Eighth International Conference on Intelligent Systems Design and Applications

Equivalence between Weight Decay Learning and Explicit Regularization to Improve Fault Tolerance of RBF

John Sum[1], Wun-He Luo[2], Yung-Fa Huang[2] and You-Ting Jheng[2]

[1]*Graduate Institute of Electronic Commerce*
National Chung Hsing University, Taichung 402, Taiwan
pfsum@nchu.edu.tw
[2]*Graduate Institute of Networking and Communication Engineering*
Chaoyang University of Technology, Taichung 41349, Taiwan
s9530608@cyut.edu.tw

Abstract

Although weight decay learning has been proposed to improve generalization ability of a neural network, many simulated studies have demonstrated that it is able to improve fault tolerance. To explain the underlying reason, this paper presents an analytical result showing the equivalence between adding weight decay and adding explicit regularization on training a RBF to tolerate multiplicative weight noise. Under a mild condition, it is proved that explicit regularization will be reduced to weight decay.

Index Terms - Weight decay, Explicit regularization, Radial basis function (RBF), Multiplicative weight noise (MWN).

1. Introduction

In the process of knowledge acquisition, one can use the deviation between network output and target output to adjust the weights of a neural network (NN) [1]. For more than a decade, analysis on fault tolerance of a NN has been a major topic in the research community [2], [3], [4], [5], [10], [11]. A fault tolerant neural network is able to persistently working whenever the hardware or software has error/fault. Two kinds of faults can usually be found while a NN is implemented by FPGA technologies. They are node dead [10], [12], [14] and weight noise (additive or multiplicative). This paper is focusing on MWN [8], [9], [11], [13], [15].

Weight decay learning algorithm [6] is the same as other learning algorithms. In which, weights are adjusted to get good generalization. But, determining the decay constant so as to get best performance is a difficult problem. Without much information about the properties of the dataset, it will need large computational and memory resource to try every possible value for the decay constant in order to dig out the best model.

The rest of this paper is organized as follows. In Section 2, the model of an RBF is defined. Then, the model for multiplicative weight noise is stated in Section 3. In the same section, two variants of RBF models will be defined as well. In Section 4, the WDL [6] and ERL [8] will be reviewed and their regularization terms are highlighted. Their equivalence is proved in Section 5. Besides, the setting of the decay constant in terms of weight noise variance and the width of basis functions are derived for one-dimensional and two-dimensional problems. Finally, the conclusion of the paper is given in Section 6.

2. Search problems

Fig. 1 shows that a system by measured datum.

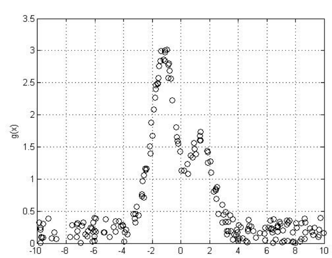

Fig. 1. The measured datum.

Fig. 2 shows that input x is transformed into $g(x)$ and output y is obtained by adding $g(x)$ with external noise. If $g(x)$ and the external noise are known, it is able to acquire the behavior of this system. But, $g(x)$ is hard to know. The researchers often assumed that $g(x)$ is a nonlinear model, such as a fuzzy system and a neural network, and using measured datum to estimate the parameter of the model. In this paper, the definition $g(x)$ is a RBF

978-0-7695-3382-7/08 $25.00 © 2008 IEEE
DOI 10.1109/ISDA.2008.146

network [1].

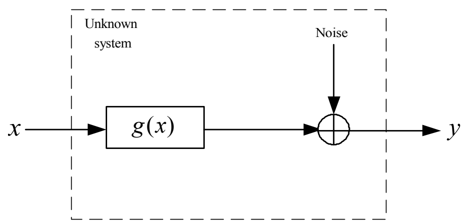

Fig. 2. Schematic diagram of the simulation environment.

In this paper, we have to solve a problem that the predicted error needs to reduce as the weight suffered different margin oscillation. In Fig. 3 shows that simulation environment. At first, we have two situations about finding a suitable model. The former, input x, weight $\hat{w}$ and output $\hat{y}$ are without adding any elements. The latter, output $\tilde{y}$ is including special factors x and $\tilde{w}$.

