurons in layer l , calculate $\sigma_{y_i}^{l^2}$ and $\sigma_{\Delta y_i}^{l^2}$ by applying formulae (32) and (33), and then $\sigma_{y_i \Delta y_i}^l$, by Formula (39);
- 5) If $l < L$, let $l = l+1$, $\sigma_{x_i}^{l^2} = \sigma_{y_i}^{l-1^2}$, $\sigma_{\Delta x_i}^{l^2} = \sigma_{\Delta y_i}^{l-1^2}$, $\sigma_{x_i \Delta x_i}^l = \sigma_{y_i \Delta y_i}^{l-1}$, then goto step (4); else
- 6) Calculate S_i^L and $\text{Rel} S_i^L$ by formulae (34) and (35), then the absolute and relative sensitivities of MLP is $S = (S_1^L, S_2^L, \dots, S_{N_i}^L)^T$, $\text{Rel} S = (\text{Rel} S_1^L, \text{Rel} S_2^L, \dots, \text{Rel} S_{N_i}^L)^T$, and the program is terminated.

7 Effects of Parameters on Sensitivity

Many factors affect the sensitivity of the MLP. These factors include the network inputs and input perturbations, weights and weight perturbations, and network architecture parameters such as the number of layer, the number of neurons in each layer and the activation function chosen. How these factors affect the sensitivity of MLP is an important criteria for neural network design in support of a particular application.

The experimental results are shown in figures 1 to 13. The main results are:

- 1) The input and weight perturbation ratios are the two most important factors affecting the sensitivity of the MLP. The absolute and relative sensitivities increase when these two ratios do.
- 2) The network sensitivities increase with the absolute value of the input expectation and the input variance. The normalization of network input can help us decrease the network sensitivity;
- 3) The increase of weight variance causes the network sensitivity to increase. So we should limit the amplitude of the weight in training process;
- 4) When the number of neurons increases, the absolute and relative sensitivities increase, but when it is significantly large, the two sensitivities will nearly remain constant. The neuron number of later layers has more influence upon the absolute sensitivity than that of the previous layer.
- 5) The absolute and relative sensitivities increase with the number of layer. When the number of layer is small, these two sensitivities increase rapidly. Otherwise they become flat. When the weight perturbation ratio increases, the absolute and relative sensitivities increase as well, i.e., if the sensitivity is supposed to be less than a certain value, the number of layer could be small when the weight perturbation ratio is large.
- 6) The larger the obliquity A of the activation function is, the more sensitive the MLP is. The sensitivities increase with $B-C$. So the threshold MLP is the most sensitive one, and The activation function $\frac{1}{1+e^{-x}}$ is more suitable than $\tanh(x)$.

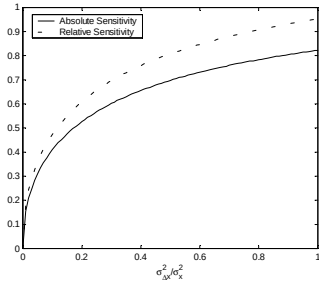


Figure 1: The absolute and relative Sensitivities of MLP versus input perturbation ratio.

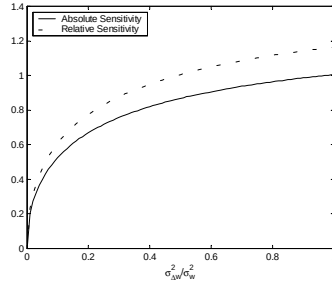


Figure 2: The absolute and relative Sensitivities of MLP versus the weight perturbation ratio.

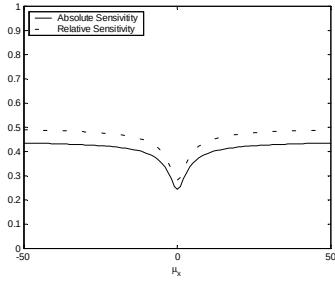


Figure 3: The absolute and relative Sensitivities of MLP versus the expectation of network input.

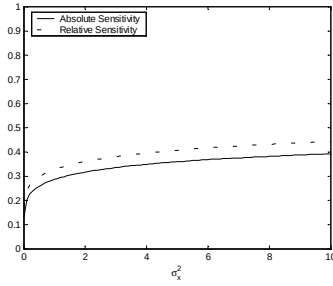


Figure 4: The absolute and relative Sensitivities of MLP versus the variance of network input.

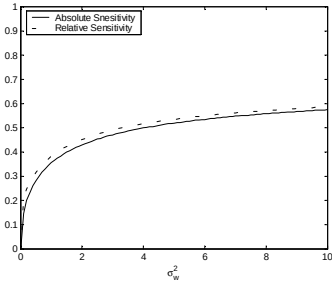


Figure 5: The absolute and relative Sensitivities of MLP versus the weight variance.

