CHAPTER I

INTRODUCTION

A radial basis function (RBF) is typically realized by a Gaussian distribution function. A RBF neural network is widely used for data learning and problem solving, especially the classification problem, and also feasible to be implemented on a VLSI chip. Although, the function is popular and beneficial, it requires a costly learning time. Once the RBF is used to classify the data, it is ideal to minimize the network size, number of hidden unit, and computational time because these properties will create the highest efficiency when it is implemented on a chip. However, there is still a chance of faults to occur due to the damage of the input data or heat generated. Therefore, this thesis is aimed to create the higher fault tolerance of a RBF neural network. There are two major issues in this thesis to be focused on. The first issue is the problem of developing a new radial basis function with less learning time, and the second issue is the mathematical model of fault immunization of the proposed elliptic radial basis function.

1.1 Problems Identification

A highly efficient network refers to a network which not only works efficiently but also has high tolerance to any unexpected fault after implemented on a chip. The problem of designing such a network is categorized into three approaches based on the solution and the constraints of the problems which are

1. The design of self-detection of a faulty network by applying the techniques of digital

circuit design [17].

2. The design of a self-recovery network when there are faulty links and faulty neurons [13].

3. The design of a self-tolerant network by injecting noise into the input vectors, perturbing weights, and pruning some links during the retraining [11].

Many fault models [7, 9, 12, 10] are introduced and the solutions are proposed based on the assumption of the stuck-at-0 and stuck-at-1. Unlike those studies, we consider a general case of stuck-at-a, where a is a real value. There is another type of fault occurring at the weight vector. The fault is intermittent and prevented by a technique called fault immunization. Although these proposed techniques work well in most situations, there are still some possible improvements to enhance the robustness and reliability of the network.

Most of the previous studies [7, 9, 12, 10, 11, 14, 15, 18, 16] of fault tolerance were focused on the feedforward supervised neural networks for performing pattern classification problem. The study of enhancing the fault tolerance in a RBF network has not been studied before. The use of multivariate Gaussian function which is able to rotate, shrink or stretch accordingly to the density distribution and eigen-direction of the data vector is an expensive process due to the covariance matrix computation and inversion of the data vectors. This thesis focuses on two related problems. The first problem is how to speed up the learning process by using a new generic elliptic radial basis function (GERBF) with adjustable center and dimensional widths using the gradient descent learning rule. The second problem is to mathematically model the fault immunization of an GERBF neuron based on these adjustable parameters.

The rest of the thesis is organized as follows. Chapter I reviews the related literatures. Chapter II summarizes the theoretical background of the related neural network models. Chapter III discusses our proposed RBF model and the learning algorithm of the proposed network and the technique to enhance fault tolerance. Chapter IV explains and compares all the experimental results. Chapter V concludes the thesis.

1.2 Literature Review

1.2.1 Review of Literatures Related to RBF Neural Networks

John Platt [2] proposed a new learning algorithm called Resource Allocating Network (RAN) that allocated a new computational unit whenever an unusual pattern was presented to the network. This network is compact and easy as well as rapid to learn. The units in this network responded only to a local region of the space of input values. The network learned by allocating new units and adjusting the parameters of existing units. If the network performed poorly on a presented pattern, then a new unit would be allocated to correctly response to the presented pattern. If the network performed well on a presented pattern, then the network parameters would be updated using standard least mean square (LMS) gradient descent.

Lu Yingwei, Narashiman Sundararajan, and P. Saratchandran [4] extended John Platt's concept and presented a detailed performance analysis of the recently developed minimal resource allocation network (M-RAN) learning algorithm. M-RAN was a sequential learning radial basis function neural network which combined the growth criterion of the resource allocating network (RAN) of Platt with a pruning strategy based on the relative contribution of each hidden unit to the overall network output. The resulting network leads toward a minimal topology for the RAN. The performance of this algorithm was compared with the multi-layer feedforward networks (MFN) trained with (1) a variance of the standard backpropagation algorithm, known as resilient propagation (RPROP), and (2) the dependence identification (DI) algorithm of Moody and Antsaklis on several benchmark problems in the function approximation and pattern classification areas. For all these problems, the M-RAN algorithm was shown to realize the networks with significantly fewer hidden neurons and better or same approximation/classification accuracy. Furthermore, the time taken for learning (training) was also considerably shorter than the learning time of M-RAN because it did not require the repeated presentation of the training data.

Qiuming Zhu, Yao Cai, and Luzhen Liu [5] presented a new learning algorithm for the construction and training of a RBF neural network. The algorithm was based on a global mechanism of learning parameters using a maximum likelihood classification approach. The resulting neurons in the RBF network partitioned a multi-dimensional pattern space into a set of maximum sized hyper-ellipsoid subspaces in terms of the statistical distributions of the training samples. An important feature of the algorithm was the learning process including both the task of discovering a suitable network structure and of determining the connection weights. The entire network and its parameters were gradually evolved during the learning process.

Qiuming Zhu, Yao Cai, and Luzhen Liu [6] presented a pattern classification scheme in which the classifier was able to grow and evolve during the operation process. The evolutionary property of the classifier was made possible by modeling the pattern vectors in multiple hyper-ellipsoid subclass distributions. Learning of the classifier only took place at the subclass levels. This property allowed the classifier to retain its previously learned patterns while accepting and learning new pattern classes. The classifier was suitable to operate in the dynamical environments where continuous updating of the pattern class distributions was needed.