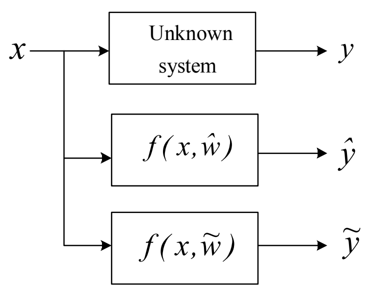

Fig. 3. Analysis the problem of simulation environment.

3. Multiplicative weight noise

A RBF NN is belonging to feedforward neural network (FNN) and it formed input layer, hidden layer and output layer. Purposing of the concept are constructing many neural nodes and using curve fitting method to find input-output mapping. A RBF $\phi(x)$ of neural node that can be represented as follows:

$$\phi(x) = \exp(-\frac{(x-c)^2}{\sigma}) \tag{1}$$

where c is radial function center and σ is the width of a RBF. Figure 4 shows a schematic diagram of the RBF network. Including N of input vectors, M of neural nodes and single output.

In this case, the output of a RBF network is:

$$f(x) = \sum_{j=1}^{M} w_j \cdot \phi_j(x) + w_0 \tag{2}$$

where $\phi_j(x)$ are the number of basis functions, $\{w_j \mid j = 1, 2, ..., M\}$ are the synaptic weights.

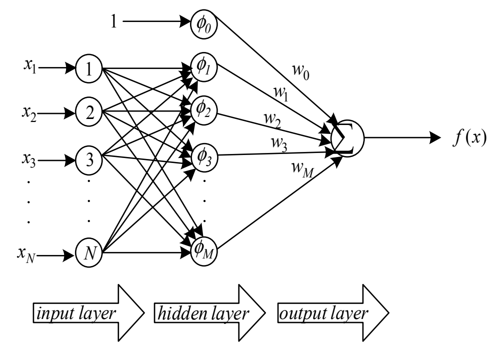

Fig. 4. Schematic diagram of the RBF network.

When a neural weight $(\hat{w}_i)$ malfunctioned $(\tilde{w}_i)$,

$$\tilde{w}_i = \hat{w}_i + s_i, \tag{3}$$

$$s_i = b_i \hat{w}_i, \tag{4}$$

where $\hat{w}_i$ is initial weight, s_i is the multiplicative weight noise and b_i is random variable of normal distribution. With $E[b_i] = 0$ and $E[b_i^2] = S_b$.

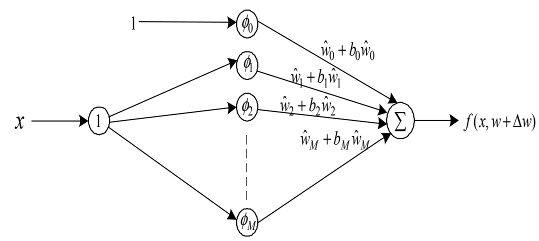

Fig. 5. Schematic diagram of MWN of RBF.

Therefore, for $\hat{w}_2 > \hat{w}_1$ the noise factor s_2 will be larger than s_1, see Fig.6.

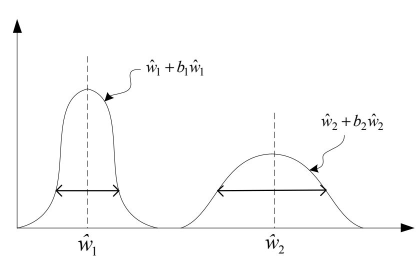

Fig. 6. Noise distribution of MWN.