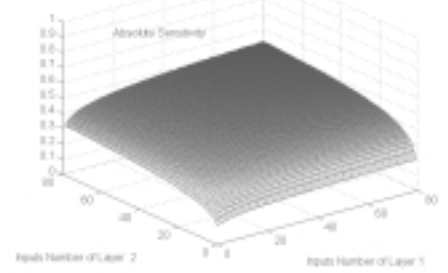


Figure 6: The absolute sensitivity of MLP versus the number of neurons in each layer.

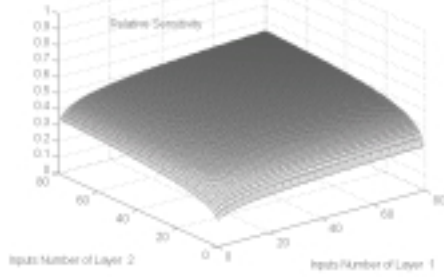


Figure 7: The relative sensitivity of MLP versus the number of neurons in each layer.

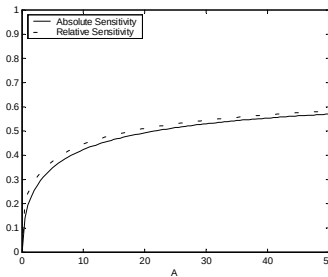


Figure 8: The absolute and relative sensitivities of a neuron versus obliquity A of the activation function.

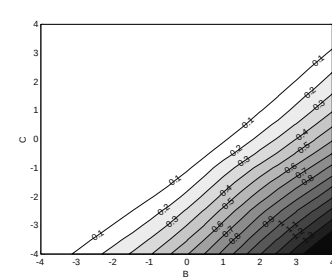


Figure 9: The absolute sensitivity of a neuron versus the max and min value of the activation function.

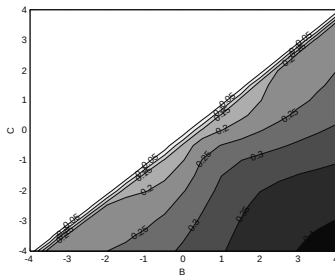


Figure 10: The relative sensitivity of a neuron versus the max and min value of the activation function.

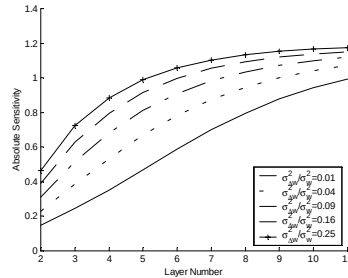


Figure 11: The absolute sensitivity of MLP versus the number of layer.

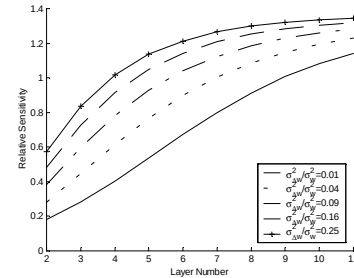


Figure 12: The relative sensitivity of MLP versus the number of layer.

8 The Design of MLP

In order to simplify the network design, we suppose that all weights have the same variances, and all weight perturbation ratios are identical. So all neurons in the same

layer have the same sensitivity. We can use a output neuron's sensitivity to represent the network sensitivity vector.

For a given application, suppose parameters L , B and C are determined by the network designer, from the sample set

and the application environment, we can estimate the following parameters: N_0 , N_L , $\mu_{x_i}^0$, $\sigma_{x_i}^{0^2}$, $\rho_{x_i, \Delta x_i}^0$, $\frac{\sigma_{\Delta x_i}^{0^2}}{\sigma_{x_i}^{0^2}}$,

$\rho_{w, \Delta w}$, $\frac{\sigma_{\Delta w}^2}{\sigma_w^2}$. Our goal is to find the unknown parameters

$N_1, \dots, N_{L-1}$, A and σ_w^2 , such that $S_R < S_R^*$, where S_R^* is a given, acceptable relative sensitivity.

Our solution is: first select a possible range for each of the unknown parameters; then using our sensitivity calculation algorithm to compute the mapping from $\{N_1, \dots, N_{L-1}, A, \sigma_w^2\}$ to $\{S_R\}$; finally, according to the mapping, determine the region $\{N_1, \dots, N_{L-1}, A, \sigma_w^2\}$ that satisfies $S_R < S_R^*$.

We present the following example to show how to design a three-layer MLP.

Example: The input and output numbers are 10, and all inputs are independent random variables that are uniformly distributed over $[-1, 1]$. Suppose the correlation coefficient between weights and weight perturbations are 0.01. Construct a three-layer MLP with input and weight perturbation ratios less than 1%, and the relative sensitivity of this MLP should be less than 10%.

