Chapter II

Fault Tolerance Immunization Concepts

2.1 General Concepts

Let $W_i = (w_{i,1}, w_{i,2}, \cdots, w_{i,n})$ be the weight vector of neuron i, $X_A^\alpha = (x_{A,1}^\alpha, x_{A,2}^\alpha, \cdots, x_{A,n}^\alpha)$ be the α^{th} vector in class A and $X_B^\beta = (x_{B,1}^\beta, x_{B,2}^\beta, \cdots, x_{B,n}^\beta)$ be the β^{th} vector in class B. Basically, a neuron i separating its input vectors into two classes, say A and B, acts as a hyperplane locating in between two input vector groups. The location of the hyperplane is captured by the weight vector W_i of the neuron and the classifying decision is based on the value of the dot product between W_i and X_μ, where X_μ possibly means X_A^α or X_B^β. If $W_i \cdot X_\mu$ is greater than a threshold value τ then X_μ is in class A otherwise is in class B. Without loss of generality, let τ equal zero. When the training is successful or converged the weight vector representing a hyperplane must be located in the empty space between input vectors in class A and class B. There are many possible locations to place the hyperplane. The interesting problem is how to find the best location that allows each $w_{i,j}$ to deviate as much as possible without violating the classification. Lursinsap and Tanprasert [11] mathematically modeled the fault immunization for vectors in binary space. We generalize their model for binary and real spaces by considering the binary space as a special case of the real space. The problem of fault immunization can be mathematically defined as follows and these following notations will be used throughout this thesis.

1. The trained weight vector of neuron i: $W_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,n})$.

2. The relocated weight vector of neuron i: $W_i' = (w_{i,1}', w_{i,2}', ..., w_{i,n}')$.

3. Input vectors in class A: $X_A^\alpha = (x_{A,1}^\alpha, x_{A,2}^\alpha, \ldots, x_{A,n}^\alpha)$, where $1 \leq \alpha \leq p$.

4. Input vectors in class B: $X_B^\beta = (x_{B,1}^\beta, x_{B,2}^\beta, \ldots, x_{B,n}^\beta)$, where $1 \leq \beta \leq q$.

5. Trained upper bound constant for $w_{i,j} : u_{i,j}$.

6. Trained lower bound constant for $w_{i,j} : l_{i,j}$.

7. Relocated upper bound constant for $w_{i,j}' : u_{i,j}'$.

8. Relocated lower bound constant for $w_{i,j}' : l_{i,j}'$.

The value of $u_{i,j}, l_{i,j}, u_{i,j}', l_{i,j}'$ are computed by gradually increasing or decreasing $w_{i,j}$ and $w_{i,j}'$, respectively, until a misclassification or output inversion occurs. The neuron is basically a binary classifier and its cost function is based on the firing characteristic of a neuron proposed by McCulloch and Pitts [12]. Since we focus our problem on a binary classification problem, the misclassification is defined as an event when the value of the output of a neuron is less than 0.5 for class B and greater or equal to 0.5 for class A. Therefore, the objective of fault immunization is to find W_i' to minimize

$$T_{i,j} = \left| \frac{x_{A,j}^{\alpha,}}{x_{B,j}^{\beta}} - \frac{u_{i,j}' - w_{i,j}'}{w_{i,j}' - l_{i,j}'} \right| ; 1 \leq j \leq n \text{ and } 1 \leq i \leq m$$

to satisfy these conditions:

1. $((w_{i,1}' + u_{i,1}'), \cdots, (w_{i,n}' + u_{i,n}')) \cdot (x_{A,1}^\alpha, \cdots x_{A,n}^\alpha) \geq 0$, for $1 \leq \alpha \leq p$ and

2. $((w_{i,1}' - l_{i,1}')), \cdots, (w_{i,n}' - l_{i,n}')) \cdot (x_{A,1}^\alpha, \cdots, x_{A,n}^\alpha) \geq 0$, for $1 \leq \alpha \leq p$ and

3. $((w_{i,1}' + u_{i,1}'), \cdots, (w_{i,n}' + u_{i,n}')) \cdot (x_{B,1}^\beta, \cdots, x_{B,n}^\beta) < 0$, for $1 \leq \beta \leq q$ and

4. $((w_{i,1}' - l_{i,1}'), \cdots, (w_{i,n}' - l_{i,n}')) \cdot (x_{B,1}^\beta, \cdots, x_{B,n}^\beta) < 0$, for $1 \leq \beta \leq q$.

Since the location of W' is defined by the shape of the empty space channel lying between vectors in class A and vectors in class B [11]. The channel is formed by all boundary vectors in both classes. The problem are how to find these boundary vectors and how to achieve the minimum of $T_{i,j}$.