4. Learning algorithms

4.1. Weight Decay Learning[6]

WDL can improve the generalization capability of a FNN. Weight decay has two nice properties. The form is that it can restrain the unsuitable elements of weight vectors. The later is that the weight can control static noise index and increase generalization ability [7]. WDL is a method that can decay a quantity of weights. WDL can decrease tiny weight to zero. Therefore, it can reduce some parameters in the network architecture. In this paper, we assumed N of inputs and outputs,

$$Data=\{(x_1,y_1),(x_2,y_2)\cdots(x_N,y_N)\} \tag{5}$$

objective function $c(w)$,

$$c(w)=\frac{1}{N}\sum_{k=1}^{N}\left(y_k-w^T\phi(x_k)\right)^2+\lambda w^T w \tag{6}$$

where y_k is output, $\phi(x_k)$ is a RBF and λ is learning rate. Assume that hidden layer has three neural nodes as ϕ_0, ϕ_1 and ϕ_2.

$$\begin{aligned} c(w)&=\frac{1}{N}\sum_{k=1}^{N}(y_k-\sum_{i=0}^{2}w_i\phi_i(x_k))^2+\lambda\sum_{i=0}^{2}w_i{}^2 \\ &=\frac{1}{N}\sum_{k=1}^{N}(y_k-\sum_{i=0}^{2}w_i\phi_i(x_k))^2+\lambda(w_0{}^2+w_1{}^2+w_2{}^2) \end{aligned} \tag{7}$$

Weight vectors that can be represented as follows:

$$\begin{bmatrix} \frac{1}{N}\sum_{k=1}^{N}y_k\phi_0(x_k) \\ \frac{1}{N}\sum_{k=1}^{N}y_k\phi_1(x_k) \\ \frac{1}{N}\sum_{k=1}^{N}y_k\phi_2(x_k) \end{bmatrix}=\begin{bmatrix} \frac{1}{N}\sum_{k=1}^{N}\phi_0(x_k)\phi_0(x_k)+\lambda & \frac{1}{N}\sum_{k=1}^{N}\phi_1(x_k)\phi_0(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_2(x_k)\phi_0(x_k) \\ \frac{1}{N}\sum_{k=1}^{N}\phi_0(x_k)\phi_1(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_1(x_k)\phi_1(x_k)+\lambda & \frac{1}{N}\sum_{k=1}^{N}\phi_2(x_k)\phi_1(x_k) \\ \frac{1}{N}\sum_{k=1}^{N}\phi_0(x_k)\phi_2(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_1(x_k)\phi_2(x_k) & \frac{1}{N}\sum_{k=1}^{N}\phi_2(x_k)\phi_2(x_k)+\lambda \end{bmatrix}\begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}, \tag{8}$$

$$w=\left[\frac{1}{N}\sum_{k=1}^{N}\phi(x_k)\phi^T(x_k)+\lambda I_{M\times M}\right]^{-1}\left[\frac{1}{N}\sum_{k=1}^{N}y_k\phi(x_k)\right] \tag{9}$$

4.2. Explicit Regularization Learning[8]

Learning algorithm are not only applying multilayer perceptron (MLP) structure but also adjusting weights for different solution schemes. Further, the maximum fault tolerant scheme on backpropagation (BP) learning algorithm was proposed. The goal of mean square error (MSE) is achieving the minimum decay. Objection function of ERL [8, 9] as follows:

$$c(w)=\frac{1}{N}\sum_{k=1}^{N}\left(y_k-w^T\phi(x_k)\right)^2+S_b w^T G w \tag{10}$$

where S_b is variance and G is a new additional term.

$$G=\frac{1}{N}\sum_{k=1}^{N}\begin{bmatrix} \phi_0^2(x_k) & 0 & \cdots & 0 \\ 0 & \phi_1^2(x_k) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \phi_M^2(x_k) \end{bmatrix} \tag{11}$$

Aggregating (10) and (11), the weight that as follows:

$$w=\left(\frac{1}{N}\sum_{k=1}^{N}\phi^T(x_k)\phi(x_k)+S_b G\right)^{-1}\left(\frac{1}{N}\sum_{k=1}^{N}y_k\phi(x_k)\right) \tag{12}$$

5. WDL=ERL ($w_o=0$)

We can use WDL and ERL on training RBF network to against MWN. However, WDL and ERL are completely different learning methods. WDL is a regularization algorithm. But ERL is aiming at MWN. Details of the proposed that fault tolerant performance of WDL=ERL on 1-D and 2-D are stated in the subsections that follow.

