

A Thesis
entitled

Artificial Neural Network-Based Approaches for Modeling the Radiated Emissions from
Printed Circuit Board Structures and Shields

by

David T. Kvale

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the
Master of Science Degree in Electrical Engineering

Dr. Vijay Devabhaktuni, Committee Chair

Dr. Daniel Georgiev, Committee Member

Dr. Roger King, Committee Member

Dr. Patricia Komuniecki, Dean
College of Graduate Studies

The University of Toledo

December 2010

Copyright 2010, David Thomas Kvale

This document is copyrighted material. Under copyright law, no parts of this document may be reproduced without the expressed permission of the author.

An Abstract of
Artificial Neural Network-Based Approaches for Modeling the
Radiated Emissions from Printed Circuit Board Structures and Shields

by

David T. Kvale

As partial fulfillment of the requirements for the
Master of Science Degree in Electrical Engineering

The University of Toledo
December 2010

This thesis introduces a new modeling approach for efficient and accurate Electromagnetic Interference/Compatibility (EMI/EMC) analysis of electronic systems. Printed Circuit Boards' (PCB) radiated emissions were investigated by varying the number of apertures on a shield, changing the locations of partially shielded PCB traces, changing the locations of PCB interconnects, and moving EMI sources within a shielding enclosure. The issue with EMC modeling is that given the complexity of solving Maxwell's equations for a given PCB configuration, the best course for many engineers is to broadly follow design guidelines that are only true for a specific geometry for a specific solution frequency instead of solving Maxwell's equations for a given problem. There are cases where the complexity of the PCB design and integrated circuits (IC) is so extensive, that it is impractical to have an exact solution of Maxwell's equations (*i.e.*, modeling a functioning populated server motherboard within a cavity). Typically, EMC revisions are made to PCB designs if the Device Under Test (DUT) does not pass regulation certification, which can be very costly and time consuming. This is one of many reasons why PCB designs are infrequently changed, or if they are changed, only small variations are

made. In this thesis, it will be shown that Artificial Neural Networks (ANN) are capable of providing accurate, fast, and computationally light estimates for radiated emissions. One case study employs this computational tool to find an optimized location on a PCB for a trace interconnect. The significance of utilizing ANNs for optimization is that ANNs provide a fast and accurate tool for design as well as for estimating radiated emissions. However, given that ANNs are highly variable, many approaches to ANN creation are examined and evaluated for specific EMC examples. Since ANN models do not require detailed geometrical configurations of the PCB and cable structures under consideration, computational overhead requirements are significantly reduced as compared to electromagnetic and circuit tools. The robustness, efficiency, accuracy, and versatility of ANN models, as demonstrated in this thesis, are particularly useful in the electronics industry since most manufacturers prefer reusing circuits and PCB layouts in new products with minor modifications to the existing time-tested designs.

For my family, I know that without your support, none of this would be possible.

“Never tell people how to do things. Tell them what to do and they will surprise you with their ingenuity”

George S. Patton, Jr.

Acknowledgments

This thesis would not have been possible without the support given to me by my advisor, Dr. Devabhaktuni, and my mentor, Dr. Khan. I also acknowledge Dr. Russ Sprinkle for his editorial help. I would also like to acknowledge my peers in the laboratory for their assistance with different aspects of my research relating to but not limited to Artificial Neural Networks, creation of posters, differential equations, editing, EMC analysis, Matlab programming, object oriented programming, and many other topics. I would also like to acknowledge the support given to me from the EECS department at the University of Toledo as well as my friends, past and present, who have helped me. I am truly grateful for your contributions to my research and would like to sincerely thank you.

Contents

Abstract	iii
Acknowledgments	vi
Contents	vii
List of Tables	xi
List of Figures	xiii
List of Abbreviations	xvii
List of Symbols	xviii
1 Introduction to Electromagnetic Compatibility Modeling	1
1.1 Motivation for EMC Modeling	1
1.2 Problem Statement	2
1.3 State of the art in EMC Modeling and Limitations	3
1.3.1 Electromagnetic Finite Element Modeling (FEM)	4
1.3.2 Method of Moments Modeling (MoM)	5
1.3.3 Transmission Line Matrix Modeling (TLM)	5
1.3.4 EMC Design Guidelines	6
1.3.5 ANNs for RF Applications	6
1.4 Thesis Organization	7
2 Introduction to ANN Modeling	9
2.1 Overview of ANNs	9

2.2	Creation of Data Samples	13
2.3	Creation of ANN Structure	14
2.3.1	Type of Hidden Neuron Discussion	15
2.3.2	Number of Hidden Neurons	16
2.3.3	Generalizations about Structure on Training Times	17
2.3.4	Generalizations about Effective ANN Model Inputs	18
2.4	Training ANN Structure	18
2.4.1	Important Training Pairs to Include for Training	21
2.4.2	Generalizations on the Size of the Training and Validation Sets	21
2.4.3	Generalizations about Using Multiple Training Algorithms	22
2.5	Testing and Re-iteration	23
2.5.1	Error Equation Comparisons	23
2.5.2	Generalized Error Patterns While Testing	24
2.5.3	Practical Limitations to Exhaustive ANN Creation/Testing	25
2.6	Reverse Modeling Algorithm	28
3	EM-ANN Modeling of EMI Emissions from Perforated Cavities	33
3.1	Preliminary Investigations for Case Study One	34
3.1.1	Various Tests to ANN I/O Structure – Magnitude vs. dB	37
3.1.2	Various Tests to I/O - Near vs Far Field	39
3.1.3	Various Tests to I/O - Measurement Direction	42
3.1.4	Investigation on Network Type – Hidden Neuron Types	44
3.1.5	Investigation on Single Training Algorithms	46
3.1.6	Investigation on Multiple Training Algorithms	46

3.1.7 Size of Training Set on Error	49
3.1.8 Cropping Input Data Space on Error	51
3.1.9 Investigation on Changes to Inputs (specifically $ E_0 $)	52
3.2 Case Study One: Source Position Varied within a Shielding Enclosure	53
3.2.1 Procedure of Case Study One	54
3.2.2 Results of Case Study One	58
3.3 Preliminary Investigation for Case Study Two	61
3.3.1 Investigation on the Significance of an Outlier	63
3.3.2 Experimental Evidence on Extrapolation vs. Interpolation Training	65
3.4 Case Study Two: Large Variations to the Number of Apertures on a Shield	67
3.4.1 Procedure of Case Study Two	68
3.4.2 Results of Case Study Two	69
3.5 Preliminary Investigation for Reverse Modeling	72
3.6 Reverse Modeling of Case Study Two	73
3.7 Comparison of Full-Wave Solution to ANN	76
3.8 Summary	77
4 EM-ANN Modelling of PCB Cavity Structures	78
4.1 PCB Cavity Case Study: Variations of a Trace Position on a PCB	78
4.1.1 Procedure of PCB Cavity Case Study	78
4.1.2 Results of PCB Cavity Case Study	80
4.2 Comparison of Full-Wave Solution to ANN	83
4.3 Summary	83
5 ANN Modeling of PCB Interconnects	84

5.1	Preliminary Investigation for PCB Interconnect Case Study	84
5.2	Case Study on Variations of an Interconnect Termination Point	87
5.2.1	Procedure of Interconnect Case Study	87
5.2.2	Results of Interconnect Case Study	88
5.2.3	Investigations on Variations to the Number of N_{train}	89
5.2.4	Investigations on Repetitive f_{ann} Creation	91
5.3	Optimization of PCB Terminations Utilizing ANNs	92
5.4	Summary	98
6	Conclusions and Future Work	99
6.1	Contributions of this Thesis	99
6.2	Future Work	100
References		103
Appendix		
A	AutoIt! and NeuroModeler	110

List of Tables

2-1	Sample input data set for <i>NeuroModeler</i> , which could be used for training, validation, or testing.	13
3-1	Summary of d_p for $ E_y $ values for preliminary investigation on arbitrary aperture geometries with source location variations.	37
3-2	Preliminary investigation comparison of $ E_x $ and $ E_y $ for creation of sigmoid ANNs.	44
3-3	Preliminary investigation on different neuron types.	45
3-4	Preliminary investigation on training algorithms for a one aperture geometry.	46
3-5	Preliminary data results for different combinations of training algorithms from eight-aperture simulations.	47
3-6	Comparison of ANNs created by comparing cropped training data sets to non-cropped training data sets.	52
3-7	Comparison of E_{test} results of ANNs trained with and without $ E_0 $	52
3-8	Comparison between ANNs trained with differing numbers of inputs.	53
3-9	Comparison of a preliminary investigation on variations to the number of apertures as a function of N_{train}	67
3-10	Reverse modeling preliminary investigation.	72
3-11	Comparison of <i>NeuroModeler</i> and RMSE equations.	73
3-12	Comparison of reverse modeling techniques.	75

3-13 Comparison of d_p values to $f_{\text{ann}}(\mathbf{x}_p, \mathbf{w})$ and $F_{\text{ann},c}(\mathbf{x}_p, \mathbf{w})$ in the proposed model of Figure 3.23.	76
3-14 Comparison of computational time and memory requirements for ANNs and <i>HFSS</i>	76
4-1 PCB EMC modeling comparison for time and memory requirements.	83
5-1 Prelimination interconnect termination points.	85
5-2 f_{ann} creation results for PCB interconnects case study.	89
5-3 PCB optimization f_{ann} results.	94

List of Figures

2-1	Representation of a 3-layer input-output ANN model for EMI/EMC analysis..	10
2-2	Flowchart for ANN creation.	11
2-3	Graphical representation of the effect that the number of hidden neurons has for a 3-layer MLP ANN.	17
2-4	Training error for Chapter 3 first case study's f_{ann} , presented as an example of training error as a function of the number of training epochs.	19
2-5	Feedback loop for ANN creation.	20
2-6	Feedback loop for training set creation.	22
2-7	Figure of available combinations.	26
2-8	Input and output relationship as represented in a desired and in a correction ANN model.	29
2-9	Flow chart illustrating the proposed correction algorithm.	31
3-1	Overview of ANN creation from physics simulator to testing.	34
3-2	Preliminary one arbitrary aperture diagram with a cutaway view of the source oriented in the y-direction.	35
3-3	Far field measuring point for a one-aperture preliminary investigation on how a source's position affects radiated electric fields.	36
3-4	Near field, one-hole aperture for the preliminary d_p values experiment (a) V/m (b) same data but in dB V/m.	38