1.2.2 Review of Literature Related to Fault Tolerance and Immunization

R.K. Chun and L.P. McNamee [7] presented a methodology and a set of computer-aided design tools for measuring and improving the fault tolerance characteristics of neural networks. Two analysis programs developed using realistic fault networks appropriate for emulating potential hardware failures were discussed. It was demonstrated how functionally identical neural networks could have significantly different reliability characteristics. The criteria for selecting an optimal architecture and trained state were discussed. A modified training strategy which reduced the sensitivity of the network to faults was introduced. The process involved the deliberate injection of faults into the network during its training phase. The proposed scheme was analogous to viral immunization in the biological domain because it was the neural network's own adaptive capability which was utilized to improve its fault tolerance characteristics.

C. Neti, M.H. Schneider, and E.D. Young [9] described an application of a neural network modeling for generating the hypotheses about the relationships between response properties of neurons and information processing in the auditory system. The goal was to study the response properties that were useful for extracting sound localization information from directionally selective spectral filtering provided by the pinna. For studying sound localization based on spectral cues provided by the pinna, a feedforward neural network model with a guaranteed level of fault tolerance was introduced. Fault tolerance and uniform fault tolerance in a neural network were formally defined and a method was described to ensure that the estimated network exhibited fault tolerance. The problem of estimating weights for such a network was formulated as a large-scale nonlinear optimization problem. Numerical experiments indicated that solutions with uniform fault tolerance exist for the considered pattern recognition problem. The solutions derived by introducing fault tolerance constraints have better generalization properties than the other solutions obtained via the unconstrained backpropagation learning.

H. Elsimary, A. Darwish, S. Mashali, and S. Shaheen [10] presented a performance evaluation of a novel algorithm for fault tolerance training of artificial neural networks (ANNs). The proposed algorithm was based on a genetic algorithms technique. A realistic, and practical fault model was adopted. It reflected the failures arose during the hardware realization of ANNs, regardless of the hardware platform used in the implementation. Using this fault model, an algorithm was developed and the experimental results were performed to test the validity of the algorithm for different feedforward network sizes and types, and to check the ability of the algorithm to cover other fault models as a subset of the adopted one. A comparison with the conventional backpropagation learning algorithm was performed. The results show that the proposed algorithm was superior to the backpropagation from the fault tolerance point of view. The proposed algorithm has potential benefits in designing ANNs that could tolerate internal faults in the hardware realization of ANNs by incorporating fault tolerance in the training phase.

J.H. Kim, C. Lursinsap, and S. Park [8] investigated self-recovery methods in artificial neural networks (ANNs) implemented on a digital VLSI chip. Fault tolerance was the potential benefit of ANNs that extended beyond the high computation rates facilitated by the massive parallelism. If a faulty neuron or a faulty link occurred in ANNs implemented on a VLSI chip, typically, ANNs no longer classified all inputs correctly.

The ability of ANNs to achieve fault tolerance was not inherent, but must be built in. Also, the built-in fault-tolerant mechanism must be practical and efficient enough for a VLSI chip implementation. A partial relearning scheme was proposed to achieve fault tolerance. The scheme was applied to only a single neuron level, not entire networks. Therefore, the execution speed of the partial relearning would be much faster than that of the normal learning. Furthermore, the partial relearning could be executed in a parallel fashion.

Chun-Shin Lin and Ing-Chyuan Wu [11]presented a good fault tolerance which was desired in many control and automated systems. It had been long claimed that neural networks are fault tolerant, i.e., they could continue to operate after sustaining a partial damage. However, a very little evidence shows that neural networks evolved from any general learning algorithms really possessed such merit. They examined a learning method that intended to maximize the fault tolerance. The method was based on the well-known backpropagation learning algorithm. During the training, each neuron was given a small probability to have a simulated failure. This modification enforced that the computation be distributed to different computing elements in the network and thus maximizes the fault tolerance. Neurocontrollers developed using the new and conventional learning methods for pole balancing are compared.

C. Lursinsap and T. Tanprasert [15] proposed the injection of some chemical substances to a cell for enhancing the ability of the cell to fight against the intruder. This immunization concept in biological cells had been applied to enhance the fault tolerance capability in a perceptron-like neuron. They considered only the case where each neuron separated its input vectors into two classes. They mathematically modeled the cell immunization in terms of weight vector relocation and proposed a polynomial time weight relocating algorithm. This algorithm could be generalized to the case where each neuron separating the input vectors into more than two classes.

K. Sunat and C. Lursinsap [18] presented the fault immunization techniques to further enhance the fault tolerance of a neural network. The technique of Chun and McNamee [7] was based on the trial-and-error training which required a high computational-time. Lursinsap and Tanprasert [15] proposed a mathematical model to capture the characteristic of the fault immunization. However, this model was performed locally to each neuron after training which might deteriorate the target error and increased the computational time. They investigated the capability of two random optimization techniques for the the fault immunization improvement. A new cost function which combined the target error function and the immunization function was also proposed.

T. Tanprasert, C. Tanprasert, and C. Lursinsap [16] demonstrated that the robustness and weight fault tolerance of a neural network trained to learn a linearly separable problem could be enhanced if the network classified the problem nonlinearly. However, a multi-layer perceptron type network trained by an existing learning algorithm could normally promote the linear separability of the problem, resulting in non-optimal solution in terms of robustness and fault tolerance. They presented the technique for forcing the network to optimize its internal representation towards robustness and fault tolerance. The technique introduced a concept of "outpost" vectors for hiding the unwanted linearly separable characteristics of problem. Since such a task was rather specific, the "outpost" vectors were deterministically selected rather than randomly selected.