2.2 Algorithms for Finding Boundary Vectors

The error function $E(W)$ depends on the vector pairs in classes A and B. In this section, we only focus our discussion on the algorithm of finding boundary vectors based on a single faulty link which may occurs to one of $w_{i,j}$'s. Later, we will discuss the reason that this algorithm is still correct when apply to multiple fault links. Since the size of W_i' is n therefore we need at least n boundary vectors from either class A or class B to compute the value of each $w'_{i,j}$. The simplest solution to finding these boundary vectors is by gradually increasing and decreasing each $w_{i,j}$ until a misclassification occurs. The misclassified vectors will be the boundary vectors. Notice that the process of increasing and decreasing the value of $w_{i,j}$ can generate at least one misclassified vector in class A and another misclassified vector in class B. For each neuron i, we find the boundary vectors with respect to each $w_{i,j}$ as follows. Let X_A^α be the vector α of class A, X_B^β the vector β of class B, and W_i the weight vector of neuron i.

Algorithm 1

1. Let $a_{i,j}^\alpha = \left| \frac{W_i \cdot X_A^\alpha}{x_{A,j}^\alpha} \right|$.
2. Let $b_{i,j}^\beta = \left| \frac{W_i \cdot X_B^\beta}{x_{B,j}^\beta} \right|$.
3. Find $\min_\alpha(a_{i,j}^\alpha)$ and $\min_\beta(b_{i,j}^\beta)$.

Theorem 1 *With respect to $w_{i,j}$, an input vector corresponding to $\min_\alpha(a^\alpha_{i,j})$ is the boundary vector in class A and an input vector corresponding to $\min_\beta(b^\beta_{i,j})$ is the boundary vector in class B.*

Proof Each $w_{i,j}$ can be rewritten in this form $(w_{i,j} + \delta_j)$, where δ_j is $a^\alpha_{i,j}$ or $b^\beta_{i,j}$ and equal to zero in normal situation. Without loss of generality, we only consider the boundary vectors in class A and prove only when $w_{i,j}$ is increased. The proof for the boundary vectors in class B and decreasing $w_{i,j}$ have the similar argument. Therefore the dot product $W_i \cdot X^\alpha_A$ becomes

$$\begin{aligned} W_i \cdot X^\alpha_A &= w_{i,1}x^\alpha_{A,1} + w_{i,2}x^\alpha_{A,2} + \cdots + (w_{i,j} + \delta_j)x^\alpha_{A,j} + \cdots + w_{i,n}x^\alpha_{A,n} \\ &= D + \delta_j x^\alpha_{A,j} \end{aligned}$$

Where $D = w_{i,1}x^\alpha_{A,1} + w_{i,2}x^\alpha_{A,2} + \cdots + w_{i,n}x^\alpha_{A,n}$. If we map the value of the dot product as a point to a real line we can see that the value of D represents the distance of this point from the threshold, which is equal to zero. From the above equation, it can be seen that increasing or decreasing δ_j is equivalent to constantly adding or subtracting $\delta_j x^\alpha_{A,j}$ from $w_{i,j}x^\alpha_{A,j}$. Therefore the number of times to increase or decrease δ_j until a misclassification occurs can be easily computed by dividing $W_i \cdot X^\alpha_A$ with $x^\alpha_{A,j}$. Hence, those vectors having $\min_\alpha(a^\alpha_{i,j})$ will be boundary vectors. Theorem tells us how to find the boundary vectors in class A and B. These boundary vectors may not be unique with respect to their $w_{i,j}$. This means that different $w_{i,j}$ and $w_{i,k}$ may generate the same set of boundary vectors. The following collorary states the condition of the uniqueness of the boundary vectors with respect to any $w_{i,j}$ and $w_{i,k}$. □

Collorary 1 *Vector X^α_A is boundary vector with respect to $w_{i,j}$ and $w_{i,k}$ if $x^\alpha_{A,j} = x^\alpha_{A,k}$.*

Proof If X^α_A is boundary vector with respect to $w_{i,j}$ and $w_{i,k}$ then we must have $\left|\frac{W_i \cdot X^\mu_A}{x^\mu_{A,j}}\right| = \left|\frac{W_i \cdot X^\alpha_A}{x^\alpha_{A,j}}\right|$. This implies that $x^\alpha_{A,j} = x^\alpha_{A,k}$. □

To find a boundary vector pair X_A^j and X_B^j, we use minimum Euclidean distance $d(X_A^j, X_B^j)$ to select each pair from classes A and B. The selecting algorithm is simply defined as follows.