5.1. One-dimensional problems

We assume some research questions on 1-D. When weights were suffered margin oscillation, we need to reduce predicted error. For MWN part, we compared with WDL and ERL.

$$\begin{bmatrix} \lambda & 0 & \cdots & 0 \\ 0 & \lambda & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \lambda \end{bmatrix} \tag{13}$$

and

$$\begin{bmatrix} S_b\frac{1}{N}\sum_{k=1}^{N}\phi_0{}^2(x_k) & 0 & \cdots & 0 \\ 0 & S_b\frac{1}{N}\sum_{k=1}^{N}\phi_1{}^2(x_k) & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & S_b\frac{1}{N}\sum_{k=1}^{N}\phi_M{}^2(x_k) \end{bmatrix} \tag{14}$$

In this paper, RBF is assumed $N\to\infty$ as $x\in[-L,L]$ and $p(x)=\frac{1}{2L}$ condition can be represented as follows:

$$\frac{1}{N}\sum_{k=1}^{N}\phi_i^2(x_k)\approx\int_{-L}^{L}\phi_i^2(x)p(x)dx \tag{15}$$

$$\begin{aligned} &\int_{-L}^{L}\phi_i^2(x)p(x)dx\approx\frac{1}{2L}\int_{-\infty}^{\infty}\phi_i^2(x)dx \\ &=\frac{1}{2L}\int\exp\left(-\frac{2(x-c_i)^2}{\sigma}\right)dx \end{aligned} \tag{16}$$

Aggregating (15) and (16), a RBF is rewritten as:

$$\int \phi_1^2(x)dx = \int \phi_2^2(x)dx = \cdots = \int \phi_M^2(x)dx \qquad (17)$$

We could have similar fault tolerance as (18) on WDL and ERL.

$$\lambda = \frac{S_b}{2L}\sqrt{\frac{\pi\sigma}{2}} \qquad (18)$$

5.2. Two-dimensional problems

Besides, we use Signum function on 2-D. Moreover, input vector x_1 and x_2 through Sign function transform into 0 or 1 as following equation:

$$f(x_1, x_2) = \begin{cases} 1 & if \;\; x_1 \cdot x_2 \geq 0 \\ 0 & if \;\; x_1 \cdot x_2 < 0 \end{cases} \qquad (19)$$

In terms of (19) shows that 3-D space. It divisions into four areas by x_1 and x_2.

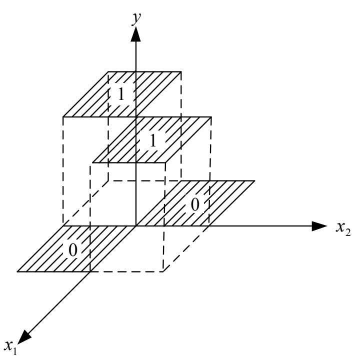

Fig. 7. Schematic diagram of 3-D space by input vector.

We assumed $N \to \infty$ on 2-D as given in (20). Input vectors (x_{k1}, x_{k2}) represent that the k th of input pattern can be defined as:

$$\begin{aligned} &If,\; \frac{S_b}{N}\sum_{k=1}^{N}\phi_i^2(x_{k1}, x_{k2}) = \frac{S_b}{N}\sum_{k=1}^{N}\phi_j^2(x_{k1}, x_{k2}) = \delta, \forall i \neq j \\ &Then,\; \lambda = \delta \end{aligned} \qquad (20)$$