3-5 Percent error surface of $ E_x $ as calculated from a sigmoid neuron type ANN in the near field.	39
3-6 Far field d_p values, $ E_x $, for a one-aperture preliminary experiment.	40
3-7 Comparison of f_{ann} estimates of $ E_y $ where (a) is the far field and (b) is the near field.	41
3-8 Comparison of $ E_x $ and $ E_y $ from the physical solver, d_p , where (a) is $ E_x $ and (b) is $ E_y $	43
3-9 Average E_{test} results from an eight-aperture investigation, (a) arc tangent hidden neurons (b) sigmoid hidden neurons.	48
3-10 Preliminary investigation of the effect N_{train} has on the average E_{train} and E_{test} , determined that (a) N_{train} has seven hundred samples and (b) N_{train} has fourteen hundred samples.	50
3-11 Preliminary cropped data set simulation, where the f_{ann} estimates the six-aperture cropped data set results.	51
3-12 Overview of a PCB EMI source location varied within a shield; all dimensions are in centimeters.	55
3-13 ANN model estimation for $ E_y $ for the structure in Figure 3.12.	56
3-14 ANN creation model that shows the importance of parameter selection.	57
3-15 Radiated emissions for adding the fourth aperture on the PCB shield: (a) ANN estimation, (b) full-wave calculations.	58
3-16 Radiated emissions for a four aperture PCB shield: (a) Y_{offset} slice from Figure 3.15 the ANN $ E_y $ estimation, (b) X_{offset} slice from Figure 3.15, the full-wave calculations.	59

3-17 Comparison of the average E_{test} of f_{ann} as a function of perturbed weights/biases.	61
3-18 Preliminary simulation for Case Study Two.	62
3-19 Preliminary results on arbitrary hole addition.	63
3-20 Preliminary experiment that highlights the importance of training points in modeling error from the ANNs with the lowest average E_{test} in Figure 3-19. ...	64
3-21 Investigation results showcasing how moving training points into testing points affects average E_{train} and E_{test} : (a) an interpolating training data set, (b) an extrapolating training data set.	66
3-22 Overview of large variations to sequential variations to the number of aperture enclosing a dipole source at its center.	68
3-23 ANN model for estimating the radiated emissions for a given number of apertures on the shield.	69
3-24 Comparison between Case Study Two's (a) sequential aperture addition average E_{test} results and (b) arbitrary aperture addition average E_{test} results.	70
3-25 Estimated radiated emissions versus number of apertures.	71
3-26 Example of step five of the reverse modeling, $F_{\text{ann},c}$ modeling of the estimation for 168 apertures.	74
3-27 Comparison of different versions of $f_{\text{ann},jc}$ parameter values for $F_{\text{ann},c}$ modeling of the estimation for 168 apertures.	74
4-1 A voltage source driving a PCB trace terminated with a load and covered by a partial PCB shield. Dimensions are in centimeters.	79
4-2 ANN model for the structure in Figure 4.1.	80

4-3 Radiated emissions from the PCB shield structure (a) model estimation, (b) <i>HFSS</i> estimation.	81
4-4 Radiated emissions from the PCB shield structure of Figure 4.3. (a) radiated emissions vs. source location, (b) radiated emission vs. frequency.	82
5-1 Overview of PCB interconnects geometry diagram.	85
5-2 Preliminary investigations calculated d_p for the PCB interconnects case study.	86
5-3 Investigation on N_{train} vs. train time and average E_{test} for the f_{ann} s with each data set's lowest average E_{test}	89
5-4 Investigation on N_{train} vs. number of train epochs and train times.	90
5-5 f_{ann} iteration vs. train time/average train error for f_{ann} number seven found in Table 5.2.	91
5-6 f_{ann} iteration vs. time per f_{ann} calculation and testing's memory usage for f_{ann} number seven found in Table 5.2.	92
5-7 The f_{ann} estimate for $ E_y $ at 1.9 GHz.	95
5-8 Physical model estimates for 1.9 GHz for the PCB interconnect location.	96
5-9 Contour graph of the f_{ann} estimate for $ E_y $ as a function of position were the source frequency is 1.9 GHz.	97

List of Abbreviations

ANN	Artificial neural network
CQN	Conjugate gradient then quasi Newton algorithms
CQNM	Conjugate gradient then quasi Newton MLP algorithms
DUT	Device under test
EM	Electromagnetic
EMC	Electromagnetic compatibility
EMI	Electromagnetic interference
FDFD	Finite difference frequency method
FDTD	Finite difference time domain
FEM	Finite element method
<i>HFSS</i>	High frequency s-parameter simulator
ICs	Integrated circuits
I/O	Input/Output
KBNN	Knowledge based neural network
MLP	Multi layer perceptrons
MLP3	Multi layer perceptrons with 3 layers
MoM	Method of moments
PCB	Printed circuit board
PEC	Perfect electrical conductor
QN	Quasi Newton algorithm
QNM	Quasi Newton MLP algorithm
RF	Radio frequency
RMS	Root mean squared
RMSE	Root mean squared error
SE	Shielding effectiveness
s.e.	Standard error
TLM	Transmission line matrix

List of Symbols

$|E_0|$ Absolute value of the electric field with the top face removed
 $|E_x|$ Absolute value of the electric field with respect to the x -axis
 $|E_y|$ Absolute value of the electric field with respect to the y -axis

α Change factor
 γ Summation of bias end all precursor neuron outputs modified by weights
 Δy Change to an f_{ann} output
 δ Sign of $f_{\text{ann,jc}}$ with Δy changes
 $\lambda^{\text{solution}}$ Solution wave length
 σ Neuron output

A_{norm} Normalized area of apertures
 d_p Response of a system for a given input
 $E(\mathbf{w})$ Error function
 $E\%$ Percent error
 $E_{m,c}$ Corresponding error for correction algorithms m^{th} input
 E_{test} Testing error
 E_{train} Training error
 E_{user} User criteria for determining when ANN usage is fine
 E_{valid} Validation error
 E_{rms} Root mean square error
 e_p Error due to p^{th} sample
 $F_{\text{ann,c}}$ Value for y returned from the reverse model
 $f(\mathbf{x})$ Detailed physics based relationship
 f_{ann} ANN function
 $f_{\text{ann,c}}$ Correction algorithm ANN
 $f_{\text{ann,ic}}$ The i^{th} potential correction algorithm
 $f_{\text{ann,jc}}$ Correction ANN with least error
 $f_{\text{ann,n}}$ Set of potential correction algorithms
 h Prior sign of E_{obj}
 h^* Current sign of E_{obj}
 i Number of potential correction models
 j Correction algorithm with the least error
 k ANN or physical model output value
 l One synapse between neural layers

m	Number of parameters in \mathbf{x}
N	Number of sample input/output vectors
N_{test}	Number of testing samples
N_{train}	Number of training samples
N_{valid}	Number of validation samples
n	Number of parameters in \mathbf{y}
$nCr^{\text{Set Creation}}$	Number of sets (combinations)
nCr^{Test}	Number of sets (test combinations)
nCr^{Train}	Number of Sets (training combinations)
p	A given input sample
\dot{Q}^{air}	Ambient air heat transfer rate
\dot{Q}^{shield}	Heat transfer rate out of the shield
q	The q^{th} norm
T	Index of training data
X_{offset}	Position of the source along the x -axis to the origin
\mathbf{x}	Stimulus and geometry
$\mathbf{x}_{\text{m,c}}$	Input (m) correction input to f_{ann} calculated value
Y_{offset}	Position of the source along the y -axis to the origin
\mathbf{y}	Response of system to a stimulus and geometry
y_1	Smaller x input value
y_2	Larger y input value
y_{ann}	Result from ANN under test
\mathbf{w}	Vector containing weights associated with the various interconnects
\mathbf{w}^*	Weights that minimize $E(\mathbf{w})$

Chapter 1

Introduction to Electromagnetic Compatibility Modeling

Electromagnetic (EM) waves have been widely known to the public as early as the days of radio and telegraph pioneers (Maxwell, Hertz, Marconi, and others) [1]. Operators of early radios and telegraphs first knew of Electromagnetic Interference (EMI) as spurious and undesired effects in their devices, such as crackling/static sounds in radios. Operators knew of numerous noisy devices, such as DC electric motors, lighting, relays, and power lines, among other noise sources. The noise produced by these machines is referred to as *interference*. Much has been done since the early days of wireless communications in terms of policy and regulations in attempts to regulate emissions from many devices such that wireless devices have little to no effect on *unintentional* receivers of wireless communications. As stated by Paul [1] on page 2:

“A system is electromagnetically compatible with its environment if it satisfies three criteria:

1. It does not cause interference with other systems
2. It is not susceptible to emissions from other systems
3. It does not cause interference with itself”

1.1 Motivation for EMC Modeling

Typically, Electromagnetic Compatibility (EMC) is concerned with the generation, transmission, and reception of EM; therefore, in any EMC problem, there exists three

parts: 1) a source, 2) a coupling path, 3) and a receiver [1] [2] [3]. Sources could be one of the noisy devices mentioned above, lightning strikes, power line overloads, electrostatic discharges, electromagnetic pulses etc. [1]. In many regions of the world, the legal requirements [4] focus on radiated and conducted EM energy that can come from emissions, such as broadcast signals, or from noisy electronic devices, such as spark-gap generators. Legal requirements also focus on susceptibility from inputs and outputs or power planes, lightning strikes, transmission overloads, electrostatic discharge etc. The second part, a coupling pathway, might be obvious, such as long, attached cables connecting to a Printed Circuit Board (PCB), or it might not be an intuitive couple. The third part is the receiver, which might be a Radio Frequency (RF) receiver utilized for testing to ensure compliance, or it might be an operational amplifier that will magnify the noise and may disrupt or cripple system performance [1].

1.2 Problem Statement

Electromagnetic Compatibility design of electronic systems and integrated circuits (ICs) typically requires cautious selection of components/devices, careful PCB floor planning, and tenuous trace routing [5] [6] [7] [8]. The most accurate way to determine the impact of design parameters on the overall EMC performance of an electronic product is by constructing a PCB prototype, populating it with devices and circuit components, and testing the product in an EMC chamber. However, the prototyping process is time consuming and expensive since it requires multiple design iterations before an optimized design can be selected. Alternatively, computational tools can be employed to evaluate multiple design configurations and perform “what if” scenarios [9] [10] [11] [12]. The available tools, however, cannot always incorporate systems employing a variety of linear, nonli-

near, and digital devices—accurate models of which are usually not available. More importantly, prohibitively large computational overhead is required to account for the resonant structures, non-uniform cable bundles, and crowded PCB layouts in systems. This is further complicated by the fact that almost all electronic products designed today must satisfy EMC requirements when they are tested in compliance chambers. Incorporating the EMI/EMC measurement environment with realistic electronic systems is quite challenging for the existing tools, *e.g.*, since a typical EMC test setup involves placing a system in a CISPR-25 chamber where the chamber couples with the system-under-test [13] [14].