Algorithm 2

1. For each vector X_A^j, find a vector X_B^k such that $d(X_A^j, X_B^k)$ is minimum. Vectors X_A^α and X_B^β are boundary vector pair.

2. For each vector X_B^l, find a vector X_A^m such that $d(X_B^l, X_A^m)$ is minimum. Vectors X_B^l and X_A^m are boundary vector pair.

These boundary vector pairs are used to relocate the weight vectors of the whole network to satisfy $E(W)$. The new position of each W_i must minimize $T_{i,j}$. The following theorems address the conditions when $T_{i,j}$ is minimum. We prove the theorems for the vectors in real space case first and then show that the condition for the binary space is a special case of the real space case.

Definition 1 *Let* $\delta_j^{(A)} = \frac{W_i \cdot X_A^\alpha}{x_{A,j}^\alpha}$ *and* $\delta_j^{(B)} = \frac{W_i \cdot X_B^\beta}{x_{B,j}^\beta}$.

For the simplicity of writing, we will drop the superscripts α and β from X_A^α, X_B^β, $x_{A,i}^\alpha$, and $x_{B,j}^\beta$. This will not change the meaning of each notation.

Theorem 2 *If* $W_i \cdot X_A \geq 0$ *and* $W_i \cdot X_B < 0$ *and* $\sum_{all\ boundary\ vector\ pairs}(W_i \cdot X_A + W_i \cdot X_B)^2 = 0$, *then the decision hyperplane is lying in the middle of space between the classes.*

Proof Let (X_A, X_B) be any boundary vector pairs with respect to the hyperplane that formed by W_i, which X_A is in class A and X_B is in class B. The minimum distance between X_A and the hyperplane is $\frac{|W_i \cdot X_A|}{||W_i||}$ and minimum distance between X_B and the hyperplane is $\frac{|W_i \cdot X_B|}{||W_i||}$, where $||W_i||$ is the Euclidean

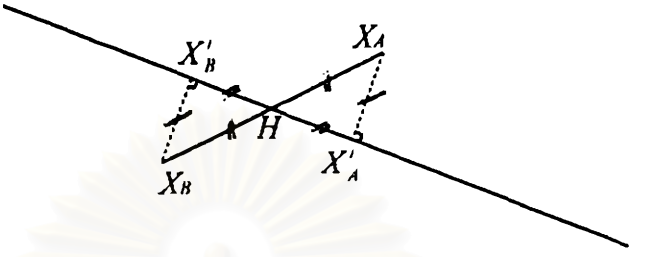

Figure 2.1: The companion figure of proof.

norm of vector. Since $\sum_{\text{all boundary vector pairs}} (W_i \cdot X_A + W_i \cdot X_B)^2 = 0$, it follows that $\frac{|W_i \cdot X_A|}{||W_i||} = \frac{|W_i \cdot X_B|}{||W_i||}$. Let H be the point on the hyperplane such that line $\overline{X_A X_B}$ intersects the hyperplane and X'_A and X'_B be two points on the hyperplane such that vectors $\overline{X_A X'_A}$ and $\overline{X_B X'_B}$ are perpendicular to the hyperplane. By the theorem of geometry we know that length of $\overline{X'_A H}$ = length of $\overline{X'_B H}$. Hence, there are two equivalence triangles and we can conclude that length of $\overline{X_A H}$ = length of $\overline{X_B H}$ (see Figure 2.1). □

Definition 2 *Let $f_{A,k}$ be the dot product of the new weight vector and the boundary vector pair k in class A, $f_{B,k}$ be the dot product of the new weight vector and the boundary vector k in class B.*

Lemma 1 $\frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} = -\frac{\delta_j^{(B)}}{\delta_j^{(A)}}$.

Proof From the algorithm 1, it follows that $u_{i,j} - w_{i,j} = \delta_j^{(A)}$ and $u_{i,j} - w_{i,j} = -\delta_j^{(A)}$. Hence, $\frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} = -\frac{\delta_j^{(B)}}{\delta_j^{(A)}}$. □

Lemma 2 *If $f_A \geq 0$ and $f_B < 0$ and $f_A + f_B = 0$, then $\delta_j^{(A)} x_{A,j} = -\delta_j^{(B)} x_{B,j}$.*

