Convergence and Objective Functions of Some Fault/Noise-Injection-Based Online Learning Algorithms for RBF Networks

Kevin I.-J. Ho, Chi-Sing Leung, *Member, IEEE*, and John Sum, *Senior Member, IEEE*

***Abstract*—In the last two decades, many online fault/noise injection algorithms have been developed to attain a fault tolerant neural network. However, not much theoretical works related to their convergence and objective functions have been reported. This paper studies six common fault/noise-injection-based online learning algorithms for radial basis function (RBF) networks, namely 1) injecting additive input noise, 2) injecting additive/multiplicative weight noise, 3) injecting multiplicative node noise, 4) injecting multiweight fault (random disconnection of weights), 5) injecting multinode fault during training, and 6) weight decay with injecting multinode fault. Based on the Gladyshev theorem, we show that the convergence of these six online algorithms is almost sure. Moreover, their true objective functions being minimized are derived. For injecting additive input noise during training, the objective function is identical to that of the Tikhonov regularizer approach. For injecting additive/multiplicative weight noise during training, the objective function is the simple mean square training error. Thus, injecting additive/multiplicative weight noise during training cannot improve the fault tolerance of an RBF network. Similar to injective additive input noise, the objective functions of other fault/noise-injection-based online algorithms contain a mean square error term and a specialized regularization term.**

***Index Terms*—Convergence, gladyshev theorem, fault tolerance, objective functions, RBF Networks.**

I. Introduction

REGULARIZATION [28], [31], [32], [43] and pruning [18], [24], [26], [27], [36], [39] are common techniques to attain a neural network with good generalization. These techniques work well under the assumption that neural networks after training can be ideally implemented (i.e., fault-free implementation). However, in electronic implementations (like field-programmable gate array (FPGA) [19]), component failure, sign bit change, open circuit [38], finite precision [41], and exposure to radiation [48] will exist. If special care is not considered, the performance of a neural network could degrade drastically.

There are various methods aiming at attaining a fault tolerant neural network. Some of these methods include injecting random node fault (stuck-at-zero fault, for instance) during training [7], [42], applying weight decay learning [29], [12], injecting weight noise during training [14], [33], [34], introducing network redundancy [38], formulating the training algorithm as a nonlinear constraint optimization problem [13], [35], hard-bounding weight magnitude during training [10], [17], [22], and regularization [2], [4], [5], [25], [45]. Readers may refer to [11], [37], and [47] for a summary of those techniques.

Among those methods, injecting fault or noise during training is a simple and yet effective method to improve the fault tolerance and generalization [11], [31], [34], [37], [49]. Sequin and Clay [42] and Bolt [7] are pioneers who proposed injecting random node fault during training for improving the fault tolerance of multilayer perceptron (MLPs). Murray and Edward [33], [34] proposed injecting weight noise during training. They experimentally showed that a resultant MLP is able to tolerate weight fault and multiplicative weight noise, and that the convergence of training is improved. Jim *et al.* [21] applied the similar idea for recurrent neural networks (RNNs) with real-time recurrent learning (RTRL). Apart from the concept of node fault or weight noise injection, injecting input noise during training is another approach [6], [30], [20], in which random noise is injected to the input of a network during training.

Although many online fault/noise injection learning algorithms have been developed, not many theoretical works have been investigated. Most of them focused on the output sensitivity or the prediction error of a trained neural network. Analysis on the convergence of these online fault/noise injection learning algorithms is scarce. Many researchers have claimed that training with noise is equivalent to the Tikhonov regularization [1], [6], [16], [40]. In fact, they only showed that, when the input of a well-trained neural network is corrupted by additive noise, its prediction error is equivalent to mean square error plus a Tikhonov regularizer. For the case of injecting weight noise, similar result has been obtained in [1], [3]–[5], and [45]. That is, if weights of a well-trained network are corrupted by weight noise, its prediction error is equivalent to mean square error plus a regularizer. Here, we should point out that the *prediction error* of a neural network being injected by noise is different from the *objective function* of a noise injection learning algorithm. The former one does not need the information regarding to the learning algorithm. The latter one requires to consider learning algorithm.

Manuscript received May 24, 2009; revised October 31, 2009 and March 09, 2010; accepted March 12, 2010. Date of publication April 12, 2010; date of current version June 03, 2010. This work was supported by the Taiwan NSC Research under Grant 97-2221-E-005-050 and by the General Research Funds from Hong Kong Government under Project CityU 116508.

K. I.-J. Ho is with the Department of Computer Science and Communication Engineering, Providence University, Sha-Lu 433, Taiwan (e-mail: ho@pu.edu.tw).

C.-S. Leung is with the Department of Electronic Engineering, The City University of Hong Kong, Kowloon, Hong Kong (e-mail: eeleungc@cityu.edu.hk).

J. Sum was with the Department of Electronic Engineering, The City University of Hong Kong, Kowloon, Hong Kong. He is now with the Institute of Electronic Commerce, National Chung Hsing University, Taichung 40227, Taiwan (e-mail: pfsum@nchu.edu.tw).

Digital Object Identifier 10.1109/TNN.2010.2046179

1045-9227/$26.00 © 2010 IEEE

An [1] presented the objective functions for the online backpropagation training with *additive* weight noise injection (see [1, Sec. 4]), in which the noise is defined as either a mean zero Gaussian noise or a mean zero uniform noise. However, An's approach is based on the derivation of the prediction error. Even though An applied a theorem from stochastic gradient descent [8], [9], the proof on the convergence of the online weight noise injection learning has not been accomplished. We will show later in this paper that a penalty term [1, eq. (4.7)] derived by An for weight noise injection during training is not totally correct.

As analysis on the objective functions and convergence of those online noise/fault injection learning is far from complete, further investigation along this line is inevitable. The goals of this paper are: 1) to prove convergence of fault/noise-injection-based online learning algorithms for radial basis function (RBF) networks, 2) to derive their corresponding objective functions, and 3) to elucidate the differences and similarities among noise-injection-based learning algorithms and other online learning algorithms. This paper will investigate the following six fault/noise-injection-based online learning algorithms:

A) injecting additive input noise [6];
B) injecting additive/multiplicative weight noise [34];
C) injecting multiplicative node noise;
D) injecting multiweight fault [46];
E) injecting multinode fault [42];
F) weight decay with injecting multinode fault [12].

Instead of considering the prediction error of a trained network, we investigate the properties of the update equations of these six algorithms. Their update equations can be summarized into one general equation, given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t \mathbf{F}(\mathbf{x}_t, y_t, \mathbf{w}_t, \mathbf{b}_t) \tag{1}$$

where $\mathbf{w}_t, (\mathbf{x}_t, y_t)$, and $\mathbf{b}_t$ are the estimated weight vector, a randomly selected training data, and a random vector controlling the fault/noise at time t step, respectively. The parameter μ_t is the step size and the vector function $\mathbf{F}(\cdot,\cdot,\cdot,\cdot)$ is the update function. Each of those six algorithms has its own update function.

In this paper, the expected solution for which $\mathbf{w}_t$ converges will first be deduced. Afterwards, the convergence of $\mathbf{w}_t$ with probability one is proved by applying the Gladyshev theorem [15]. Subsequently, the corresponding objective function being minimized is devised. Notice that Algorithm C is newly proposed in this paper. It will be shown later that injecting multiplicative node noise is able to improve the fault tolerance ability of an RBF network. However, injecting multiplicative weight noise during training is not able to do so.

The paper is organized as follows. In Section II, the definition of an RBF network and fault/noise models are first introduced. Afterwards, the six fault/noise-injection-based online algorithms are mathematically defined. In Section III, we will prove that all the six fault/noise-injection-based algorithms can converge with probability one. Their convergence properties together with their objective functions will be given. Owing to highlighting of the similarities among the objective functions explored in this paper and those objective functions developed by other researchers, we will also give some comments on those six algorithms. The relationship between injecting fault to regularization-based learning algorithms is elucidated in Section IV. Finally, concluding remarks and possible future works are given in Section V.