1.3 State of the art in EMC Modeling and Limitations

While designing PCBs, it can be difficult to estimate specific layout regions on the board that will cause emissions to exceed tolerances. EMC modeling often requires a full-wave solution to identify problematic areas on a PCB [15] [16]. For most cases, given the nature of EM solutions, analytic solutions are possible for specific geometries [17] [18] [19], but in most cases, numerical techniques must be applied. Full-wave solutions that solve Maxwell’s equations can be computationally expensive, even when using an optimized code running in a Fortran environment. Writing code that can be utilized to diagnose EMC design problem areas and solutions to the design problems can be arduous. Therefore, many EMC engineers choose to utilize a variety of less computationally optimized methods that are less arduous for the EMC engineer. These less laborious programs solve Maxwell’s equations in conjunction with a meshing program through techniques such as the following: Finite Element Method (FEM) analysis, Method of Moments (MoM), Finite Difference Time Domain (FDTD) method, Finite Difference Fre-

quency Domain (FDFD) method, Transmission Line Matrix (TLM) method, generalized multi-pole technique, conjugate gradient method, boundary element method, uniform theory of diffraction, and hybrid techniques [2]. In this thesis, physical modelers based on the techniques outlined in this section will be utilized to train ANNs.

1.3.1 Electromagnetic Finite Element Modeling (FEM)

Commercial packages such as *ANSYS*'s *HFSS* use hybrid FEM techniques to obtain full-wave solutions for a bound region [2] [20] [21]. This is done by dividing the simulation geometry by meshing it into homogenous tetrahedrals or voxels, and the computational goal is to solve the field quantities at the nodes [22]. FEM are *variational* techniques that work by minimizing or maximizing an expression that is known to be stationary about the true solution [2]. Generally, this is accomplished by minimizing a functional energy. This process is both time consuming and memory intensive. It is also possible to have convergence issues. Expert experience in EMC problems can give an insightful design review that can give additional focus on probabilistic regions of interest, simulation simplification (*i.e.*, symmetry, mesh size/convergence, idealization), and past experience. Even with this expertise, many EMC problems will still be computationally expensive if not impractical to simulate.

For FEM codes, the error of the calculated values can be generalized as approaching zero as the number of tetrahedral and series sums approaches infinity [23]. However, given that the problems must be solved numerically, the sums must be truncated, typically to user-specified tolerances. This tolerance is significant because the time required is non linear, and for a simulation to converge within a few percent error of a measured value might take several hours to several days to complete, if at all, with given computational

tools. However, FEM simulations are well suited for the research presented in this thesis, and the physics-based values are derived from this method.

1.3.2 Method of Moments Modeling (MoM)

Method of Moments Modeling (MoM) is a method for solving complex integral equations by reducing them into a system of simpler linear equations [2] [24]. A popular commercial solver utilizing a hybrid of this method is *FEKO*. Paraphrasing Hubing on page five [2], the MoM technique differs from FEM by utilizing a technique known as the *method of weighted residuals*. The residuals are a measure of the difference between the trial solution and the true solution, where variable parameters are determined such that a *best fit* of the trial function is based on a minimization of the residuals. “Moment method techniques applied to integral equations are not very effective when applied to arbitrary configurations with complex geometries or inhomogeneous dielectrics,” according to Hubing [2] on page seven; therefore, they are not well suited for the analysis of the interior of conductive enclosures and therefore are not suited for many of the problems presented in this thesis. Therefore, this method was not further examined.

1.3.3 Transmission Line Matrix Modeling (TLM)

According to Hubing [2], a TLM solution is unique; however, it is similar to an FDTD method in that the analysis is conducted in a grid of nodes with each node interconnected by transmission lines. Lossy media can be modeled by introducing loss into the TLM equations or by loading the nodes with lossy stubs. Typically, the meshed grid is done in \mathbb{R}^3 , and each node is orthogonal to other nodes. The nature of this technique is that it is well suited for parallelization. Given the nature of the grid, a voluminous problem requires a fine grid which creates excessive amounts of computation. However, these tech-

niques' advantages are that complex nonlinear materials are readily modeled [25] [26] [27]; however, given time constraints, TLM modeling was not utilized extensively in the research presented in this thesis.

1.3.4 EMC Design Guidelines

Given that numerical solutions of full-wave simulations for EMC simulation can be computationally expensive or impractical, a variety of other techniques have been developed to give EMC guidelines to PCB engineers [28] [29]. Practically, these guidelines are relevant to case-specific applications and therefore are situationally useful. These guidelines can be applied to cases outside of the specific application and are therefore inaccurate to the point that the guidelines could create more EMC issues than choosing not to apply them at all. One reason that these guidelines are still used today is that the guidelines require several orders of magnitude smaller computational overhead as compared to full-wave simulations. Another constraint with full-wave simulations is that several EM commercial packages are cost prohibitive, and although packages are available at a substantially reduced cost, there are other tradeoffs to be considered. EMC problems that utilize full-wave solvers and design guidelines could benefit greatly from an alternative technique that is fast, reliable, and computationally efficient. One such alternative is ANNs.

1.3.5 ANNs for RF Applications

ANNs learn the behavior of a given system through an optimization process and provide a “system level transform” for a given set of inputs and outputs. For this purpose, ANNs require a relatively limited amount of representative data (measured or simulated). In this context, ANNs are more versatile than analytical models since they are not limited to cer-

tain structures and are more efficient than the full-wave simulation tools since they do not need to re-model a system for any change in the input parameters.

An ANN is a series of mathematical functions that can be trained utilizing a multitude of algorithms to approximate a variety of phenomena given the universal approximation theorem [30]. ANNs have been shown to be successful with a variety of tasks for Radio Frequency (RF) modeling [31] [32] [30] [33] and design for PCB component modeling, such as bends [34], embedded passive components [35], transmission-line components [36] [37] [38], transistors [39], vias [40], coplanar waveguide components [41], spiral inductors [42], and amplifiers [43] [44] [45]. Utilizing current research tools, ANNs have been applied in a limited capacity for EMC problems such as detection and identification of vehicles based on their unintended electromagnetic emissions [46], calculation of multilayer magnetic shielding [47], estimating PCB configuration from EMI measurements [48], characterization and modeling of the susceptibility of ICs to conducted EM disturbances [49], recognition and identification of radiated EMI for shielding apertures [50]. This thesis presents an ANN-based alternative to EMC modeling and radiated emission estimation that has not been utilized before, specifically as a tool for estimating RF emissions from a PCB source, within a shielded enclosure, or as a PCB interconnect.

1.4 Thesis Organization

Chapter 2 is an introduction and in-depth discussion of ANN techniques and recommended ANN usage for EMC problems, as well as an introduction to an ANN reverse modeling technique.

Chapter 3 is an ANN-based modeling approach to estimate the radiations from two distinct perforated enclosures. This chapter presents preliminary investigations and pri-

mary case studies. The focus of the preliminary investigations is to verify the interest in the problem and the ANN recommendations as outlined in Chapter 2 in an experimentally relevant context. In addition, these preliminary investigations also verified that the case studies could be analyzed with an FEM solver. Then cases studies are presented and explained in greater depth than they are in the preliminary investigations. The two case studies are as follows: (1) a design project where an EMC source location is varied within a shielding enclosure and the project has the source location and number of apertures varied to examine the effects on emissions, and (2) a design where emissions from an EMC source located within a shielding enclosure is measured; however, this second case study includes large changes to the number of apertures. The solution frequency is fixed for these studies; also, the reverse modeling approach is applied to the second case study.

Chapter 4 provides an ANN approach for estimating the EM radiation from a PCB trace. This case study focuses on a PCB design problem where a source emanating inside of a cavity connects to a trace that exits the shield and connects to a load on a PCB plane. This experiment examines how accurately ANNs can estimate the emissions for a given trace location within the cavity as well as variations to the solution frequency.

Chapter 5 provides an ANN approach to estimate the emissions from a PCB interconnect. This focus is on how interconnects' termination points for two PCBs affect radiation emissions as a function of space and frequency. These ANNs are utilized for optimizing the location of the interconnect for a particular frequency that minimizes emissions.

Finally, Chapter 6 provides conclusions and recommendations for future work. An appendix is included to outline a program created to expedite ANN modeling.

Chapter 2

Introduction to ANN Modeling

As mentioned in Chapter 1 (Section 1.3), ANNs have been successfully used for RF and EMC modeling. This chapter introduces general ANN modeling, specifically utilizing *NeuroModeler* version 1.5 [51]. *NeuroModeler* is a research-based application whose purpose is to create ANNs for microwave engineering and design. After an introduction to ANNs, ANN recommendations are presented. ANN recommendations presented in this chapter are discussed in the subsequent chapters in relevant preliminary investigations, giving insight on how ANN guidelines perform for EMC modeling. The significance of the recommendations is to simplify ANN creation which is based on collected data. These recommendations generally created more successful ANNs for EMC. The chapter concludes with a reverse modeling technique that can reduce the ANN's estimation average and/or max error.

2.1 Overview of ANNs

ANNs can generally be thought of as an approximation or fitting function. The robustness and generalization capabilities of ANN models make them good candidates for EMI/EMC simulations. Depending on the user's requirements, the training/validation sets can be tailored for the user's specific application. This tailoring process involves adjust-

ing the size of the sets utilized in training, testing, and validation; the order of samples within the sets; and the samples to be included in each set. Once a preliminary training set is used, intelligent decisions can be made about changing the network structures if the error of the network is greater than user tolerances. The general structure of ANNs that have been utilized extensively in this thesis is abstractly shown in Figure 2.1.