Proof Given $f_A \geq 0$ and $f_B < 0$ imply that $W_i \cdot X_A \geq 0$, $W_i \cdot X_B < 0$. If $f_A + f_B = 0$, then we have $W_i \cdot X_A + W_i \cdot X_B = 0$. Hence, there are $\delta_j^{(A)}$ and $\delta_j^{(B)}$ such that $W_i \cdot X_A + \delta_j^{(A)} x_{A,j} = 0$ and $W_i \cdot X_B + \delta_j^{(B)} x_{i,j} = 0$. It follows that

$$0 \;=\; W_i \cdot X_A + \delta_j^{(A)} x_{A,j} + W_i \cdot X_B + \delta_j^{(B)} x_{B,j} \,.$$

$$
\begin{aligned}
&= W_i \cdot X_A + W_i \cdot X_B + \delta_j^{(A)} x_{A,j} + \delta_j^{(B)} x_{B,j} \\
&= f_A + f_B + \delta_j^{(A)} x_{A,j} + \delta_j^{(B)} x_{B,j} \\
&= \delta_j^{(A)} x_{A,j} + \delta_j^{(B)} x_{B,j}
\end{aligned}
$$

Hence, $\delta_j^{(A)} x_{A,j} = -\delta_j^{(B)} x_{B,j}$. □

Theorem 3 *If* $f_A \geq 0$ *and* $f_B < 0$ *and* $f_A + f_B = 0$, *then* $T_{i,j} = \left| \frac{x_{A,j}}{x_{B,j}} - \frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} \right| = 0.$

Proof From Lemma 1, we have $\frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} = -\frac{\delta_j^{(B)}}{\delta_j^{(A)}}$, and from Lemma 2, we have $\frac{\delta_j^{(A)}}{\delta_j^{(B)}} = -\frac{x_{A,j}}{x_{B,j}}$. Hence, $T_{i,j} = \left| \frac{x_{A,j}}{x_{B,j}} - \frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} \right| = 0$. □

Collorary 2 *If input vector is in a binary space then if* $f_A + f_B = 0$ *implies* $T_{i,j} = \left| 1 - \frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} \right| = 0.$

Proof In binary space $x_{A,j} = x_{B,j} = 1$. Hence, $\delta_j^{(A)} = -\delta_j^{(B)}$. It follows that $\frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} = \frac{\delta_j^{(A)}}{\delta_j^{(B)}}$ and

$$
\begin{aligned}
T_{i,j} &= \left| \frac{x_{A,j}}{x_{B,j}} - \frac{u_{i,j} - w_{i,j}}{w_{i,j} - l_{i,j}} \right| \\
&= \left| 1 - \frac{\delta_j^{(A)}}{\delta_j^{(B)}} \right| \\
&= |1 - 1| = 0. \qquad \square
\end{aligned}
$$

Hence, the immunization problem can be transformed to be the problem of optimizing a cost function. The minimization of $T_{i,j}$ can be equivalently written in terms of the target error and the location error functions. This equivalence makes it is easy to find W_i'.

Let E_t be the target error of the network, E_l be the total sum of location error of hyperplanes in the network, W be the set of $w_{i,j}$, $\forall i, j$, after training,

and W' be the set of $w'_{i,j}$, $\forall i,j$, after relocation. We define an error function $E(W)$ by combining E_t and E_l as follows:

$$E(W) = E_t(W) + E_l(W), \tag{2.1}$$

$$E_t(W) = \frac{1}{2}\sum_{\mu}^{P}\sum_{j}^{N}(t_j^\mu - o_j^\mu)^2, \tag{2.2}$$

$$E_l(W) = \sum_{l=1}^{L}\sum_{k=1}^{K_l}(f_{A,k} + f_{B,k})^2, \tag{2.3}$$

where N is the number of output neurons, $f_{A,k}$ is the dot product of the new weight vector and the boundary vector pair k in class A, $f_{B,k}$ is the dot product of the new weight vector and the boundary vector k in class B, P is the number of patterns, K_l is the number of boundary vector pairs of neuron l, t_j^μ is the target of pattern μ of neuron j, o_j^μ is the output for pattern μ of neuron j, and L is the number of neurons in the network. We add the following two more conditions having these two error functions:

5. $E_t(W') \leq E_t(W)$ and

6. $E_l(W') = 0$.

2.3 Tolerance Measure

We define the following measure to evaluate the tolerance of $w_{i,j}$ and $w'_{i,j}$ computed from the algorithm. The measure defined below is similar to $T_{i,j}$ discussed in Section 2.1 except that we tighten the ratio of $\frac{x_{A,j}}{x_{B,j}}$ to

$$\left(\frac{x_{A,j}}{x_{B,j}}\right)_{\min_\rho d(X_A^\rho, X_B^\rho)}.$$

The reason that we must tighten this ratio is because the deviation of $w_{i,j}; \forall i,j$ depends upon empty space in between classes A and B whose width is measured