II. Fault/Noise-Injection-Based Learning

Let $\mathcal{D} = \{(\mathbf{x}_k, y_k)\}_{k=1}^N$ be a set of measurement data obtained from an unknown system of the following form :

$$y_k = f(\mathbf{x}_k) + e_k \tag{2}$$

where $\mathbf{x}_k \in \Re^n$ is the input, $y_k \in \Re$ is the output, and e_k is a mean zero Gaussian noise with finite variance. We assume that the nonlinear function $f(\mathbf{x}_k)$ can be realized by an RBF network with N_{node} RBF nodes, given by

$$y_k = \sum_{i=1}^{N_{\text{node}}} w_i^* \phi_i(\mathbf{x}_k) + e_k\,. \tag{3}$$

The basis functions $\phi_i(\mathbf{x}_k)$'s are given by

$$\phi_i(\mathbf{x}_k) = \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{c}_i\|^2}{\sigma}\right) \tag{4}$$

where $\mathbf{c}_i \in \Re^n$ is the center of the ith basis function, and $\sigma > 0$ is the width of basis functions.

In this paper, we consider that $\mathbf{c}_i$'s and σ are fixed. Hence, an RBF network can be regarded as a linear model and (3) can be rewritten in vector form as follows:

$$y_k = \Phi(\mathbf{x}_k)^T \mathbf{w}^* + e_k \tag{5}$$

where $\Phi(\cdot) = (\phi_1(\cdot), \ldots, \phi_{N_{\text{node}}}(\cdot))^T$ and $\mathbf{w}^* = (w_1^*, \cdots, w_{N_{\text{node}}}^*)^T$. In the online least mean square (LMS) learning, the update equation is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t (y_t - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)\Phi(\mathbf{x}_t) \tag{6}$$

where μ_t (for $t \geq 1$) is the step size at time t.

A. Additive Input Noise Injection Training

In the case of injecting additive input noise, the update equation is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t (y_t - \Phi^T(\tilde{\mathbf{x}}_t)\mathbf{w}_t)\Phi(\tilde{\mathbf{x}}_t) \tag{7}$$

where $\tilde{\mathbf{x}}_t$ is a noise version of the input vector, given by

$$\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{b_t}, \qquad \mathbf{b_t} \sim \mathcal{N}(\mathbf{0}, S_b \mathbf{I}_{n\times n}). \tag{8}$$

In (7) and (8), $\mathbf{b}_t = (b_{1,t}, \ldots, b_{n,t})^T$, and $b_{i,t}$'s are independently identical zero mean Gaussian noise with variance equal to S_b.

B. Additive/Multiplicative Weight Noise Injection Training

While an RBF network is trained by the idea of weight noise injection, the update equation is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t (y_t - \Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}_t)\Phi(\mathbf{x}_t) \tag{9}$$

where the ith element $\tilde{w}_{it}$ of $\tilde{\mathbf{w}}_t$ is given by

$$\tilde{w}_{i,t} = \begin{cases} w_{i,t} + b_{i,t}, & \text{for additive noise} \\ w_{i,t} + b_{i,t} w_{i,t}, & \text{for multiplicative noise.} \end{cases} \tag{10}$$

C. Multiplicative Node Noise Injection

In the case of injecting multiplicative node noise, the update equation is given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y_t - \tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}_t)\tilde{\Phi}(\mathbf{x}_t) \tag{11}$$

where $\tilde{\Phi}(\,\cdot\,) = (\tilde{\phi}_1(\,\cdot\,), \ldots, \tilde{\phi}_{N_{\rm node}}(\,\cdot\,))^T$, and

$$\tilde{\phi}_i(\mathbf{x}_t) = (1 + b_{i,t})\phi_i(\mathbf{x}_t).$$

D. Multiweight Fault Injection Training

In the case of injecting multiweight fault, we have

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y_t - \Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}_t)\Phi(\mathbf{x}_t) \tag{12}$$

where $\tilde{\mathbf{w}}_t = (\tilde{w}_{1,t}, \ldots, \tilde{w}_{N_{\rm node},t})^T, \tilde{w}_{i,t} = (1 - \beta_{i,t})w_{i,t}$, and $\beta_{i,t}$'s are independent binary random variables with probability, given by

$$P(\beta_{i,t}) = \begin{cases} p, & \text{if } \beta_{i,t} = 1 \\ (1-p), & \text{if } \beta_{i,t} = 0 \end{cases} \qquad \forall i = 1, \ldots, N_{\rm node}. \tag{13}$$

E. Multinode Fault Injection Training

While an RBF network is trained by multinode fault injection, the update equation is

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y_t - \tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}_t)\tilde{\Phi}(\mathbf{x}_t) \tag{14}$$

where $\tilde{\Phi}(\,\cdot\,) = (\tilde{\phi}_1(\,\cdot\,), \ldots, \tilde{\phi}_{N_{\rm node}}(\,\cdot\,))^T$, and $\tilde{\phi}_i(\mathbf{x}_t) = (1 - \beta_{i,t})\phi_i(\mathbf{x}_t)$.

F. Weight Decay-Based Multinode Fault Injection Training

The case of weight decay-based multinode fault injection training is similar to that of simple multinode fault injection, except that a decay term is added. The update equation is

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t\left\{(y_t - \tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}_t)\tilde{\Phi}(\mathbf{x}_t) - \lambda\mathbf{w}_t\right\}. \tag{15}$$

III. Main Results

Theory of stochastic approximation has been developed for analyzing the convergence of recursive algorithms. Advanced theoretical works for complicated recursive algorithms are still under investigation [23]. The theorem applied in this paper is based on the Gladyshev theorem [15].

Consider a general form of recursive algorithms, given by

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t M(\mathbf{w}_t, \omega_t) \tag{16}$$

where $\mathbf{w}_t$ and $M(\mathbf{w}_t, \omega_t) \in \Re^m$ for all $t = 0, 1, 2, \ldots$. ω_t are independent identically distributed (i.i.d.) random vectors with probability density function $P(\omega)$. In the fault/noise injection algorithms, ω_t corresponds to a vector augmenting $\mathbf{x}_t, y_t$, and $\mathbf{b}_t$ (or $\boldsymbol{\beta}_t = (\beta_{1,t}, \ldots, \beta_{M,t})$).

Denoting the expectation of $M(\mathbf{w}, \omega)$ over ω as $h(\mathbf{w})$

$$h(\mathbf{w}) = \int M(\mathbf{w}, \omega)P(\omega)d\omega. \tag{17}$$

Moreover, it is assumed that $h(\mathbf{w})$ has a unique solution $\mathbf{w}^*$ such that $h(\mathbf{w}^*) = 0$. The convergence of (16) can be proved if the conditions stated in the following theorem can hold.

Theorem 1 (Gladyshev Theorem [15]): For a recursive algorithm given by (16), suppose there exist positive constants κ_1 and κ_2 such that for all $\mathbf{w} \in \Re^m$ the following conditions are satisfied.

C1) $\mu_t \geq 0, \sum_t \mu_t = \infty$ and $\sum_t \mu_t^2 < \infty$.

C2) $\inf_{\varepsilon<\|\mathbf{w}-\mathbf{w}^*\|<\varepsilon^{-1}} (\mathbf{w} - \mathbf{w}^*)^T h(\mathbf{w}) < 0, \forall\, \varepsilon > 0$.

C3) $\int \|M(\mathbf{w}, \omega)\|^2 P(\omega)d\omega \leq \kappa_1 + \kappa_2\|\mathbf{w}\|^2$.

Then, for $t \to \infty$, $\mathbf{w}_t$ converges to $\mathbf{w}^*$ with probability one.