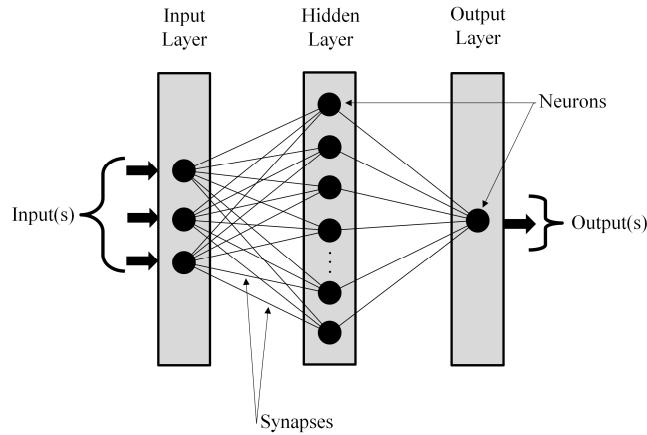


Figure 2.1: Representation of a 3-layer input-output ANN model for EMI/EMC analysis.

For the EMC ANNs presented, generally a multi-layer perceptron (MLP) network with three layers was used, where the input layer contained relay neurons; the middle layer, referred to as the *hidden layer*, was sigmoid-type neurons, and the output layer consisted of linear-type neurons. ANNs need to be trained and should be validated before general usage. After training, the ANNs are validated, and if the error is not within tolerances, changes to the ANN creation can be made, such as changing the structure. Changes to the structure could include changes to the inputs and outputs of the ANN, the type of neurons, the number of neurons in the network, the training algorithms, the number of epochs while training, the training error threshold dictated by the user, and swapping training and validation samples to the training sets. This is shown in Figure 2.2.

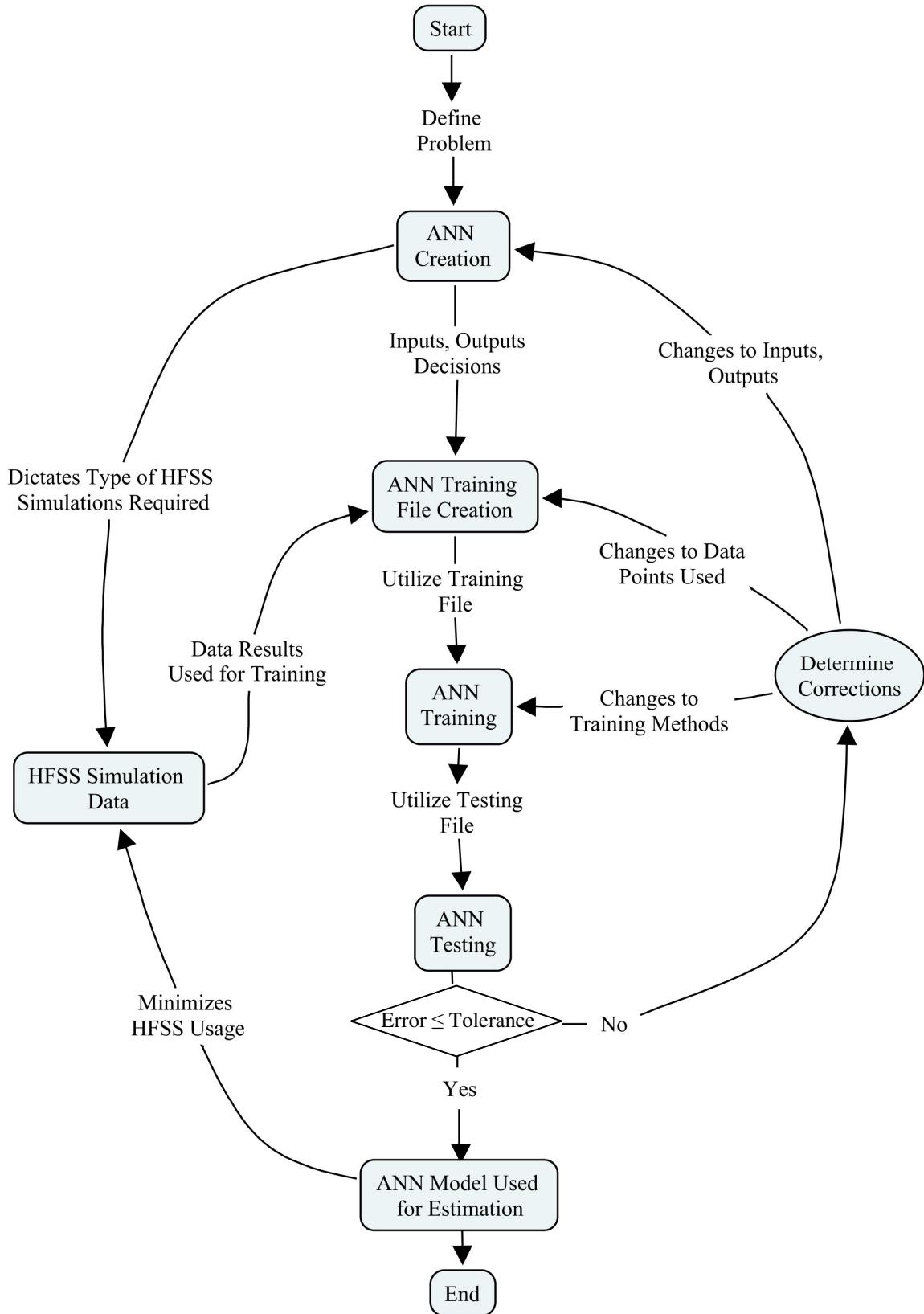


Figure 2.2: Flowchart for ANN creation.

Training parameters that can be user specified are the maximum number of epochs, error tolerance, Hessian refresh interval, maximum step size, and model parameter tolerance.

After the user has validated the networks, the ANNs can then be used for EMC modeling.

Figure 2.2 can be simplified into three steps:

Step 1: *Input-Output Definition*: Define a set of inputs, \mathbf{x} , and a set of outputs, \mathbf{y} . This definition may be critical in the overall model accuracy as the output, \mathbf{y} , may not be sensitive to some input parameters defined in the vector, \mathbf{x} .

Step 2: *Training ANN models*: Accurate data samples ($\mathbf{x}_p, \mathbf{d}_p$), where p is the p^{th} sample number in a data set, are employed for training the ANN. To this end, several training algorithms are available, including the following: quasi-Newton (QN); quasi-Newton MLP (QNM); conjugate gradient algorithm, conjugate gradient quasi-Newton algorithm (CQN); the conjugate gradient quasi-Newton MLP algorithm (CQNM); where CQN and CQNM are algorithms that run sequentially.

Step 3: *Validation and Testing ANN models*: To verify the accuracy of the trained ANN, the model is tested against a set of new data samples.

Given the constraints of *NeuroModeler*, i.e., being arduous for the creation of a large number of ANNs, a program was developed to automate ANN creation. This program was created utilizing *AutoIt!* and is reviewed briefly in the appendix. Typically, the creation of ANNs was accomplished by specifying the ANN structures, and then the automation program produced the networks. With this overview to ANN structure, a broad introduction to modeling is presented. In the following sections, the ANN modeling approach's specific parts and the associated training process are discussed in greater depth.

2.2 Creation of Data Samples

The training and testing sets for the EMC ANNs can be created by a variety of sources, such as measured values, analytic solutions, FEM solvers, MoM solvers, TLM solvers, and other methods as mentioned in Section 1.3. The training data can also be obtained through measurements. The data is then formatted into sample (vector) pairs which include the inputs and the outputs. A sample data set is demonstrated in Table 2.1.

Table 2.1: Sample input data set for *NeuroModeler*, which could be used for training, validation, or testing.

Sample Number	\mathbf{x}			\mathbf{y}
	Frequency (GHz)	X_{offset} (cm)	Y_{offset} (cm)	$ E_y $ (dB V/m)
1	1.5	6	7	-120
:		:		:
$p-1$		\mathbf{x}_{p-1}		\mathbf{d}_{p-1}
p		\mathbf{x}_p		\mathbf{d}_p
$p+1$		\mathbf{x}_{p+1}		\mathbf{d}_{p+1}
:		:		:
N		\mathbf{x}_N		\mathbf{y}_N

The symbols in Table 2.1 indicate the following: \mathbf{d}_p is the physical model value, N is the number of samples in a data set, and $|E_y|$ is the measured radiated electric field from a source with respect to the y -axis. The ANN under creation and test, referred to as f_{ann} , can also be trained to predict other quantities, such as magnetic field magnitudes and shielding effectiveness. The regions of the table highlighted with the dark borders would be the required data that is needed by *NeuroModeler* to create ANNs. If the user wanted trained f_{ann} to calculate output values, then only \mathbf{x} is required. The relationship between \mathbf{y} and \mathbf{x} can be represented as $\mathbf{y} = f(\mathbf{x})$, where f represents a detailed physics-based relationship (*e.g.*, Maxwell's equations connecting charges/currents to the resulting electromagnetic fields). The ANN model for such a relationship can be represented as $\mathbf{y} = f_{\text{ann}}(\mathbf{x}, \mathbf{w})$,

where w is a vector containing weights associated with the various interconnections/links among the neurons.

The inputs are from the model geometry (variable parameter, array changes etc.) and the excitation (frequency change, polarization changes, phase etc.). The outputs are user specified; for this thesis, the assumed output is the absolute value of the electric field with respect to a direction typically in units of dB V/m several wavelengths from a source. Other outputs that might be desirable for a user would be voltage (noise), magnetic field values, Poynting vector etc. Given that ANNs are an approximation function, it is highly recommended that the extremes in the input data space be included in the training set to ensure the lowest error in the f_{ann} s.

2.3 Creation of ANN Structure

After data collection, training, validation, and testing sets are formed, and the ANN creation can begin. An ANN typically contains an input layer, an output layer, and one or more hidden layers [51]. This can be seen in Figure 2.1. Moody and his associates [52] have completed a detailed analysis on the effects of parameters for ANNs; however, many of the theories are out of scope for this thesis. These layers are composed of several neurons, which are connected through links or synapses. Hidden-layer neurons are essentially mathematical transfer functions. Synapses are weighing coefficients (w) that modify the neuron output values by a fixed amount. This parameter is randomized between zero and one when ANNs are initialized in *NeuroModeler*. Techniques exist that optimize initialization or ANN structures, such as knowledge-based neural networks (KBNN) [37]; however, this is out of scope for this thesis given that this is an advanced ANN technique. Types of neurons available in *NeuroModeler* that could be used for ANN structures in-

clude the following: Sigmoid, User Linear, Resistor Model, Relay, Quadratic, Positive Linear, Positive Constant, Normalize, Neural Model, Multiply2, MultiSymbolic Fixed 1-3, MultiSymbolic 1-6, MultiSigmoid, MultiLinear2-6, MexicanHat, Linear, Hyperbolic Tangent, Gaussian, ExtSimulator, Dip, DecaySinusoidal, Constant, CktSimulatorPS, CktSimulator, and Arctangent. All these functions are bounded, continuous, monotonic, and continuously differentiable. Neurons also have a bias parameter. Biases are randomized between zero and one for ANN initialization in *NeuroModeler*. In this thesis, perturbing the weights and biases has the same effect as though a new ANN structure were created in *NeuroModeler*. Also, given these random initialization values, ANNs are a stochastic process. After the ANN is initialized, the ANN has to be given initial weights before training. The structure of an ANN is defined by the number of layers, number of neurons in a given layer, type of transfer function selected, number of inputs, and the desired outputs.