For this example, the known parameters are $L=2$, $N_0 = N_2 = 10$, $\mu_{x_i}^0 = 0$, $\sigma_{x_i}^{0^2} = 4/12$, $\rho_{x_i, \Delta x_i}^0 = 0$,

$\rho_{w, \Delta w} = 0.01$, $\frac{\sigma_{\Delta x_i}^{0^2}}{\sigma_{x_i}^{0^2}} = \frac{\sigma_{\Delta w}^2}{\sigma_w^2} = 0.01$, $S_R^* = 0.1$. Suppose $B=1$,

$C=0$, and the possible ranges for parameters A , N_1 and σ_w^2 are $[0, 1.7725]$, $[1, 100]$ and $[0, 25]$ respectively. Using our sensitivity algorithm to calculate the mapping over the above ranges. Here, we plot two contour maps of S_R . When the obliquity A is large and small, the contour lines of S_R with respect to N_1 and σ_w are shown in figures 13 and 14 respectively. The region below the contour line at $S_R = 0.1$ is our design result.

Our sensitivity analysis can be used to determine the network structure numerically. Furthermore, it can estimate the permitted weight range for network training. We find that the larger N_1 is, the smaller the maximum permitted weight variance is. That is to say, if we select larger number of neurons in the hidden layer, we must restrict the weight variance to be less in the training of MLP; when the obliquity of activation function increases, the solution region reduces. This means that if we select activation function with smaller obliquity, we can permit larger number of neurons in the hidden layer and larger weight variance in the network training.

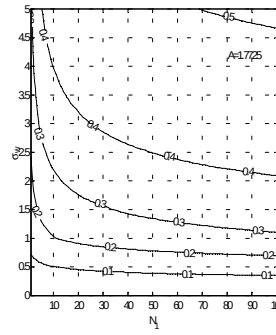


Figure 13: The contour plot of relative sensitivity of three-layer MLP with respect to N_1 and σ_w . $A=1.7725$.

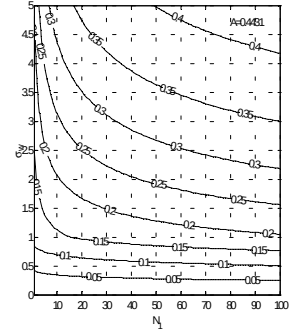


Figure 14: The contour plot of relative sensitivity of three-layer MLP with respect to N_1 and σ_w . $A=0.4431$.

9 Conclusion

In this paper an improved stochastic model of MLP is proposed to better describe the, and the function approximation method is used to obtain an universal expression of the network sensitivity for sigmoidal activation functions. We use the expression of sensitivity to investigate the effects of network parameters on the network sensitivity. It is also used to design the network architecture and provide useful guideline for the network training.

References

- [Hoff, 1962] M. E. Hoff, Learning phenomena in network of adaptive switching circuits. Stanford Electron. Labs., Stanford, CA, Tech. Rept. 1554-1, July 1962.
- [Glanz, 1965] F. H. Glanz. Statistical extrapolation in certain adaptive pattern recognition systems. Stanford Electron. Labs., Stanford, CA, Tech. Rept. 6767-1, May 1965.
- [Winter, 1989] R. G. Winter. Madaline rule II: A new method for training network s of Adalines. Stanford University, Stanford, CA, Ph.D. dissertation, Jan. 1989.
- [Stevenson, 1990a] M. Stevenson. Sensitivity of Madalines to weight errors. Stanford University, Stanford CA, Ph.D. dissertation, Dec. 1990.
- [Stevenson et al., 1990b] M. Stevenson, R. Winter, and B. Widrow. Sensitivity of feedforward neural networks to weight errors. *IEEE Trans. Neural Networks*, 1(1):71-80, 1990.
- [Cheng and Yeung, 1999] Antony Y. Cheng, D.S. Yeung/ Sensitivity Analysis of Neocognitron. *IEEE Trans. Syst., Man, And Cyber.*, 29(2):238-248, May 1999.
- [Yeung and Wang, 1999] D.S. Yeung, X.Z. Wang, Initial analysis on sensitivity of multilayer perceptron. *IEEE SMC '99 Conference Proceedings*, 3:407–411, 1999.
- [Choi and Choi, 1992] J. Y. Choi and C. -H. Choi. Sensitivity analysis of multilayer perceptron with differentiable activation functions. *IEEE Trans. Neural Networks*, 3:101-107, Jan. 1992.
- [Piché, 1992] S. Piché. Selection of weight accuracies for neural networks. Stanford University, Stanford, CA, Ph.D., dissertation, May 1992.
- [Piché, 1995] Stephen W. Piché. The selection of weight accuracies for Madalines. *IEEE Trans. Neural Networks*, 6(2):432-445, 1995