The first condition C1) is usually satisfied because the step size μ_t could be predefined as $\frac{c}{t}$ (c is a constant). Therefore, we skip the proof of condition C1) in the rest of this section. To simplify the presentation, we let

$$\mathbf{H} = \frac{1}{N}\sum_{k=1}^{N}\Phi(\mathbf{x}_k)\Phi^T(\mathbf{x}_k), Y = \frac{1}{N}\sum_{k=1}^{N} y_k\Phi(\mathbf{x}_k)$$

$$\Gamma = \frac{1}{N}\sum_{k=1}^{N}\nabla_x\Phi(\mathbf{x}_k)\nabla_x\Phi(\mathbf{x}_k)^T$$

and

$$\mathbf{Q} = \mathbf{diag}(\mathbf{H}). \tag{18}$$

A. Additive Input Noise Injection

If the input noise variance S_b is small, the following theorem can be shown by applying the Gladyshev theorem

Theorem 2: For injecting additive input noise during training an RBF network, the weight vector $\mathbf{w}_t$ will converge with probability one to

$$\mathbf{w}^* = (\mathbf{H} + S_b\Gamma)^{-1}Y \tag{19}$$

where $\mathbf{H}, \Gamma$, and Y are given in (18). Besides, the corresponding objective function $\mathcal{L}(\mathbf{w}|\mathcal{D})$ to be minimized is given by

$$\mathcal{L}(\mathbf{w}|\mathcal{D}) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2 + S_b\mathbf{w}^T\Gamma\mathbf{w}. \tag{20}$$

Proof: For small S_b, in (7)

$$\Phi^T(\tilde{\mathbf{x}}_t) = \Phi^T(\mathbf{x}_t + \mathbf{b}_t) = (\Phi^T(\mathbf{x}_t) + \mathbf{b}_t^T\nabla_x\Phi(\mathbf{x}_t)^T).$$

Hence, the correction term

$$\tilde{\delta}_t = (y_t - \Phi^T(\tilde{\mathbf{x}}_t)\mathbf{w}_t)\Phi(\tilde{\mathbf{x}}_t)$$

in (7) can be rewritten as follows:

$$\tilde{\delta}_t = (y_t - (\Phi^T(\mathbf{x}_t) + \mathbf{b}_t^T\nabla_x\Phi(\mathbf{x}_t)^T)\mathbf{w}_t)(\Phi(\mathbf{x}_t) + \nabla_x\Phi(\mathbf{x}_t)\mathbf{b}_t). \tag{21}$$

Taking expectation of (21) with respect to $\mathbf{b}_t$ yields

$$\int \tilde{\delta}_t\, P(\mathbf{b}_t)d\,\mathbf{b}_t = (y_t - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)\Phi(\mathbf{x}_t) - S_b\nabla_x\Phi(\mathbf{x}_t)\nabla_x\Phi(\mathbf{x}_t)^T\mathbf{w}_t. \tag{22}$$

After we further take expectation of the above equation with respect to $\mathbf{x}_t$ and y_t, we have $h(\mathbf{w}_t)$, given by

$$h(\mathbf{w}_t) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}_t)\Phi(\mathbf{x}_k) - S_b\Gamma\mathbf{w}_t. \quad (23)$$

Setting $h(\mathbf{w})$ in (23) to a zero vector yields the solution $\mathbf{w}^*$, given by

$$\mathbf{w}^* = [\mathbf{H} + S_b\Gamma]^{-1}Y. \quad (24)$$

Hence, for all $\|\mathbf{w} - \mathbf{w}^*\| > 0$, we have

$$(\mathbf{w} - \mathbf{w}^*)^T h(\mathbf{w}) = (\mathbf{w} - \mathbf{w}^*)^T (Y - [\mathbf{H} + S_b\Gamma]\mathbf{w})$$

which is less than zero. Therefore, condition C2) holds.

For condition C3), we consider that

$$M(\mathbf{w}, \omega) = y_t\Phi(\tilde{\mathbf{x}}_t) - \Phi(\tilde{\mathbf{x}}_t)\Phi^T(\tilde{\mathbf{x}}_t)\mathbf{w}. \quad (25)$$

As all the elements of $\Phi(\tilde{\mathbf{x}}_t)$ are in between zero and one

$$\begin{aligned} \|M(\mathbf{w}, \omega)\|^2 &= \|y_t\Phi(\tilde{\mathbf{x}}_t) - \Phi(\tilde{\mathbf{x}}_t)\Phi^T(\tilde{\mathbf{x}}_t)\mathbf{w}\|^2 & (26) \\ &\le 2\|y_t\Phi(\tilde{\mathbf{x}}_t)\|^2 + 2\|\Phi(\tilde{\mathbf{x}}_t)\Phi^T(\tilde{\mathbf{x}}_t)\mathbf{w}\|^2. & (27) \end{aligned}$$

In the above, we use the parallelogram law to establish the inequality. Note that $\Phi(\tilde{\mathbf{x}}_t)\Phi^T(\tilde{\mathbf{x}}_t)$ is a matrix with eigenvalues 0 and $\|\Phi(\tilde{\mathbf{x}}_t)\|^2$, which is less than or equal to N_{node}. Therefore

$$\|M(\mathbf{w}, \omega)\|^2 \le 2N_{\text{node}}\, y_t^2 + 2N_{\text{node}}^2\|\mathbf{w}\|^2. \quad (28)$$

Since the right-hand side of (28) is independent of random vector $\mathbf{b}_t$

$$\int \|M(\mathbf{w}, \omega)\|^2 P(\mathbf{b}_t) d\mathbf{b}_t \le 2N_{\text{node}}\, y_t^2 + 2N_{\text{node}}^2\|\mathbf{w}\|^2.$$

Further taking the expectation of the above inequality with respect to $\mathbf{x}_t$ and y_t, one can readily show that condition C3) is satisfied. The convergence proof is completed. By the fact that $\mathbf{w}^*$ is the solution of

$$\frac{\partial}{\partial \mathbf{w}}\left\{\frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}))^2 + S_b\mathbf{w}^T\Gamma\mathbf{w}\right\} = 0$$

and $(1/N)\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2 + S_b\mathbf{w}^T\Gamma\mathbf{w}$ is in quadratic form, $\mathbf{w}^*$ is unique and the objective function is given by (20). The proof of Theorem 2 is completed. Q.E.D.

One might notice that the objective function [see (20)] obtained in Theorem 2 is the same as those derived in [1], [6], [16], and [40] for multilayer perceptron and in [2] for RBF network. But we take a different approach to come up with this objective function.

B. Weight Noise Injection

Applying the Gladyshev theorem, the following theorem can be proved for injecting multiplicative weight noise with bounded S_b.

Theorem 3: For injecting (additive or multiplicative) weight noise during training an RBF network, the weight vector $\mathbf{w}_t$ will converge with probability one to

$$\mathbf{w}^* = \mathbf{H}_\phi^{-1}Y \quad (29)$$

where $\mathbf{H}$ and Y are given in (18). Besides, the corresponding objective function to be minimized is given by

$$\mathcal{L}(\mathbf{w}|\mathcal{D}) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2. \quad (30)$$

Proof: For an RBF network that is trained by injecting multiplicative weight noise, $\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y_t - \Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}_t)\Phi(\mathbf{x}_t)$, where $\tilde{w}_{i,t} = (1 + b_{i,t})w_{i,t}$. Taking expectation of the correction term $(y_t - \Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}_t)\Phi(\mathbf{x}_t)$ with respect to $\mathbf{b}_t$, we have

$$\int (y_t - \Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}_t)\Phi(\mathbf{x}_t)P(\mathbf{b}_t)d\mathbf{b}_t = (y_t - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)\Phi(\mathbf{x}_t).$$

From the expectation of the above equation with respect to $\mathbf{x}_t$ and y_t, $h(\mathbf{w}_t)$ is given by

$$h(\mathbf{w}_t) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}_t)\Phi(\mathbf{x}_k). \quad (31)$$

Therefore, $\mathbf{w}^* = \mathbf{H}^{-1}Y$.

Next, we are going to apply the Gladyshev theorem for the convergence proof. We skip the proof of condition C1) for simplicity. Recall that we define $Y = (1)/(N)\sum_{k=1}^{N} y_k\Phi(\mathbf{x}_k)$ and $\omega = (\mathbf{x}_t, y_t, \mathbf{b}_t)$. Hence, for all $\|\mathbf{w} - \mathbf{w}^*\| > 0$, we have

$$(\mathbf{w} - \mathbf{w}^*)^T h(\mathbf{w}) = (\mathbf{w} - \mathbf{w}^*)^T (Y - \mathbf{H}\mathbf{w})$$

which is less than zero. Therefore, condition C2) holds.