2.3.1 Type of Hidden Neuron Discussion

In general, sigmoid hidden neurons are one of the best choices for the hidden neuron types utilized for EMC. This is because sigmoid neurons are one of the easiest neurons to train, give one of the lowest systematic network errors, and require no additional advanced training techniques [51]. The sigmoid function is given by

$$\sigma(\gamma) = \frac{1}{1 + e^{-\gamma}}, \quad (2.1)$$

where σ is the neuron's output and γ is the summation of all precursor neurons' outputs multiplied by their weights added to a bias.

γ is defined by:

$$\gamma = \sum_{l=1}^m w_l * x_l + w_0, \quad (2.2)$$

where m is the m^{th} input of x , l is one of the synapse linkages between the neuron and the preceding layer, w_l is the weight given to the l^{th} input, x_l is the l^{th} input, and w_0 is the initial bias. This thesis primarily utilizes sigmoid hidden neurons, and this recommendation is experimentally examined further in Chapter 3. To summarize the equation 2.1, σ goes to one as γ goes to infinity, and σ goes to zero as γ goes to negative infinity. This guideline is examined experimentally further in a preliminary experiment-relevant context in Section 3.1.4.

2.3.2 Number of Hidden Neurons

The number of hidden neurons recommended in this thesis follows the assumption that as the number of hidden neurons approaches zero or infinity, the training/testing error becomes sporadic. This is summarized in Figure 2.3. From Koroglu [47]; their experimental evidence utilizing ANNs found that the number of hidden neurons to an MLP affects the error associated with the f_{ann} s. ANNs which have small and large structures can become either under or over trained due to too few or too many neurons.

Examining region A in Figure 2.3, there appears to be a number of hidden neurons in that region that minimizes error. The estimated minimum average testing error (E_{test}) is data-set specific as well as dependent on neuron type; it is also network-type specific. For the purposes of this thesis, case studies extensively utilize sigmoid hidden neurons 3-layer MLPs.

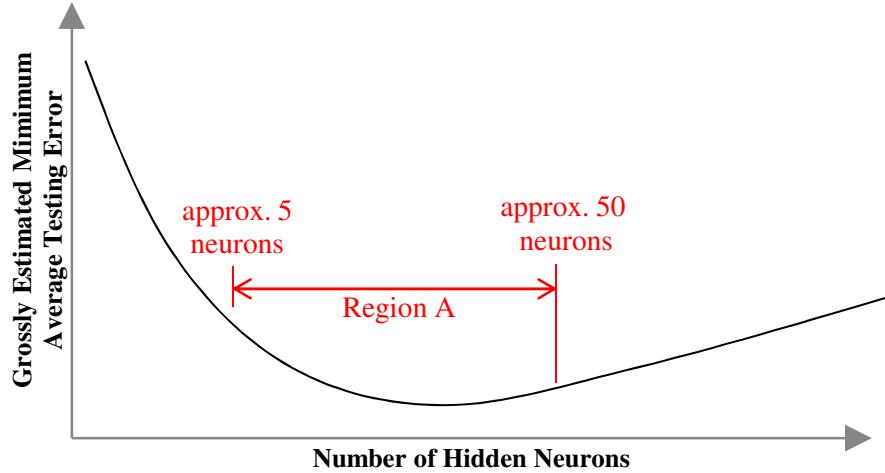


Figure 2.3: Graphical representation of the effect that the number of hidden neurons has for a 3-layer MLP ANN.

For this thesis, an assumption was made that the ideal number of hidden neurons that exists is most probably between five and fifty hidden neurons. This boundary was arbitrarily chosen from preliminary investigation results presented in Chapters 3 through 5, qualitatively verifying that as the number of neurons increased past fifty, there was a generalized logarithmic increase in the validation error. Finding the ideal number of hidden neurons that minimizes error is a significant challenge to ANN creation.

2.3.3 Generalizations about Structure on Training Times

A generality for training times is that as the number of hidden neurons decreases, there is generally a decrease in training times. However, faster training times could result in a significant increase in validation error (E_{valid}), or E_{test} . The value of training time is left to the end user's discretion to determine if reduced training is desirable. This topic is further explored in a relevant experimental result as shown in Section 5.2.2. For the purposes of this thesis, it is assumed that an ANN that takes longer than two minutes was over trained. Another factor that affects training times is N_{train} . If N_{train} is large, utilizing a

batch mode algorithm, training times are increased. This is discussed further in the following section.

2.3.4 Generalizations about Effective ANN Model Inputs

Different model parameters can be used as inputs, which could have varying effects on reducing the amount of error for an ANN. An ANN input should vary in the input data space. If the input never changes in the training set, then the input should not change in the validation and the testing set. Therefore, a static input would act as an additional bias in the ANN. Preliminary investigations found that additional inputs reduce error; however, a significant number of inputs (more than approximately ten inputs), causes the ANN to become over trained. The rationale is that more inputs require more neurons in the hidden layer, and the effects of additional hidden neurons were already discussed. Additional inputs also increase the training times. Preliminary investigations on increasing the number of inputs are examined in the simulations. Assuming the ANNs were successfully trained, it was found the error could be significantly reduced if non-obvious relevant inputs are included as shown in Section 3.1.9.

2.4 Training ANN Structure

The purpose of the training process for the ANN is to adjust biases and weights of the neurons within the ANN so as to minimize the training error. For case studies, a variety of training algorithms was utilized to find algorithms that gave the lowest f_{ann} error. Both preliminary investigations and case studies used a variety of different algorithms with a variety of different ANN structures and number of training epochs. Training algorithms that were used included the following: Backpropagation, Sparse Training, Conjugate Gradient, Adaptive Backpropagation, QN, QNM, Huber-Quasi-Newton, Auto Pilot

(MLP3), Simplex Method, Genetic algorithm, and Simulated Annealing algorithm. These algorithms adjust the network weights through a training process using the train data set, where the training set is $\{(\mathbf{x}_p, \mathbf{d}_p), p \in T\}$. Here, T represents the index set of training data. For the purpose of training, we define an error function $E(\mathbf{w})$ as:

$$E(\mathbf{w}) = \sum_{p \in T} [e_p(\mathbf{w})]^q, \quad (2.3)$$

where q represents the q^{th} norm, and $e_p(\mathbf{w})$ is the error due to the p^{th} sample given by,

$$e_p(\mathbf{w}) = \frac{1}{n} \left[\sum_{k=1}^n |f_{\text{ann},k}(\mathbf{x}_p, \mathbf{w}) - d_{pk}| \right]^{1/q}, \quad (2.4)$$

where d_{pk} is the k^{th} element of d_p , n is the n^{th} output of y , and $f_{\text{ann},k}(\mathbf{x}_p, \mathbf{w})$ is the k^{th} ANN output for input sample \mathbf{x}_p . The objective of ANN training is to find \mathbf{w}^* that minimizes $E(\mathbf{w})$. An example of training error is presented in Figure 2.4, and the training error will be further examined in Chapters 3 through 5.

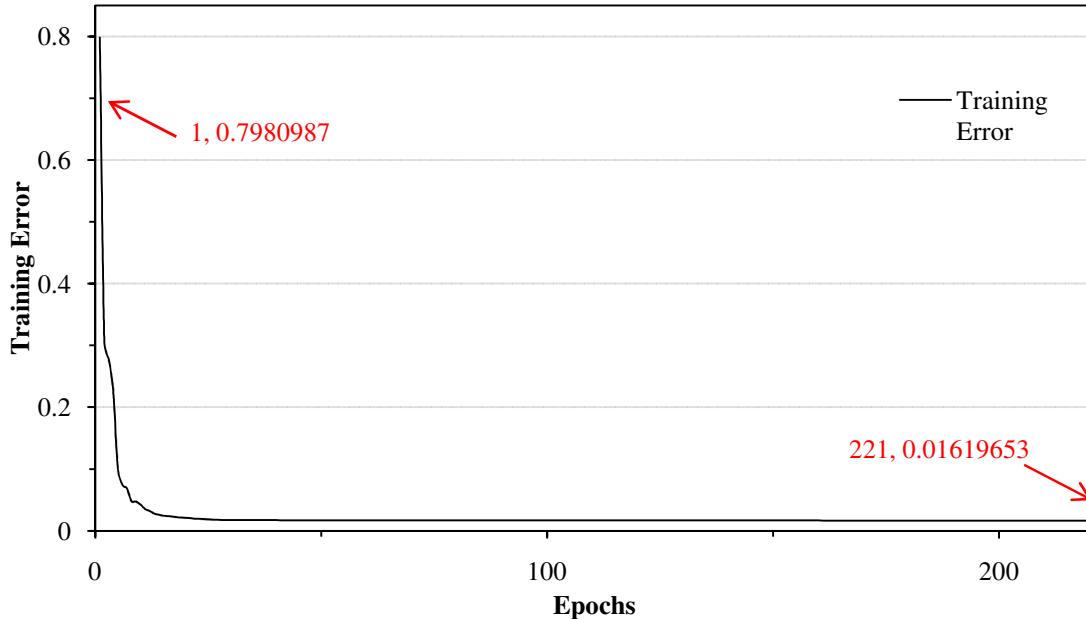


Figure 2.4: Training error for Chapter 3 first case study's f_{ann} , presented as an example of training error as a function of the number of training epochs.

As the figure shows, the error is large for the first few epochs and then reaches a steady-state level several epochs later, approximately around fifty epochs in. The training process is summarized in Figure 2.5.