By using (20) to operation integration as follows:

$$\begin{aligned} &\frac{1}{N}\sum_{k=1}^{N}\phi_i^2(x_{k1}, x_{k2}) \\ &\approx \int_{-1}^{1}\int_{-1}^{1}\phi_i^2(x_{k1}, x_{k2})p(x_{k1}, x_{k2})dx_1 dx_2 \\ &\approx \frac{1}{4}\int_{-1}^{1}\int_{-1}^{1}\phi_i^2(x_{k1}, x_{k2})dx_1 dx_2 \end{aligned} \qquad (21)$$

$$\begin{aligned} &\int \phi_i^2(x)dx = \int \exp\left(-\frac{2}{\sigma}(x-c)^T(x-c)\right)dx \\ &\int \exp\left(-\frac{1}{2}(x-\mu)^T \sum\nolimits^{-1}(x-\mu)\right)dx \\ &= (2\pi)^{N_x/2}|\Sigma|^{1/2} \end{aligned} \qquad (22)$$

5.2.1 Case I Assume that the dimension of input vector is 2($N_x = 2$) and σ is the width of neural node as follows:

$$\begin{aligned} &2\Sigma = \begin{bmatrix} \sigma & 0 \\ 0 & \sigma \end{bmatrix} \Big/ 2 \\ &\therefore |\Sigma| = \frac{1}{4}\sigma^2 \end{aligned} \qquad (23)$$

Aggregating (20) to (23), it can get a new learning rate λ by (24). By using λ can has the same fault tolerance on 1-D and 2-D.

$$\lambda = \frac{\pi}{2}\sigma^2 S_b \qquad (24)$$

5.2.2 Case II Furthermore, we consider the multivariate normal distribution. From Khintchine weak law of large numbers as follows:

$$\lim_{N\to\infty} P\left(\left|\frac{1}{N}\sum_{k=1}^{N}\phi_i^2(x_k) - E\left(\phi_i^2(x_k)\right)\right| < \varepsilon\right) = 1 \qquad (25)$$

where all $\varepsilon > 0$ and E is expectation value. When input vector is 2-D that can be represented as follows:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\phi_i^2(x_{k1}, x_{k2}) = E\left(\phi_i^2(x_1, x_2)\right) \qquad (26)$$

The expectation value of random variables can be defined as:

$$\begin{aligned} &\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\phi_i^2(x_1, x_2)p(x_1, x_2)dx_1 dx_2 \\ &= \int_{-\infty}^{\infty}\phi_i^2(\underline{x})p(\underline{x})d(\underline{x}) \end{aligned} \qquad (27)$$

where $p(\underline{x})$ is probability density function (PDF) and $\underline{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$.

In Fig. 8 shows that two vectors average distribution between $-L$ and L. Thus, we can be represented $p(\underline{x})$ as follows:

$$p(\underline{x}) = \begin{cases} \dfrac{1}{4L^2}, & -L \leq x_1 \leq L \text{ and } -L \leq x_2 \leq L \\ 0, & \text{others} \end{cases} \qquad (28)$$

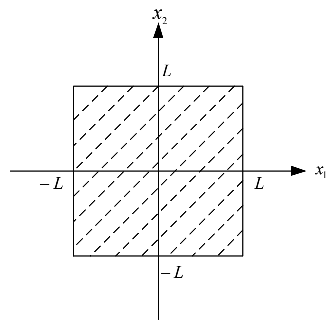

Fig. 8. Average distribution area of x_1 and x_2

Besides, we defined density function of normal variables as follows:

$$\phi(\underline{x}) = \exp\left[\frac{-1}{2}(\underline{x}-\underline{c}_i)' \Sigma^{-1}(\underline{x}-\underline{c}_i)\right] \quad (29)$$

where $\underline{c}_i$ is the i th of RBF center and Σ is variance-covariance matrix. In order to convenient, we use vector symbols to represent formula further. First, we can be represented Σ, $|\Sigma|$ and Σ^{-1} as follows:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad (30)$$

$$|\Sigma| = \sigma_1^2\sigma_2^2(1-\rho^2) \quad (31)$$

$$\Sigma^{-1} = \begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix} \Big/ \sigma_1^2\sigma_2^2(1-\rho^2) \quad (32)$$

where ρ is coefficient of x_1 and x_2. Aggregating (30) to (32), it can rewritten bivariate as follows:

$$\phi(x_1,x_2) = \exp\left\{\frac{-1}{2(1-\rho^2)}\left[\left(\frac{x_1-c_{i1}}{\sigma_1}\right)^2 + \left(\frac{x_2-c_{i2}}{\sigma_2}\right)^2 - 2\rho\left(\frac{x_1-c_{i1}}{\sigma_1}\right)\left(\frac{x_2-c_{i2}}{\sigma_2}\right)\right]\right\}, \quad (33)$$