For condition C3), we consider that

$$M(\mathbf{w}, \omega) = y_t\Phi(\mathbf{x}_t) - \Phi(\mathbf{x}_t)\Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}. \quad (32)$$

Since $\tilde{\mathbf{w}} = \mathbf{w} + \mathbf{b}_t \otimes \mathbf{w}$

$$M(\mathbf{w}, \omega) = y_t\Phi(\mathbf{x}_t) - \Phi(\mathbf{x}_t)\Phi^T(\mathbf{x}_t)\mathbf{w} - \Phi(\mathbf{x}_t)\Phi^T(\mathbf{x}_t)\mathbf{b}_t \otimes \mathbf{w}$$

where $\otimes$ is the element multiplication operation. That means $\mathbf{b}_t \otimes \mathbf{w} = [b_{1,t}w_1, b_{2,t}w_2, \ldots]^T$. From the parallelogram law, we have

$$\begin{aligned} &\|M(\mathbf{w}, \omega)\|^2 \\ &\quad \le 2\|y_t\Phi(\mathbf{x}_t) - \Phi(\mathbf{x}_t)\Phi^T(\mathbf{x}_t)\mathbf{w}\|^2 + 2N_{\text{node}}^2\|\mathbf{b}_t \otimes \mathbf{w}\|^2 \\ &\quad \le 4N_{\text{node}}^2 y_t^2 + 4N_{\text{node}}^2\|\mathbf{w}\|^2 + 2N_{\text{node}}^2\|\mathbf{b}_t \otimes \mathbf{w}\|^2. \quad (33) \end{aligned}$$

Since $\int \|\mathbf{b}_t \otimes \mathbf{w}\|^2 P(\mathbf{b}_t)d\mathbf{b}_t = 2S_b\|\mathbf{w}\|^2$

$$\begin{aligned} &\int \|M(\mathbf{w}, \omega)\|^2 P(\mathbf{b}_t)d\mathbf{b}_t \\ &\qquad = 4y_t^2 N_{\text{node}} + N_{\text{node}}^2(4 + 2S_b)\|\mathbf{w}\|^2. \quad (34) \end{aligned}$$

Therefore, one can readily show that condition C3) is satisfied and the proof is completed.

As the solution $\mathbf{w}^*$ is identical to the solution of

$$\frac{\partial}{\partial \mathbf{w}} \frac{1}{N} \sum_{k=1}^{N} (y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}))^2 = 0$$

and $(1/N)\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2$ is in quadratic form, $\mathbf{w}^*$ is unique and the objective function is given by (30). The proof of Theorem 3 is completed. Q.E.D.

One might notice that the objective function (30) obtained in Theorem 3 is the mean square training errors. Therefore, injecting weight noise during training an RBF network should not be able to improve its tolerance to weight noise.

C. Multiplicative Node Noise Injection

Applying the Gladyshev theorem for analyzing the properties of training an RBF network by injecting multiplicative node noise with bounded S_b, the following theorem can be proved.

Theorem 4: For injecting multiplicative node noise during training an RBF network, the weight vector $\mathbf{w}_t$ will converge with probability one to

$$\mathbf{w}^* = [\mathbf{H} + S_b\mathbf{Q})]^{-1} Y \tag{35}$$

where $\mathbf{H}, \mathbf{Q}$, and Y are given in (18). Besides, the corresponding objective function to be minimized is given by

$$\mathcal{L}(\mathbf{w}|\mathcal{D}) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)^2 + S_b \mathbf{w}^T\mathbf{Q}\mathbf{w}. \tag{36}$$

Proof: Recall that $\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y_t - \tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}_t)\tilde{\Phi}(\mathbf{x}_t)$. Taking the expectation of $(y_t - \tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}_t)\tilde{\Phi}(\mathbf{x}_t)$ with respect to $b_{i,t}, \mathbf{x}_t$ and y_t, we have

$$h(\mathbf{w}_t) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}_t)\Phi(\mathbf{x}_k) - S_b\mathbf{Q}\mathbf{w}_t. \tag{37}$$

Thus, the solution $\mathbf{w}^*$ is given by

$$\mathbf{w}^* = [\mathbf{H} + S_b\mathbf{Q}]^{-1} Y. \tag{38}$$

Hence, for all $\|\mathbf{w} - \mathbf{w}^*\| > 0$, we have

$$(\mathbf{w} - \mathbf{w}^*)^T h(\mathbf{w}) = (\mathbf{w} - \mathbf{w}^*)^T (Y - [\mathbf{H} + S_b\mathbf{Q}]\mathbf{w})$$

which is less than zero. Therefore, condition C2) holds.

For condition C3), we consider $M(\mathbf{w},\omega) = y_t\tilde{\Phi}(\mathbf{x}_t) - \tilde{\Phi}(\mathbf{x}_t)\tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}$. For any $\mathbf{b}_t$

$$\begin{aligned}\|M(\mathbf{w},\omega)\|^2 &\leq 2\|y_t\tilde{\Phi}(\mathbf{x}_t)\|^2 + 2\|\tilde{\Phi}(\mathbf{x}_t)\tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}\|^2 \\ &\leq 4\|y_t\Phi(\mathbf{x}_t)\|^2 + 4\|y_t(\mathbf{b}_t \otimes \Phi(\mathbf{x}_t))\|^2 \\ &\quad + 2\|\tilde{\Phi}(\mathbf{x}_t)\tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}\|^2. \end{aligned} \tag{39}$$

Note that

$$\int \|y_t(\mathbf{b}_t \otimes \Phi(\mathbf{x}_t))\|^2 P(\mathbf{b}_t) d\mathbf{b}_t \leq S_b y_t^2 N_{\text{node}}. \tag{40}$$

By Lemma 1, $\Upsilon = \int (\tilde{\Phi}(\mathbf{x}_t)\tilde{\Phi}^T(\mathbf{x}_t))^2 P(\mathbf{b}_t)d\mathbf{b}_t$ is a nonnegative symmetric matrix whose elements depend on $\Phi(\mathbf{x}_t)$ and S_b. As the elements in $\Phi(\mathbf{x}_t)$ are bounded by zero and one, the largest eigenvalue $\lambda_{\max}\{\Upsilon\}$ of Υ must be bounded. Then

$$\|\tilde{\Phi}(\mathbf{x}_t)\tilde{\Phi}^T(\mathbf{x}_t)\mathbf{w}\|^2 \leq \lambda_{\max}\{\Upsilon\}\|\mathbf{w}\|^2. \tag{41}$$

By (40) and (41), the expectation of $\|M(\mathbf{w},\omega)\|^2$ over $\mathbf{b}_t$ is bounded by the following inequality:

$$\int \|M(\mathbf{w},\omega)\|^2 P(\mathbf{b}_t)d\mathbf{b}_t \leq 4N_{\text{node}} y_t^2 (1+S_b) + 2\lambda_{\max}\{\Upsilon\}\|\mathbf{w}\|^2. \tag{42}$$

Further taking the expectation of the above inequality with respect to $\mathbf{x}_t$ and y_t, one can readily show that condition C3) is satisfied.

Since $\mathbf{w}^*$ is the solution of

$$\frac{\partial}{\partial \mathbf{w}}\left\{\frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}))^2 + S_b\mathbf{w}^T\mathbf{Q}\mathbf{w}\right\} = 0$$

and $(1/N)\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2 + S_b\mathbf{w}^T\mathbf{Q}\mathbf{w}$ is in the quadratic form, $\mathbf{w}^*$ is unique and the objective function is given by (36). The proof of Theorem 4 is completed. Q.E.D.

Note that this objective function is the same as the objective functions derived by Bernier *et al.* [4] and Sum *et al.* [45]. The following conclusion can be obtained from this objective function. If one would like to train an RBF to tolerate anticipated multiplicative weight noise, multiplicative node noise (instead of multiplicative weight noise) should be injected during training.

D. Multiweight Fault Injection Training

applying the Gladyshev theorem for analyzing the properties of training an RBF network by injecting multiweight fault (each weight is of fault rate p), the following theorem can be proved.

Theorem 5: For injecting multiweight fault during training an RBF network, the weight vector $\mathbf{w}_t$ will converge with probability one to

$$\mathbf{w}^* = [(1+p)\mathbf{H})]^{-1} Y \tag{43}$$

where $\mathbf{H}$ and Y are given in (18). Besides, the corresponding objective function to be minimized is given by

$$\mathcal{L}(\mathbf{w}|\mathcal{D}) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)^2 + p\mathbf{w}^T\mathbf{H}\mathbf{w}. \tag{44}$$

Proof: Recall that $\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t(y_t - \Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}_t)\Phi(\mathbf{x}_t)$. Taking the expectation of the second term $(y_t - \Phi^T(\mathbf{x}_t)\tilde{\mathbf{w}}_t)\Phi(\mathbf{x}_t)$ with respect to $b_{i,t}, \mathbf{x}_t$ and y_t, we have

$$h(\mathbf{w}_t) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}_t)\Phi(\mathbf{x}_k) - p\mathbf{H}\mathbf{w}_t. \tag{45}$$

From (45), the solution $\mathbf{w}^*$ is given by $(1+p)\mathbf{H}\mathbf{w}^* = Y$. For nonsingular $\mathbf{H}$

$$\mathbf{w}^* = [(1+p)\mathbf{H}]^{-1}Y. \tag{46}$$

Hence, for all $\|\mathbf{w}-\mathbf{w}^*\| > 0$, we have

$$(\mathbf{w}-\mathbf{w}^*)^T h(\mathbf{w}) = (\mathbf{w}-\mathbf{w}^*)^T \left(Y - [(1+p)\mathbf{H}]\,\mathbf{w}\right)$$

which is less than zero. Therefore, condition C2) holds.