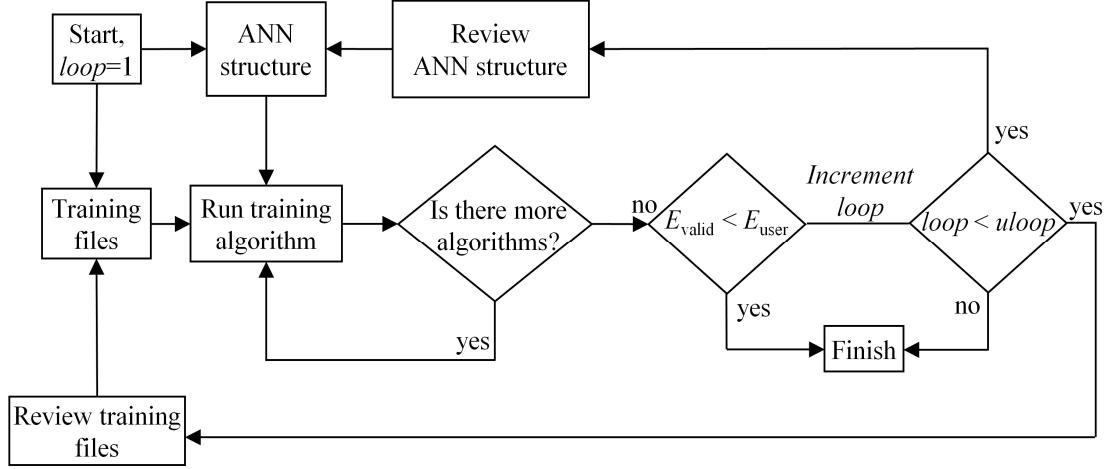


Figure 2.5: Feedback loop for ANN creation.

In Figure 2.5, $loop$ is the loop counter, $uloop$ is the user specified termination loop value, and error equation as utilized by *NeuroModeler* is E_{valid} .

NeuroModeler error equation is defined as:

$$E_{\text{valid}} = 100\% \cdot \sqrt{\frac{1}{N_{\text{valid}}} \sum_{p=1}^{N_{\text{valid}}} \left| \frac{f_{\text{ann}}(x_p, w) - d_p}{\max_{\text{train}} d_p - \min_{\text{train}} d_p} \right|^2}, \quad (2.5.a)$$

$$\max E_{\text{valid}} = 100\% \cdot \sqrt{\max_p \left| \frac{f_{\text{ann}}(x_p, w) - d_p}{\max_{\text{train}} d_p - \min_{\text{train}} d_p} \right|^2}, \quad (2.5.b)$$

where N_{valid} is the number of input and output vectors in a validation set, and the denominator is from the max/min d_p over the train set, and $\max E_{\text{valid}}$ is the maximum error from the set. Equations 2.5a and 2.5.b are used all through the thesis, and it should be assumed, unless otherwise stated, that the average error equation is calculated using equation 2.5.a. This can be utilized for testing and training error in addition to validation. *NeuroModeler* utilizes this error equation given that it is better for ANN evaluation (E_{train}) [51].

The number of epochs and train times is significant in determining if an f_{ann} is ready for validation and testing; after a certain point, these parameters have little to no correlation with the final errors. The number of epochs during training is largely based on how much w needs to be varied to find w^* . The number of training epochs were recorded; however, only generalizations were made about the training process in terms of number of epochs and the training times. The *AutoIt!* Program limited the number of epochs to approximately 6,000, although the ANNs rarely exceeded 2,000 epochs. Typically, E_{tol} was set to zero during training, and that forced the internal mechanisms inside of *Neuro-Modeler* to arbitrarily end training.

2.4.1 Important Training Pairs to Include for Training

Beginning end points in the input data space, as well as input regions in the data space where $d^2(f_{\text{ann}}(\mathbf{x}, \mathbf{w})) / d\mathbf{x}^2$ were a large value, are critical for keeping error minimized in an expected data set range. Investigations were completed that examined how a cropped input training and testing set might reduce f_{ann} error; however, nothing substantial was discovered. The significance of an outlier was examined in Section 3.3.1. Investigations on variations to the training sets, covered in Chapters 3 through 5, show that although the significance of some training points is immediately obvious, the significance of some p -samples is not immediately obvious with respect to f_{ann} error.

2.4.2 Generalizations on the Size of the Training and Validation Sets

Experimentally, it was found that large training data sets help reduce ANN error; however, large training sets are not necessarily required, which means ANNs are well suited for applications where data is limited. In preliminary investigations examined in Section 3.1.8, N_{train} was varied. Samples were arbitrarily chosen and moved between data sets if

$d^2(f_{\text{ann}}(\mathbf{x}, \mathbf{w})) / d\mathbf{x}^2$ had a small value. It was found that these points had a large effect on E_{valid} . Shown in Figure 2.6 is an overview of the training process with respect to the data sets.

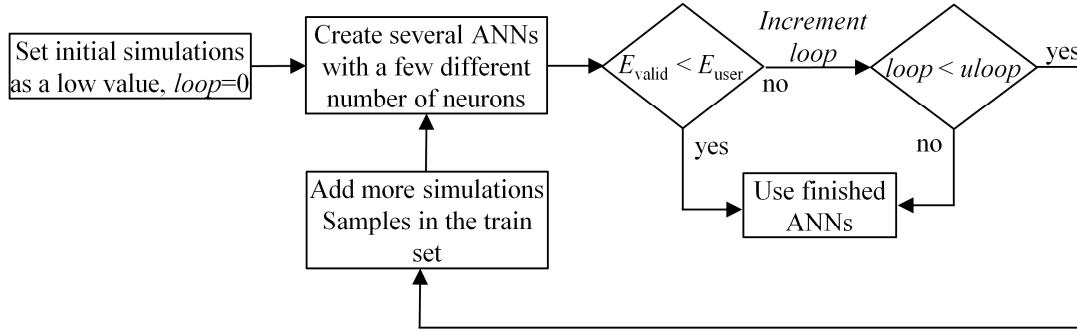


Figure 2.6: Feedback loop for training set creation.

In Figure 2.6, E_{tol} is defined as the error tolerance of the user. However, further examinations of an *exhaustive* approach to creating f_{ann} s with the lowest error are beyond the scope of this thesis. It is suffice to generalize that training sets presented could be further reduced without large changes to error.

2.4.3 Generalizations about Using Multiple Training Algorithms

As mentioned previously, several of the available algorithms can be utilized sequentially in order to provide faster training or lower average E_{valid} and E_{test} . Reviewing Figure 2.5, the combinations of algorithms that are utilized for all case studies in Chapters 3 through 5 are QN, QNM, CQN, and CQNM. If N_{train} is sufficiently large, such that the training times are prohibitively long, then a back propagation technique could be utilized. The recommendation is that for large sets of data or large structures, a conjugate gradient algorithm is meant to quickly optimize the randomized initial weights and biases such that batch mode training algorithms (QN and QNM) complete in fewer epochs, leading to possibly reduced training times and lower validation errors. The value of combinations of training algorithms is examined in Chapters 3 through 5 as well as the effect of combina-

tions of multiple training algorithms. The effects of single and multiple algorithms on training times are shown in Section 5.2.2.

2.5 Testing and Re-iteration

ANN testing is a critical final phase of ANN creation. In this phase, the f_{ann} under test will have the error results of the training, validation, and testing data sets evaluated. With these unique results, an opportunity is presented for critical review of the ANNs as shown in Figure 2.5 and Figure 2.6. The review is utilized to determine necessary changes to the ANN creation process. These changes will be expressed in greater depth in an experimentally relevant context in Chapter 3; however, to summarize the changes from analyzing the error, the resulting changes could be inputs and outputs; training, validation, and testing data sets; ANN structure; and ANN training procedure.

2.5.1 Error Equation Comparisons

An important factor in evaluation of ANNs is the definition of error. In this section, different error equations are presented. The average error equation utilized by *NeuroModeler* was presented in Equation 2.5. Another error type utilized for entire data sets is called Standard Error (s.e) and is defined as follows:

$$s.e = \sqrt{\frac{\sum_{s=1}^m \sum_{i=1}^n y_{is}^2}{(n_y - 1)(n_y)}}, \quad (2.6)$$

where for this equation exclusively, s is the series number, i is the point number in the series s , m is the number of series for point y in chart, n is the number of points in each series, y_{is} is the data value of series s and the i^{th} point, and n_y is the total number of data values in all series [53].

Another error equation utilized for validation is that of Root Mean Square Error (RMSE). RMSE was utilized for a subsection of a few experiments given that it is more

recognized. Given the popularity of root mean squared (RMS), it was utilized for comparison purposes primarily to examine the numerical difference between equation 2.5.a and 2.6. This comparison is made in Section 3.6. The RMSE is given by the following:

$$E_{\text{rms}} = 100\% \cdot \sqrt{\frac{1}{N} \sum_{p=1}^N \left| \frac{f_{\text{ann}}(x_p, w) - d_p}{d_p} \right|^2}, \quad (2.7)$$

where E_{rms} is Root Mean Square Error. In the case study on reverse modeling, the magnitude of the error for RMSE and *NeuroModeler* was similar; however, there was consistently a small numerical difference between the error equations. This difference in percentages can be accounted for by the denominator within the square, which is the min/max d_p values found in the training set for Equation 2.4.a as opposed to d_p for Equation 2.7.

The percent error equation is utilized in several sections and is an appropriate equation to compare only two data points. The equation for percent error is shown in Equation 2.8. The percent error is given by:

$$E_{\%} = 100\% \cdot \left| \frac{y_{\text{ann}} - d_p}{d_p} \right|, \quad (2.8)$$

where $E_{\%}$ is percent error. This has been utilized in Chapter 3.

To summarize, the available error equations E_{train} , E_{valid} , and E_{test} are used for testing of entire data sets. The max E_{train} , E_{valid} , and E_{test} are other criteria that could be used for accepting or rejecting ANNs based on user criteria. The RMSE is used for comparison purposes for testing the entire sets. Percent error was used to show the difference between data sample values.

2.5.2 Generalized Error Patterns Encountered While Testing

From preliminary investigations, it should be noted that the maximum error of the ANN

also needs to be considered if the error is in a region of interest or not. Often in the d_p samples, it was found that as the radiated emissions became numerically small (on the order of approximately -240 dB), settings for the *HFSS* software's auto-mesher could have had higher systematic numerical error. Because of this, at very small field values, derivatives ($|E|$ with respect to position) could vary rapidly and can thereby cause regions with almost singularity points, which, given the nature of ANNs, probably had a higher error in those regions with rapid changes in $|E|$'s second derivative with respect to location. Therefore, error at small magnitudes should carry less significance than at larger magnitudes. This was observed through qualitative means and descriptions. For an investigation on recommendations for minimizing error, see Section 3.1.1. Also, given the nature of design constraints and user tolerances, larger errors in insignificant regions should carry less clout in determining whether to use a f_{ann} for EMC simulations.