In fact, we can easily to find the exponent party(ignore coefficient, $-\frac{1}{2}$) by (29) is equal to under equation:

$$\begin{aligned}&[x_1-c_{i1} \quad x_2-c_{i2}]\Sigma^{-1}\begin{bmatrix} x_1-c_{i1} \\ x_2-c_{i2} \end{bmatrix} \\ &= (\underline{x}-\underline{c}_i)' \Sigma^{-1}(\underline{x}-\underline{c}_i)\end{aligned} \quad (34)$$

Moreover, we take $\underline{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ and $\underline{c}_i = \begin{pmatrix} c_{i1} \\ c_{i2} \end{pmatrix}$ substitution (34) as follows:

$$\frac{1}{(1-\rho^2)}\left[\left(\frac{x_1-c_{i1}}{\sigma_1}\right)^2 + \left(\frac{x_2-c_{i2}}{\sigma_2}\right)^2 - 2\rho\left(\frac{x_1-c_{i1}}{\sigma_1}\right)\left(\frac{x_2-c_{i2}}{\sigma_2}\right)\right], \quad (35)$$

In Fig. 9 and 10 show that volumes of bivariate normal density are distributed between c_{i1} and c_{i2}. And then the direction of an ellipse is decided by Σ^{-1}.

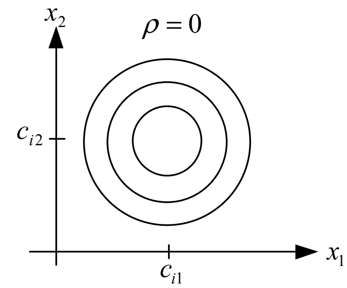

Fig. 9. The silhouette of equal-density.

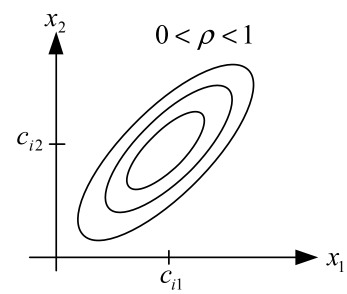

Fig. 10. The silhouette of equal-density.

In addition to all of the above, we defined $\phi^2(x)$:

$$\int_{-\infty}^{\infty}\phi^2(\underline{x})d\underline{x} = \int_{-\infty}^{\infty}\exp\left[-\frac{1}{2}(\underline{x}-\underline{c}_i)'\left(\frac{\Sigma}{2}\right)^{-1}(\underline{x}-\underline{c}_i)\right]d\underline{x} \quad (36)$$

In the statistics theorem, multivariate normal density function be defined as:

$$f(x_1,x_2) = \frac{1}{\sqrt{(2\pi)^2|\Sigma|}}\exp\left[-\frac{1}{2}(\underline{x}-\underline{\mu})' \Sigma^{-1}(\underline{x}-\underline{\mu})\right] \quad (37)$$

And joint density function of 2-D has below characteristic:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x_1,x_2)dx_1dx_2 = 1 \quad (38)$$

At last, we integration above equations to get a new term:

$$\int_{-\infty}^{\infty}\exp\left[-\frac{1}{2}(\underline{x}-\underline{c}_i)'\left(\frac{\Sigma}{2}\right)^{-1}(\underline{x}-\underline{c}_i)\right]d(\underline{x}) = \frac{1}{4L^2}2\pi|\Sigma|^{1/2} \quad (39)$$

Moreover, the diagonal elements of matrix (S_bG) will be equal to $\frac{S_b\pi}{2L^2}|\Sigma|^{1/2}$ and the coefficients of multivariate are independent as follows:

$$S_b G = \frac{S_b \pi}{2L^2} \begin{bmatrix} |\Sigma_1|^{1/2} & 0 & \cdots & 0 \\ 0 & |\Sigma_2|^{1/2} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & |\Sigma_M|^{1/2} \end{bmatrix} \quad (40)$$