From (12) and the parallelogram raw, we have the following inequality:

$$\begin{aligned}\|M(\mathbf{w},\omega)\|^2 &\le 2\|y_t\Phi(\mathbf{x}_t)\|^2 + 2\tilde{\mathbf{w}}^T(\mathbf{H}_t)^2\tilde{\mathbf{w}}\\ &\le 2\|y_t\Phi(\mathbf{x}_t)\|^2 + 2\lambda_{\max}\{(\mathbf{H}_i)^2\}\|\mathbf{w}\|^2 \end{aligned} \tag{47}$$

where $\mathbf{H_t} = \Phi(\mathbf{x}_t)\Phi^T(\mathbf{x}_t)$, and $\lambda_{\max}\{(\mathbf{H_i})^2\}$ is the largest eigenvalue of the matrix $(\mathbf{H_i})^2$. The last inequality in (47) is due to the fact that $\|\tilde{\mathbf{w}}\|^2 \le \|\mathbf{w}\|^2$. Then, taking expectation of (47) with respect to $\mathbf{x}_t$ and y_t, we can easily observe that condition C3) holds.

Since $\mathbf{w}^*$ is the solution of

$$\frac{\partial}{\partial \mathbf{w}}\left\{\frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}))^2 + p\mathbf{w}^T\mathbf{H}\mathbf{w}\right\} = 0$$

and $(1/N)\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2 + p\mathbf{w}^T\mathbf{H}\mathbf{w}$ is in quadratic form, $\mathbf{w}^*$ is unique and the objective function is given by (44). The proof of Theorem 5 is completed. Q.E.D.

E. Multinode Fault Injection Training

Applying the Gladyshev theorem, the following theorem can be proved for injecting multinode fault training.

Theorem 6: For injecting multinode fault during training an RBF network, the weight vector $\mathbf{w}_t$ will converge with probability one to

$$\mathbf{w}^* = [\mathbf{H} + p(\mathbf{Q}-\mathbf{H})]^{-1}Y \tag{48}$$

where $\mathbf{H}, \mathbf{Q}$, and Y are given in (18). Besides, the corresponding objective function to be minimized is given by

$$\mathcal{L}(\mathbf{w}|\mathcal{D}) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)^2 + p\mathbf{w}^T(\mathbf{Q}-\mathbf{H})\mathbf{w}. \tag{49}$$

Proof: To prove condition C2), we need to consider the mean update equation $h(\mathbf{w}_t)$. Taking expectation of the second part of (14) with respect to $\beta_{i,t}, \mathbf{x}_t$ and y_t, we have

$$h(\mathbf{w}_t) = \frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}_t)\Phi(\mathbf{x}_k) - p(\mathbf{H}-\mathbf{Q})\mathbf{w}_t \tag{50}$$

where the solution $\mathbf{w}^*$ is given by

$$\mathbf{w}^* = [\mathbf{H} + p(\mathbf{Q}-\mathbf{H})]^{-1}Y. \tag{51}$$

Hence, for all $\|\mathbf{w}-\mathbf{w}^*\| > 0$, we have

$$(\mathbf{w}-\mathbf{w}^*)^T h(\mathbf{w}) = (\mathbf{w}-\mathbf{w}^*)^T (Y - [\mathbf{H} + p(\mathbf{Q}-\mathbf{H})]\mathbf{w})$$

which is less than zero. Therefore, condition C2) holds.

For condition C3), we have

$$\|M(\mathbf{w},\omega)\|^2 \le 2y_t^2\|\tilde{\Phi}(\mathbf{x}_t)\|^2 + 2\mathbf{w}^T(\tilde{\mathbf{H}}_\mathbf{t})^2\mathbf{w}. \tag{52}$$

where $\tilde{\mathbf{H}}_\mathbf{t} = \tilde{\Phi}(\mathbf{x}_t)\tilde{\Phi}^T(\mathbf{x}_t)$. Since $\tilde{\phi_i}(\mathbf{x}_t) = (1-\beta_{i,t})\phi_i(\mathbf{x}_t)$ where $\beta_{i,t}$ is equal to 1 or 0, we can easily have $\tilde{\Phi}^T(\mathbf{x}_t)\tilde{\Phi}(\mathbf{x}_t) \le \|\Phi(\mathbf{x}_t)\|^2$, and $\mathbf{w}^T(\tilde{\mathbf{H}}_\mathbf{t})^2\mathbf{w} \le \|\Phi(\mathbf{x}_t)\|^2\mathbf{w}^T\tilde{\mathbf{H}}_\mathbf{t}\mathbf{w}$. Besides

$$\int \mathbf{w}^T\tilde{\mathbf{H}}_\mathbf{t}\mathbf{w}P(\boldsymbol{\beta}_t)d\boldsymbol{\beta}_t = \mathbf{w}^T\mathbf{R_t}\mathbf{w}$$

where $\mathbf{R_t} = (1-p)((1-p)\mathbf{H_t} + p\mathbf{G_t}), \mathbf{H_t} = \Phi(\mathbf{x}_t)\Phi^T(\mathbf{x}_t)$, and $\mathbf{G_t} = \mathbf{diagH_t}$). Notice that $\mathbf{R_t}$ is symmetric and nonnegative. So, we have

$$\begin{aligned}&\int\left\{\|M(\mathbf{w},\omega)\|^2\right\}P(\boldsymbol{\beta}_t)d\boldsymbol{\beta}_t\\ &\qquad\le 2y_t^2\|\Phi(\mathbf{x}_t)\|^2 + 2\|\Phi(\mathbf{x}_t)\|^2\lambda_{\max}\{\mathbf{R_t}\}\|\mathbf{w}\|^2.\end{aligned} \tag{53}$$

Taking expectation of the above inequality with respect to $\mathbf{x}_t$ and y_t, one can readily show that condition C3) is satisfied.

Since $\mathbf{w}^*$ is the solution of

$$\frac{\partial}{\partial \mathbf{w}}\left\{\frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)^2 + p\mathbf{w}^T(\mathbf{Q}-\mathbf{H})\mathbf{w}\right\} = 0$$

and $(1/N)\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2 + p\mathbf{w}^T(\mathbf{Q}-\mathbf{H})\mathbf{w}$ is in quadratic form, $\mathbf{w}^*$ is unique and the objective function is given by (49). The proof of Theorem 6 is completed. Q.E.D.

It is worthwhile to note that (49) is also identical to the objective function derived in [25]. In other words, injecting random node fault during an RBF training is able to improve its ability to tolerate anticipated multinode random fault.

F. Weight Decay-Based Multinode Fault Injection Training

For weight decay-based multinode fault injection training, we will prove the following theorem.