With this background, this thesis investigates ANN modeling capabilities for some very important and common EMI/EMC scenarios involving PCB traces, shielding enclosures, and PCB interconnects.

2.5.3 Practical Limitations to Exhaustive ANN Creation/Testing

Based on an examination of all the parameters for f_{ann} creation, testing all the variations that could be done to ANNs would be impractical. In order to determine general constraints for the ANN modeling, a list of variables is shown below in Figure 2.7. It may not be obvious that in order to do an exhaustive ANN creation, there must exist a large number of combinations, making it arduous/computationally prohibitive to execute a complete set of variations to the combinations available in *NeuroModeler* v1.5.

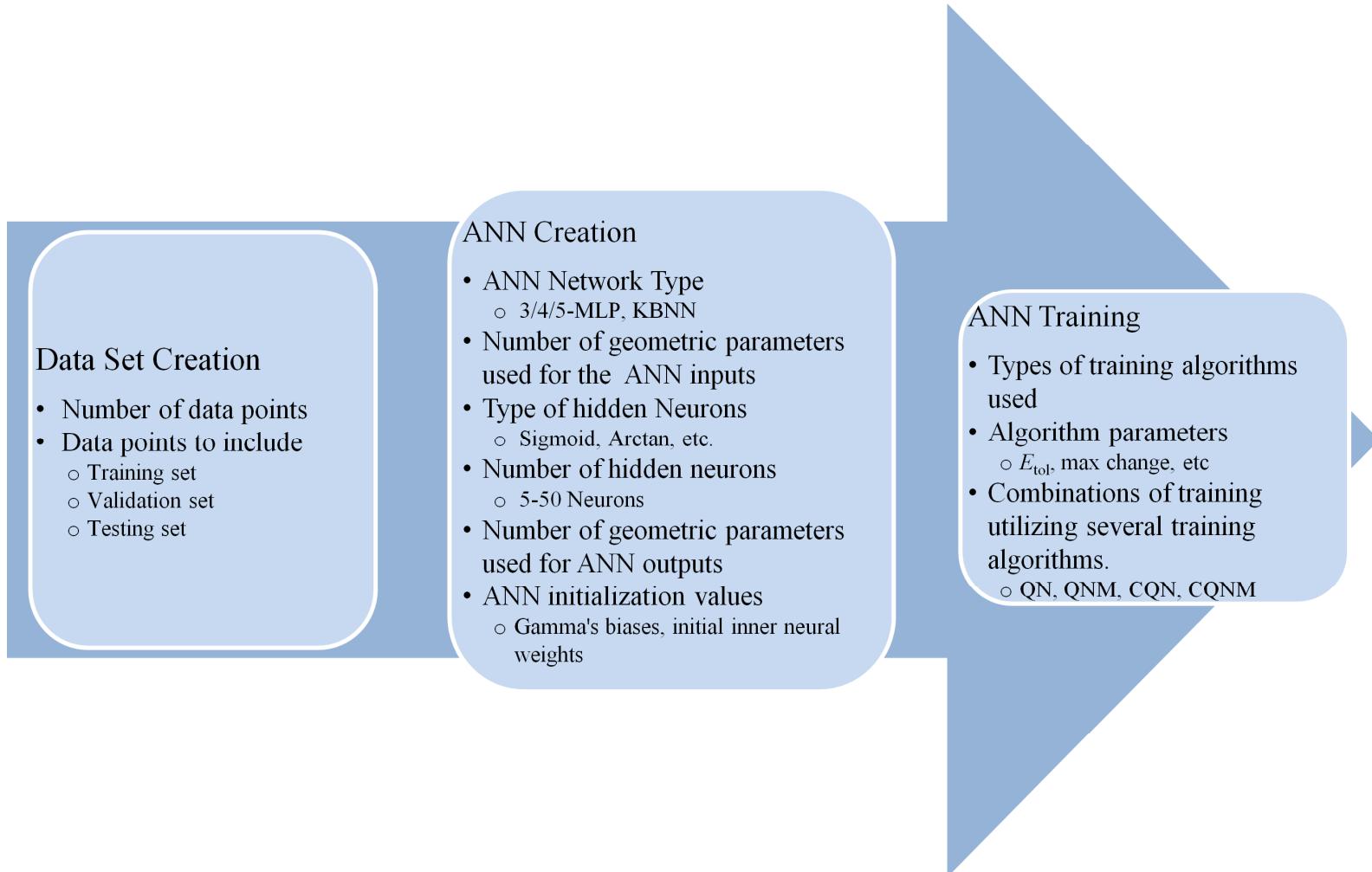


Figure 2.7: Figure of available combinations.

Hypothetically, if a data set of one hundred data samples were to be used for ANN training, validation, and testing, and the data could be split into three data sets (assuming sixty samples are used for training and twenty samples are used for validation and testing), there would be 1.3746e+028 combinations given by the following:

$$nCr^{\text{Set Creation}} = nCr^{\text{Train}} \binom{100}{60} + nCr^{\text{Valid}} \binom{40}{20} + nCr^{\text{Test}} \binom{20}{20}, \quad (2.9)$$

where $nCr^{\text{Set Creation}}$ is the number of sets, nCr^{Train} is the number of training combinations, nCr^{valid} is the number of validation sets, and nCr^{Test} is the number of testing combinations.

From preliminary investigations and case studies, if an ANN could be trained and tested every sixty seconds, to test all those combinations it would take 2.6153e+022 years. If the age of the universe is 13.75 billion years old [54], it would take approximately a hundred trillion times the age of the universe to finish training an f_{ann} for all the possible data set combinations. Given temporal constraints, a thorough variation to the data set creation is impractical, meaning local minimums are to be assumed. In *NeuroModeler* v1.5, there are seven different template ANN network types, thirty-six different types of ANNs, approximately fifty practical numbers of hidden neuron combinations, approximately five ANN inputs and outputs that could be defined for any given geometry, and an infinite number of initialization values for the network. It can be seen that exhaustive variations to ANN creation is impractical. ANN training has a similar large number of parameters (eleven training algorithms, six different algorithm settings, practically up to ten combinations of using multiple training algorithms).

In order to find a user-defined acceptable EMC f_{ann} , the user must specify criteria to evaluate f_{ann} s. For this thesis, typically f_{ann} s were chosen based on lowest aver-

age/maximum train/valid/testing error, where 300-400 distinct ANNs were created for each case study.

2.6 Reverse Modeling Algorithm

The accuracy of ANN models depends on the accuracy and size of the available data. However, in EMC design scenarios, only a limited amount of training data is available. The resulting ANN models may not meet user expectations and therefore cannot be relied on when making EMC design decisions.

Consider a scenario where the standard approach fails to generate f_{ann} satisfying $E_{\text{test}} \leq E_{\text{user}}$, where E_{user} represents the desired tolerance or accuracy from an acceptable model. Now, let us define two neural models, namely the desired model f_{ann} and a correction model $f_{\text{ann},c}$. While the structure of f_{ann} itself is based on the given modeling problem, the structure of $f_{\text{ann},c}$ is not known *a priori*. Let

$$y = f_{\text{ann}}(x_1, x_2, \dots, x_m) \quad (2.10)$$

represent the desired model. Now define a set of potential correction models as:

$$\begin{aligned} x_{1,c} &= f_{\text{ann},1}(y, x_2, \dots, x_m) \\ x_{2,c} &= f_{\text{ann},2}(x_1, y, \dots, x_m), \\ &\vdots \\ x_{m,c} &= f_{\text{ann},m}(x_1, x_2, \dots, y) \end{aligned} \quad (2.11)$$

where $f_{\text{ann},i}$ and $x_{i,c}$ represent the i^{th} potential correction model and its output, respectively.

The outputs of these correction models are inputs of the desired model as shown in Figure 2.8.

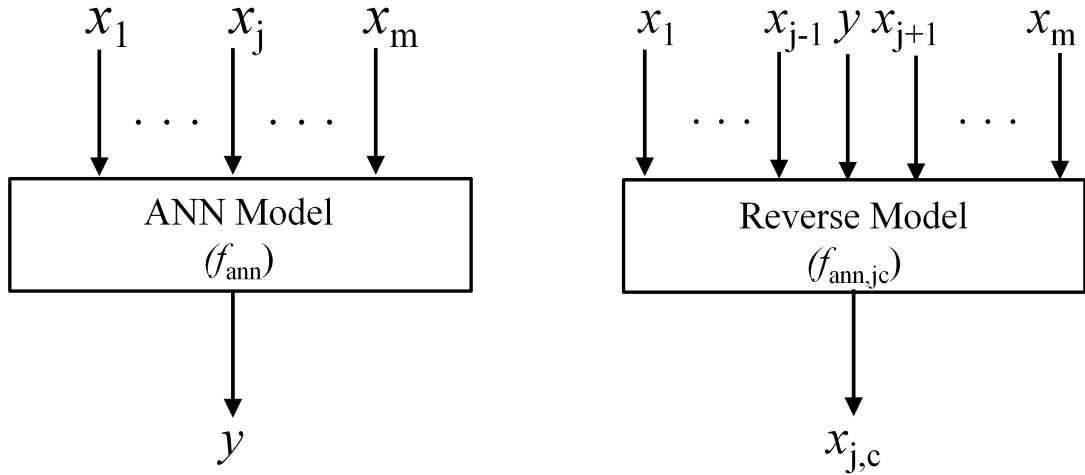


Figure 2.8: Input and output relationship as represented in a desired and in a correction ANN model.

Therefore, the important step in this approach is identifying the model to be used as the correction model.

Consider the potential correction model $f_{\text{ann},1}$ after rearranging the training data accordingly. Given a rearranged sample as input, $f_{\text{ann},1}$ provides an output $x_{1,c}$ which closely approximates x_1 . Quality measure $E_{\text{train},1}$ of $f_{\text{ann},1}$ is then evaluated. This process is repeated m times, resulting in m potential correction models with corresponding error measures $E_{1,c}, E_{2,c}, \dots, E_{m,c}$, respectively. $E_{m,ic}$ are derived from using equation 2.5.a. The correction model $f_{\text{ann, jc}}$, with the least error, is then selected using:

$$j = \arg (\min_i E_{i,c}) \quad (2.12)$$

Once a model with the least error is selected, it is used to estimate the corrected desired output, y , as follows:

Step 1: Given a new data (x_1, x_2, \dots, x_m) , f_{ann} is used to determine an approximate output y . Initialize, $\Delta y = \alpha > 0$ and $h = 1$. Where α is a scalar, Δy is the change to the output.