Suppose $|\Sigma_1| = |\Sigma_2| = \cdots |\Sigma_M| = \mathrm{Z}$, it is able to derive that

$$\lambda = S_b G = \frac{S_b \sqrt{\mathrm{Z}}}{L^2} \times \frac{\pi}{2} \quad (41)$$

In the same way, we use new learning rate λ by (41) to train RBF with weight decay and explicit regularization also have the same fault tolerant.

6. Conclusion

This paper presented a theoretical result showing the equivalence of applying weight decay learning and explicit regularization learning for training RBF to against MWN. At particular conditions on 1-D and 2-D can achieve similar fault tolerant ability. Training a neural network by weight decay to improve fault tolerance ability has been one simple approach that has been used for more than one decade. Though, simulation results have demonstrated its success in tolerating multimode fault and multiplicative weight noise, not much theoretical work has been bone along these finding. The equivalence result presented in this paper provides direct proof to explain the use of weight decay to tolerate multiplicative weight noise.

Acknowledgement

The work is supported by a research grant from the Taiwan National Science Council under project number NSC 95-2221-E-040-009.

7. References

[1] S. Haykin, *Neural networks*, "A Comprehensive foundation," *Macmillan College Publishing Company Inc.*, 1994.

[2] A.F. Murray and P.J. Edwards, "Enhanced MLP performance and fault tolerance resulting from synaptic weight noise during training," *IEEE Transactions on Neural Networks*, vol. 5, no. 5, pp. 792-802, 1994.

[3] O. Fontenla-Romero, *et al.*, "A measure of fault tolerance for functional networks," *Neurocomputing*, vol. 62, pp. 327-347, 2004.

[4] J. Sum, "On a multiple nodes fault tolerant training for RBF: Objective function, sensitivity analysis and relation to generalization," *Proceedings of TAAI'05*, Tainan, ROC, 2005.

[5] C.S. Leung and J. Sum, "A fault tolerant regularizer for RBF networks," *IEEE Transactions on Neural Networks*, vol. 19, pp. 493-507, 2008.

[6] J.E. Moody, "Note on generalization, regularization, and architecture selection in nonlinear learning systems," *First IEEE-SP Workshop on Neural Networks for Signal Processing*, 1991.

[7] C. Bishop, "*Neural Networks for Pattern Recognition*," *Oxford University Press*, 1995.

[8] J.L. Bernier *et al.*, "Obtaining fault tolerant multilayer perceptrons using an explicit regularization," *Neural Processing Letters*, vol. 12, pp. 107-113, 2000.

[9] J. Sum, *et al.*, "On objective function, regularizer and prediction error of a learning algorithm for dealing with multiplicative weight noise," accepted for publication in *IEEE Transactions on Neural Networks.*

[10] J. Sum, *et al*., "On prediction error of a fault tolerant neural network," *Neural Information Processing*, Springer LNCS no. 4232, pp. 521-528, 2006. Spring-Verlag Berlin Heidelberg.

[11] C.S. Leung and P.f. Sum, "Analysis on Bidirectional Associative Memories with multiplicative weight noise," *Neural Information Processing*, Springer LNCS, 2007. Spring-Verlag Berlin Heidelberg.

[12] J. Sum, "Towards an objective function based framework for fault tolerant learning," in *Proceedings of TAAI'2007*.

[13] W.H. Luo, J. Sum, Y.F. Huang, "Empirical study of a fault tolerant RBF," in *Proceedings of TAAI'2007*.

[14] J. Sum, C.S. Leung, L.P. Hsu, Y.F. Huang, "An objective function for single node fault RBF learning," in *Proceedings of TAAI'2007*.

[15] J. Sum, C.S. Leung and L. Hsu, "Fault tolerant learning using Kullback-Leibler Divergence," in *Proceedings of TENCON'2007* Taipei.