Theorem 7: For injecting multinode fault during weight decay to train an RBF network, the weight vector $\mathbf{w}_t$ will converge with probability one to

$$\mathbf{w}^* = \left[\mathbf{H} + p(\mathbf{Q}-\mathbf{H}) + \frac{\lambda}{1-p}\mathbf{I}_{N_{\text{node}}\times N_{\text{node}}}\right]^{-1}Y \tag{54}$$

where $\mathbf{H}, \mathbf{Q}$, and Y are given in (18). Besides, the corresponding objective function to be minimized is given by

$$\begin{aligned}\mathcal{L}(\mathbf{w}|\mathcal{D}) = &\frac{1}{N}\sum_{k=1}^{N}(y_k - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)^2\\ &+\mathbf{w}^T\left\{p(\mathbf{Q}-\mathbf{H}) + \frac{\lambda}{1-p}\mathbf{I}_{N_{\text{node}}\times N_{\text{node}}}\right\}\mathbf{w}.\end{aligned} \tag{55}$$

TABLE I
OBJECTIVE FUNCTIONS OF THE FAULT/NOISE-INJECTION-BASED ONLINE LEARNING FOR RBF

Algo.	Fault/Noise injection	Objective Function
(A)	$\tilde{\mathbf{x}}_t = \mathbf{x}_t + \mathbf{b}_t,\, \mathbf{b}_t \sim \mathcal{N}(0, S_b)$	$\mathrm{MSE}(\mathbf{w}) + S_b \mathbf{w}^T \Gamma \mathbf{w}$
(B)	$\tilde{w_i} = (1 + b_i)w_i,\, b_i \sim \mathcal{N}(0, S_b)$	$\mathrm{MSE}(\mathbf{w})$
(C)	$\tilde{\phi_i} = (1 + b_i)\phi_i,\, b_i \sim \mathcal{N}(0, S_b)$	$\mathrm{MSE}(\mathbf{w}) + S_b \mathbf{w}^T \mathbf{Q} \mathbf{w}$
(D)	$\tilde{w_i} = (1 - \beta_i)w_i,\, P(\beta_i = 1) = p$	$\mathrm{MSE}(\mathbf{w}) + p\mathbf{w}^T \mathbf{H} \mathbf{w}$
(E)	$\tilde{\phi_i} = (1 - \beta_i)\phi_i,\, P(\beta_i = 1) = p$	$\mathrm{MSE}(\mathbf{w}) + p\mathbf{w}^T (\mathbf{Q} - \mathbf{H}) \mathbf{w}$
(F)	$\tilde{\phi_i} = (1 - \beta_i)\phi_i,\, P(\beta_i = 1) = p$	$\mathrm{MSE}(\mathbf{w}) + \mathbf{w}^T \left\{ p(\mathbf{Q} - \mathbf{H}) + \frac{\lambda}{1-p} I \right\} \mathbf{w}$

$MSE(\mathbf{w}) = (1/N) \sum_{k=1}^{N} (y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2$

Proof: By taking the expectation of the second part of (15) with respect to $\beta_{i,t}, \mathbf{x}_t$ and y_t, we have

$$h(\mathbf{w}_t) = (1-p) \left\{ \frac{1}{N} \sum_{k=1}^{N} (y_k - \Phi^T(\mathbf{x}_k)\mathbf{w}_t)\Phi(\mathbf{x}_k) - p(\mathbf{H} - \mathbf{Q})\mathbf{w}_t \right\} - \lambda \mathbf{w}_t. \quad (56)$$

With the about equation, the solution $\mathbf{w}^*$ is given by

$$\mathbf{w}^* = \left[\mathbf{H} + p(\mathbf{Q} - \mathbf{H}) + \frac{\lambda}{1-p} \mathbf{I}_{N_{\text{node}} \times N_{\text{node}}} \right]^{-1} Y. \quad (57)$$

Hence, for all $\|\mathbf{w} - \mathbf{w}^*\| > 0$, we have

$$(\mathbf{w} - \mathbf{w}^*)^T h(\mathbf{w}) = (\mathbf{w} - \mathbf{w}^*)^T \left\{ Y - [\mathbf{H} + p(\mathbf{Q} - \mathbf{H}) + \frac{\lambda}{1-p} \mathbf{I}_{N_{\text{node}} \times N_{\text{node}}}] \mathbf{w} \right\}$$

which is less than zero. Therefore, condition C2) holds. For condition C3), the proof is similar to that of the injecting multinode fault.

Then, taking expectation of the above inequality with respect to $\mathbf{x}_t$ and y_t, one can show that condition C3) is satisfied.

By the fact that $\mathbf{w}^*$ is the solution of

$$\frac{\partial}{\partial \mathbf{w}} \left\{ \frac{1}{N} \sum_{k=1}^{N} \left(y_k - \Phi^T(\mathbf{x}_t)\mathbf{w}_t \right)^2 + \mathbf{w}^T \left\{ p(\mathbf{Q} - \mathbf{H}) + \frac{\lambda}{1-p} \mathbf{I}_{N_{\text{node}} \times N_{\text{node}}} \right\} \mathbf{w} \right\} = 0$$

and $(1/N) \sum_{k=1}^{N} (y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2 + \mathbf{w}^T \{ p(\mathbf{Q} - \mathbf{H}) + \frac{\lambda}{1-p} \mathbf{I}_{N_{\text{node}} \times N_{\text{node}}} \} \mathbf{w}$ is in quadratic form, $\mathbf{w}^*$ is unique and the objective function is given by (55). The proof of Theorem 7 is completed. Q.E.D.

One should notice the weight decay effect is scaled up when random node fault is injected. The larger the p value is, the larger will be the weight decay effect. With proper control on the parameter λ, one is able to train an RBF to tolerate anticipated multinode random fault as well as improve generalization.

IV. NOISE-INJECTION-BASED LEARNING VERSUS EXPLICIT REGULARIZATION

Table I summarizes the results obtained in Section III. Due to the fact that some of these objective functions resemble the objective functions presented recently by other approaches [4], [45], it is necessary to explain the relations among these learning algorithms and those developed by other approaches.

As mentioned in Section I, the objective function derived in [1], [4], and [25] is basically the prediction error of a neural network that is corrupted by weight noise. Let us denote it by $\mathcal{L}_{\text{WN}}(\mathbf{w})$

$$\mathcal{L}_{\text{WN}}(\mathbf{w}) = \frac{1}{N} \sum_{k=1}^{N} \int (y_k - \Phi^T(\mathbf{x}_k)(\mathbf{w} + \Delta\mathbf{w}))^2 P(\Delta\mathbf{w}) d\Delta\mathbf{w} \quad (58)$$

where $P(\Delta\mathbf{w})$ is the probability density function of $\Delta\mathbf{w}$. If $\Delta\mathbf{w}$ is zero mean and its variance is equal to $S_b I_{N_{\text{node}} \times N_{\text{node}}}$, we have shown in [45] that

$$\mathcal{L}_{\text{WN}}(\mathbf{w}) = \frac{1}{N} \sum_{k=1}^{N} (y_k - \Phi^T(\mathbf{x}_k)\mathbf{w})^2 + S_b \mathbf{w}^T \mathbf{Q} \mathbf{w}. \quad (59)$$

By applying the idea of stochastic gradient descent, we can show that (59) is the objective function for the following online learning algorithm that is proposed by Bernier *et al.* [4][1]:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \mu_t \Big\{ (y_t - \Phi^T(\mathbf{x}_t)\mathbf{w}_t)\Phi(\mathbf{x}_t) - S_b \mathbf{diag} \left\{ \phi_1^2(\mathbf{x}_t), \ldots, \phi_{N_{\text{node}}}^2(\mathbf{x}_t) \right\} \mathbf{w}_t \Big\}. \quad (60)$$

Comparing the update equation given by (60) and the expectation of (9), we can see that they are corresponding to two different learning algorithms. On the contrary, it is worth to note that the update equation given by (60) and the expectation of (11) are actually the same equation. In other words, the algorithms as given by (60) and (11) are virtually the same. With the same initial conditions, both equations will converge to the same solution.

As a result, we can conclude the following. To obtain a single output RBF that is able to tolerate multiplicative weight noise, one could apply either injecting multiplicative node noise during training based on (11) or the regularization-based learning algorithm based on (60).

V. CONCLUSION

In this paper, the six fault-injection-based online training algorithms for RBF networks have been analyzed. Their corresponding objective functions have been deduced and their convergence proofs have been shown. Except for the case of injecting weight noise, all objective functions are of similar form

$$\text{mean square error} + \text{regularizer}.$$

[1] Refer to [4, eq. (6)].

For the case of injecting weight noise (either multiplicative or additive) during training, the objective function is the mean square error. Thus, we can conclude that online weight noise injection training is not able to improve an RBF network tolerance to weight noise effect. To obtain an RBF network that is able to tolerate multiplicative weight noise, one could inject multiplicative node noise during training [see (11)]. Owing to the similarities among the objective functions derived for fault/noise-injection-based algorithms and those based on regularization approach, a discussion on their relations is presented. While this paper focuses on RBF networks, it is worth to mention that convergence analysis on injecting weight noise during training an MLP is still an open problem. In a recent study [44], the divergence of weight vector during training has found. In such a case, theoretical analysis on the convergence and objective function of injecting multiplicative weight noise during training nonlinear neural networks will be one of our future work.

APPENDIX

Lemma 1: Let $\tilde{w} \in R^n$ be a random vector defined as follows:

$$\tilde{w} = w + Au$$

where i) $w \in R^n$ is a constant vector, ii) $A = [a_1, a_2, \cdots, a_m]$ (where $a_i \in R^n$) is a constant matrix, iii) $u = (u_1, u_2, \ldots, u_m)^T$ is a random Gaussian vector with mean zeros, and $u_i \sim \mathcal{N}(0, S)$. The expectation of the random matrix $\tilde{w}\tilde{w}^T\tilde{w}\tilde{w}^T = (\tilde{w}\tilde{w}^T)^2$ over the random vector u is given by

$$\begin{aligned}\int (\tilde{w}\tilde{w}^T)^2 P(u)du = (ww^T)^2 + 2Sww^TAA^T \\ + S\sum_{i=1}^{m} a_i^T a_i\, ww^T + Sw^TwAA^T \\ + 2SAA^Tww^T + 3S^2(AA^T)^2. \quad (61)\end{aligned}$$

Proof: By expanding $(w + Au)(w + Au)^T(w + Au)(w + Au)^T$, and taking expectation with respect to u, one can obtain that

$$\begin{aligned}\int (\tilde{w}\tilde{w}^T)^2 P(u)du = \Lambda_1 + \Lambda_2 + \Lambda_3 + \Lambda_4 \\ + \Lambda_5 + \Lambda_6 + \Lambda_7 + \Lambda_8 \quad (62)\end{aligned}$$

where

$$\begin{aligned}\Lambda_1 &= (ww^T)^2 \\ \Lambda_2 &= \int ww^T(Au)(Au)^T P(u)du \\ \Lambda_3 &= \int w(Au)^T w(Au)^T P(u)du \\ \Lambda_4 &= \int w(Au)^T (Au)w^{\mathrm{T}}\mathrm{P}(u)du \\ \Lambda_5 &= \int (Au)w^T w(Au)^T P(u)du \\ \Lambda_6 &= \int (Au)w^T (Au)w^T P(u)du \\ \Lambda_7 &= \int (Au)(Au)^T ww^{\mathrm{T}}\mathrm{P}(u)du\end{aligned}$$

and

$$\Lambda_8 = \int (Au)(Au)^T(Au)(Au)^{\mathrm{T}}\mathrm{P}(u)du.$$

Note that

$$\begin{aligned}Au = \sum_{i=1}^{m} u_i a_i \quad w(Au)^T = \sum_{i=1}^{m} u_i w a_i^T \\ (Au)w^T = \sum_{i=1}^{m} u_i a_i w^T \quad wa_i^T wa_i^T = ww^T a_i a_i^T \\ a_i w^T w a_i^T = w^T w a_i a_i^T \quad \sum_{i=1}^{m} a_i a_i^T = AA^T.\end{aligned}$$

Hence, the terms Λ_2 to Λ_7 in (62) can be obtained as follows:

$$\int ww^T(Au)(Au)^T P(u)du = Sww^TAA^T \quad (63)$$

$$\int w(Au)^T w(Au)^T P(u)du = Sww^TAA^T \quad (64)$$

$$\int w(Au)^T(Au)w^{\mathrm{T}}\mathrm{P}(u)du = S\sum_{i=1}^{m} a_i^T a_i\, ww^T \quad (65)$$

$$\int (Au)w^T w(Au)^T P(u)du = Sw^TwAA^T \quad (66)$$

$$\int (Au)w^T(Au)w^T P(u)du = SAA^Tww^T \quad (67)$$

$$\int (Au)(Au)^T ww^{\mathrm{T}}\mathrm{P}(u)du = SAA^Tww^T. \quad (68)$$

The eighth term will then be given by

$$\begin{aligned}&\int ((Au)(Au)^T)^2 P(u)du \quad (69) \\ &= \int \sum_{i,j,r,s} u_i u_j u_r u_s a_i a_j^T a_r a_s^T P(u) \\ &= 3S^2 \sum_{i=1}^{m} \left(a_i a_i^T\right)^2 + 3S^2 \sum_{i \neq j} \left(a_i a_i^T\right)\left(a_i a_j^T\right) \\ &= 3S^2 \sum_{i=1}^{m}\sum_{j=1}^{m} \left(a_i a_i^T\right)(a_j a_j)^T. \quad (70)\end{aligned}$$

Therefore

$$\int (Au)(Au)^T(Au)(Au)^{\mathrm{T}}\mathrm{P}(u)du = 3S^2AA^TAA^T. \quad (71)$$

Summing the results of the above seven terms [(63)–(68)] and ww^Tww^T is clearly obtained and the proof is completed.

Q.E.D.

ACKNOWLEDGMENT

The authors would like to thank the reviewers and associate editor for the valuable comments.

REFERENCES

[1] G. An, "The effects of adding noise during backpropagation training on a generalization performance," *Neural Comput.*, vol. 8, no. 3, pp. 643–674, Apr. 1996.

[2] J. L. Bernier, A. Diaz, F. Fernandez, A. Canas, J. Gonzalez, P. Martin-Smith, and J. Ortega, "Assessing the noise immunity and generalization of radial basis function networks," *Neural Process. Lett.*, vol. 18, no. 1, pp. 35–48, Aug. 2003.

[3] J. L. Bernier, J. Ortega, I. Rojas, and A. Prieto, "Improving the tolerance of multilayer perceptrons by minimizing the statistical sensitivity to weight deviations," *Neurocomputing*, vol. 31, pp. 87–103, Jan. 2000.

[4] J. L. Bernier, J. Ortega, I. Rojas, E. Ros, and A. Prieto, "Obtaining fault tolerant multilayer perceptrons using an explicit regularization," *Neural Process. Lett.*, vol. 12, no. 2, pp. 107–113, Oct. 2000.

[5] J. L. Bernier, J. Ortega, E. Ros, I. Rojas, and A. Prieto, "A quantitative study of fault tolerance, noise immunity, and generalization ability of MLPs," *Neural Comput.*, vol. 12, no. 12, pp. 2941–2964, Dec. 2000.

[6] C. M. Bishop, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, Jan. 1995.

[7] G. Bolt, "Fault tolerant multi-layer perceptron networks," Ph.D. dissertation, Dept. Comput. Sci., Univ. York, York, U.K., Jul. 1992.

[8] L. Bottou, "Stochastic gradient learning in neural networks," in *Proc. Neuro-Nîmes*, Nimes, France, Nov. 1991, pp. 687–706 [Online]. Available: http://leon.bottou.org/papers/bottou-91c

[9] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, D. Saad, Ed. Cambridge, U.K.: Cambridge Univ. Press, 1998, pp. 9–24 [Online]. Available: http://leon.bottou.org/papers/bottou-98x

[10] S. Cavalieri and O. Mirabella, "A novel learning algorithm which improves the partial fault tolerance of multilayer neural networks," *Neural Netw.*, vol. 12, no. 1, pp. 91–106, Jan. 1999.

[11] P. Chandra and Y. Singh, "Fault tolerance of feedforward artificial neural networks—A framework of study," in *Proc. Int. Joint Conf. Neural Netw.*, Portland, OR, Jun. 2003, vol. 1, pp. 489–494.

[12] C. T. Chiu, K. Mehrotra, C. K. Mohan, and S. Ranka, "Modifying training algorithms for improved fault tolerance," in *Proc. Int. Conf. Neural Netw.*, Orlando, FL, Jun. 1994, vol. 4, pp. 333–338.

[13] D. Deodhare, M. Vidyasagar, and S. Sathiya Keerthi, "Synthesis of fault-tolerant feedforward neural networks using minimax optimization," *IEEE Trans. Neural Netw.*, vol. 9, no. 5, pp. 891–900, Sep. 1998.

[14] P. Edwards and A. Murray, "Fault tolerant via weight noise in analog VLSI implementations of MLP's—A case study with EPSILON," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 45, no. 9, pp. 1255–1262, Sep. 1998.

[15] E. Gladyshev, "On stochastic approximation," *Theory Probab. Appl.*, vol. 10, no. 2, pp. 275–278, Jan. 1965.

[16] Y. Grandvalet, S. Canu, and S. Boucheron, "Noise injection: Theoretical prospects," *Neural Comput.*, vol. 9, no. 5, pp. 1093–1108, July 1997.

[17] N. Hammadi and H. Ito, "A learning algorithm for fault tolerant feedforward neural networks," *IEICE Trans. Inf. Syst.*, vol. 80, no. 1, pp. 21–27, Jan. 1996.

[18] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in Neural Information Processing Systems 5*. San Francisco, CA: Morgan Kaufmann, 1993, pp. 164–171.

[19] S. Himavathi, D. Anitha, and A. Muthuramalingam, "Feedforward neural network implementation in FPGA using layer multiplexing for effective resource utilization," *IEEE Trans. Neural Netw.*, vol. 18, no. 3, pp. 880–888, May 2007.

[20] Y. Jiang, R. Zur, L. Pesce, and K. Drukker, "A study of the effect of noise injection on the training of artificial neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 2009, pp. 2784–2788.

[21] K. Jim, C. Lee Giles, and B. Horne, "An analysis of noise in recurrent neural networks: Convergence and generalization," *IEEE Trans. Neural Netw.*, vol. 7, no. 6, pp. 1424–1439, Nov. 1996.

[22] N. Kamiura, T. Isokawa, Y. Hara, N. Matsui, and K. Yamato, "On a weight limit approach for enhancing fault tolerance of feedforward neural networks," *IEICE Trans. Inf. Syst.*, vol. 83, no. 11, pp. 1931–1939, Nov. 2000.

[23] T. Lai, "Stochastic approximation," *Ann. Stat.*, vol. 31, no. 2, pp. 391–406, Apr. 2003.

[24] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems 2*. San Mateo, CA: Morgan Kaufmann, 1990, pp. 598–605.

[25] C. S. Leung and J. Sum, "A fault-tolerant regularizer for RBF networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 3, pp. 493–507, May 2008.

[26] C. S. Leung, K. W. Wong, P. F. Sum, and L. W. Chan, "On-line training and pruning for RLS algorithms," *Electron. Lett.*, vol. 32, no. 23, pp. 2152–2153, Nov. 1996.

[27] C. S. Leung, K. W. Wong, P. F. Sum, and L. W. Chan, "A pruning method for the recursive least squared algorithm," *Neural Netw.*, vol. 14, no. 2, pp. 147–174, Mar. 2001.

[28] C. S. Leung, G. H. Young, J. Sum, and W. K. Kan, "On the regularization of forgetting recursive least square," *IEEE Trans. Neural Netw.*, vol. 10, no. 6, pp. 1482–1486, Nov. 1999.

[29] A. C. Mallofre and X. P. Llanas, "Fault tolerance parameter model of radial basis function networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jun. 1996, vol. 2, pp. 1384–1389.

[30] K. Matsuoka, "Noise injection into inputs in back-propagation learning," *IEEE Trans. Syst. Man Cybern.*, vol. 22, no. 3, pp. 436–440, May 1992.

[31] J. E. Moody, "Note on generalization, regularization, and architecture selection in nonlinear learning systems," in *Proc. 1st IEEE-SP Workshop Neural Netw. Signal Process.*, Sep. 1991, DOI: 10.1109/NNSP.1991.239541 .

[32] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion-determining the number of hidden units for an artificial neural network model," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 865–872, Nov. 1994.

[33] A. F. Murray and P. J. Edwards, "Synaptic weight noise during multilayer perceptron training: Fault tolerance and training improvements," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 722–725, Jul. 1993.

[34] A. F. Murray and P. J. Edwards, "Enhanced MLP performance and fault tolerance from synaptic weight noise during training," *IEEE Trans. Neural Netw.*, vol. 5, no. 5, pp. 792–802, Sep. 1994.

[35] C. Neti, M. H. Schneider, and E. D. Young, "Maximally fault tolerance neural networks," *IEEE Trans. Neural Netw.*, vol. 3, no. 1, pp. 14–23, Jan. 1992.

[36] M. W. Pedersen, L. Hansen, and J. Larsen, "Pruning with generalization based weight saliencies: γ-OBD, γ-OBS," in *Advances in Neural Information Processing Systems 8*. Cambridge, MA: MIT Press, 1995, pp. 521–527.

[37] D. S. Phatak, "Relationship between fault tolerance, generalization and the Vapnik-Cervonenkis (vc) dimension of feedforward ANNs," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 1999, vol. 1, pp. 705–709.

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

[39] R. Reed, "Pruning algorithms a survey," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, Sep. 1993.

[40] R. Reed, R. J. Marks II, and S. Oh, "Similarities of error regularization, sigmoid gain scaling, target smoothing, and training with jitter," *IEEE Trans. Neural Netw.*, vol. 10, no. 3, pp. 529–538, May 1995.

[41] A. Savich, M. Moussa, and S. Areibi, "The impact of arithmetic representation on implementing MLP-BP on FPGAs: A study," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 240–252, Jan. 2007.

[42] C. Sequin and R. Clay, "Fault tolerance in artificial neural networks," in *Proc. Int. Joint Conf. Neural Netw.*, Jan. 1990, pp. 703–708.

[43] M. Sugiyama and H. Ogawa, "Optimal design of regularization term and regularization parameter by subspace information criterion," *Neural Netw.*, vol. 15, no. 3, pp. 349–361, Apr. 2002.

[44] J. Sum and K. Ho, "SNIWD: Simultaneous weight noise injection with weight decay for MLP training," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, Dec. 2009, vol. 5863, pp. 494–501.

[45] J. Sum, C. S. Leung, and K. I.-J. Ho, "On objective function, regularizer, and prediction error of a learning algorithm for dealing with multiplicative weight noise," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 124–138, Jan. 2009.

[46] I. Takanami, "A fault-value injection approach for multiple-weight-fault tolerance of MNNs," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw.*, Washington, DC, Jul. 2000, vol. 3, p. 3515.

[47] E. Tchernev, R. Mulvaney, and D. Phatak, "Investigating the fault tolerance of neural networks," *Neural Comput.*, vol. 17, no. 7, pp. 1646–1664, Jul. 2005.

[48] R. Velazco, A. Assoum, N. Radi, R. Ecoffet, and X. Botey, "SEU fault tolerance in artificial neural networks," *IEEE Trans. Nuclear Sci.*, vol. 42, no. 6, pt. 1, pp. 1856–1862, Dec. 1995.

[49] J. Zhu and P. Sutton, "FPGA implementation of neural networks—A survey of a decade of progress," in *Proc. 13th Int. Conf. Field Programmable Logic Appl.*, Sep. 2003, pp. 1062–1066.

Kevin I.-J. Ho received the B.S. in computer engineering from the National Chiao Tung University, Taiwan, in 1983 and the M.S. and Ph.D. degrees in computer science from the University of Texas, Dallas, in 1990 and 1992, respectively.

From 1985 to 1987, he was an Assistant Engineer at the Institute of Information Industry, Taiwan. Currently, he is an Associate Professor of the Department of Computer Science and Communication Engineering, Providence University, Sha-Lu, Taiwan. His current research interests include image processing, algorithm design and analysis, neural computation, scheduling theory, and computer networks.

Chi-Sing Leung (M'04) received the B.Sci. degree in electronics, the M.Phil. degree in information engineering, and the Ph.D. degree in computer science from the Chinese University of Hong Kong, Hong Kong, in 1989, 1991, and 1995, respectively.

He is currently an Associate Professor at the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. His research interests include neural computing, data mining, and computer graphics.

Dr. Leung received the 2005 IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award for his paper titled, "The Plenoptic Illumination Function" published in 2002. He was a member of Organizing Committee of the 2006 International Conference on Neural Information Processing (ICONIP). He is the Guest Editor of *Neurocomputing* and *Neural Computing and Applications*. He is the Program Chair of the 2009 ICONIP. He is also a governing board member of the Asian Pacific Neural Network Assembly (APNNA).

John Sum (SM'05) received the B.Eng. degree in electronic engineering from the Hong Kong Polytechnic University, Hong Kong, in 1992 and the M.Phil. and Ph.D. degrees in computer science and engineering from the Chinese University of Hong Kong, Hong Kong, in 1995 and 1998, respectively.

He spent six years teaching in several universities in Hong Kong, including the Hong Kong Baptist University, the Open University of Hong Kong, and the Hong Kong Polytechnic University. In 2005, he moved to Taiwan and started to teach at the Chung Shan Medical University. Currently, he is an Assistant Professor at the Institute of E-Commerce, National Chung Hsing University, Taichung, Taiwan. His research interests include neural computation, mobile sensor networks and scale-free network.

Dr. Sum is an associate editor of the *International Journal of Computers and Applications*.