Step 2: Data inputs $(x_1, x_2, \dots, x_{j-1}, y, x_{j+1}, \dots, x_n)$ are fed to the correction model $f_{\text{ann}, j}$ leading to an output $x_{j,c}$.

Step 3: Evaluate the objective function, $E_{\text{obj}} = \frac{x_j - x_{j,c}}{|x_j|} \times 100$. If $|E_{\text{obj}}| \leq E_{\text{user}}$, RETURN y

(*i.e.*, neural model output). E_{user} is defined as the tolerance set by the user.

Step 4: Set $y_1 = y - \Delta y$, $y_2 = y + \Delta y$,

$$a = f_{\text{ann}, j}(x_1, \dots, x_{j-1}, y_1, x_{j+1}, \dots, x_n),$$

$$b = f_{\text{ann}, j}(x_1, \dots, x_{j-1}, y_2, x_{j+1}, \dots, x_n),$$

$h_{\text{new}} = \text{sign}(E_{\text{obj}})$, and $\delta = \text{sign}(b-a)$. Here, a and b is defined as the intermediate calculations, h is the sign of E_{obj} , and δ is the sign of the intermediate calculations.

Step 5: If $h \neq h_{\text{new}}$, set $h = h_{\text{new}}$ and $\alpha = \alpha / 2$.

Step 6: Set $\Delta y = \alpha * h * \delta$ and $y = y + \Delta y$. GOTO Step 2.

The process of $F_{\text{ann}, c}$ can be summarized in a flow chart, as illustrated in Figure 2.9, where $F_{\text{ann}, c}$ is the y value returned from the reverse modeling algorithm.

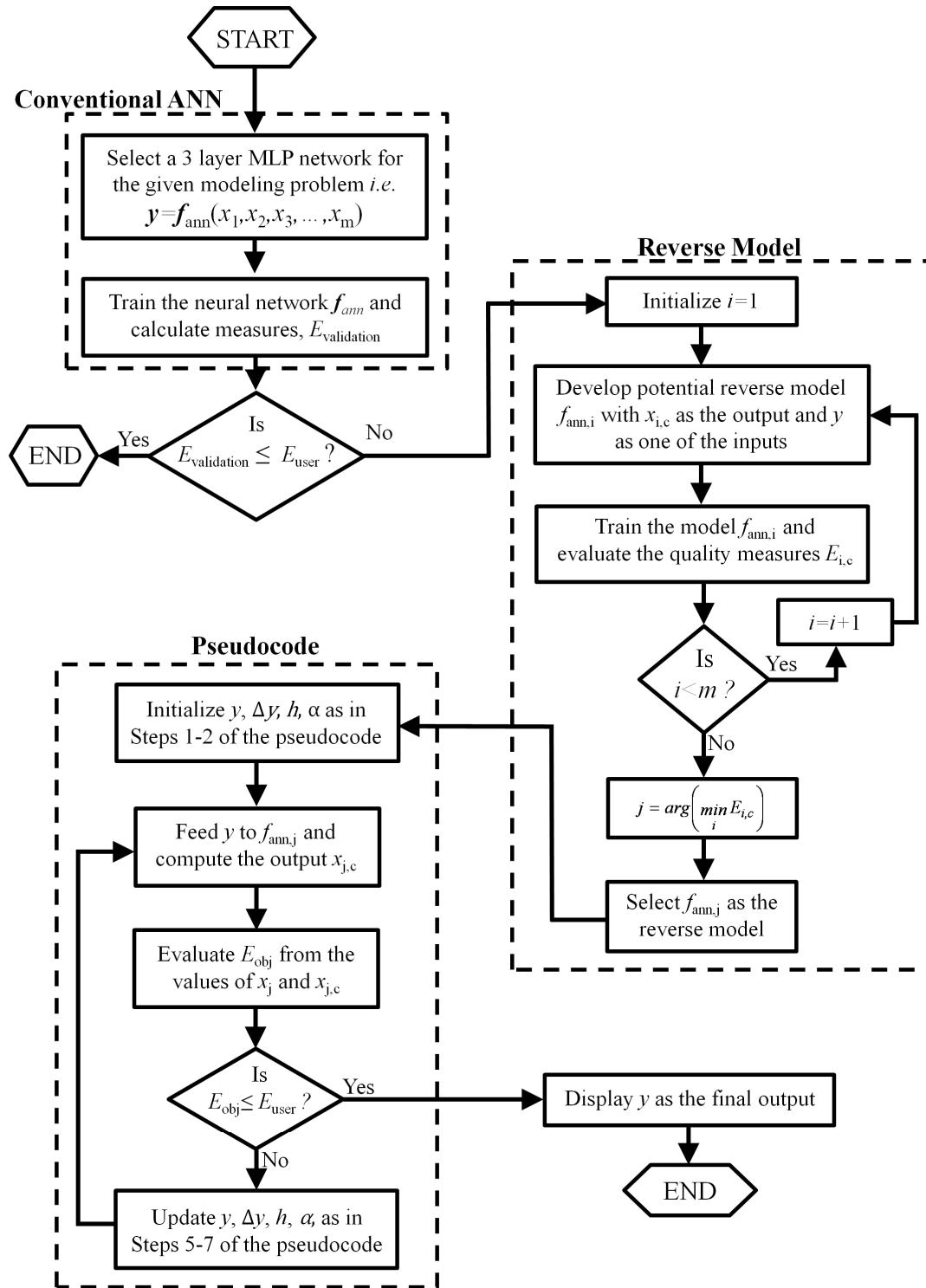


Figure 2.9: Flow chart illustrating the proposed correction algorithm.

The reverse modeling algorithm given in Figure 2.9 is performed on a sample-by-sample basis. In Section 3.6, the reverse modeling approach was applied on an arbitrary f_{ann} from Section 3.4 in order to reduce the error of the desired model. To this end, a reverse modeling algorithm is shown that could increase the accuracy of f_{ann} models using the limited data.

Chapter 3

EM-ANN Modeling of EMI Emissions from Perforated Cavities

This chapter discusses two EMC design case studies using the proposed ANN approach. The first problem models an EMI source's radiation as its location inside a shielding enclosure and the enclosure's apertures are varied. In the second case study, ANN models are employed to predict radiated emissions for variations to the number of apertures on a PCB shield. The holes are added for ventilation or access purposes and may act as antennas that release radiation outside the cavity [55] [45] [56] [57] [58] [59]. An overview of the ANN creation process can be seen in Figure 3.1. The case studies in this chapter are idealizations, and dimensions were chosen in terms of the solution frequency wavelength such that distinct modes could be seen in the radiated output. Preliminary investigations showcase how adaptive ANNs are; as network parameters are modified, such as different input sets, algorithms, initializations, and structure, the error still remains relatively low. As mentioned in Chapter 2, most ANNs created were done using *AutoIt!*. Non-idealized problems, such as Laird Shields [60], would be impractical to directly import into *HFSS* and are beyond the scope of this thesis.

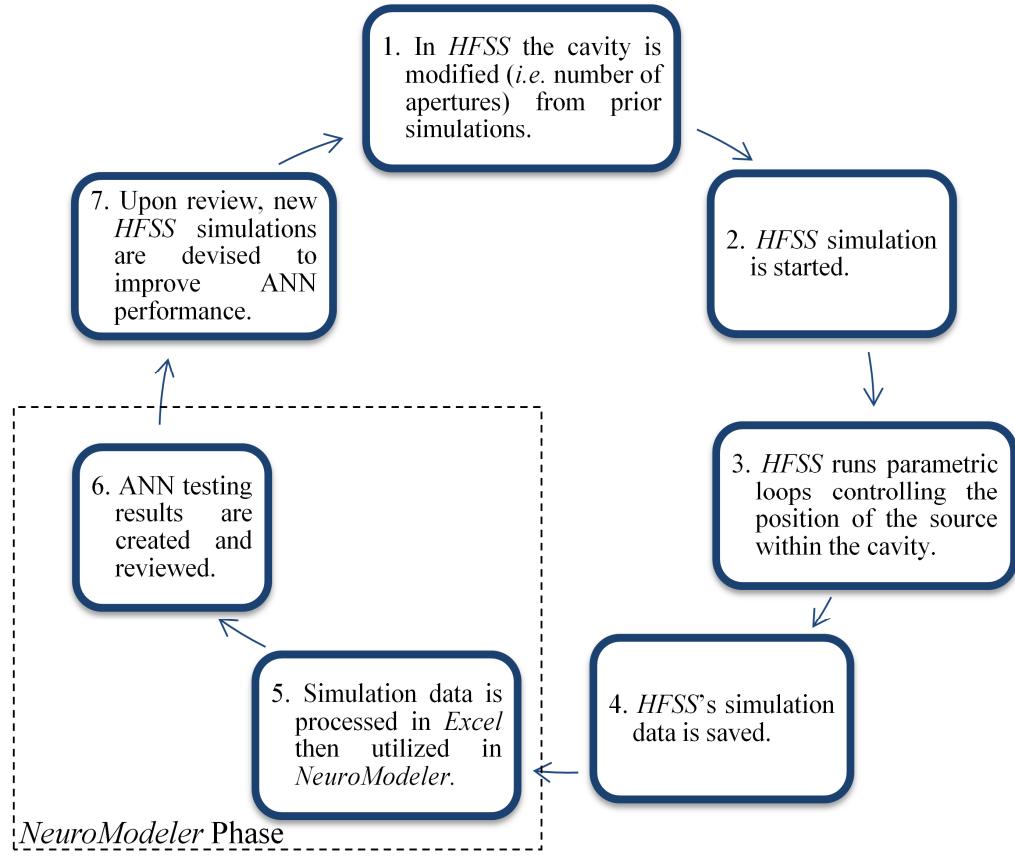


Figure 3.1: Overview of ANN creation from physics simulator to testing.

3.1 Preliminary Investigations for Case Study One

EMC shields have a natural tradeoff between general *shielding effectiveness* (SE in dB) vs. thermal considerations. The thermal issue with shields is that the air flow rate over the components goes to zero as the SE increases, which would require components to sustain higher temperatures. This can be represented by the formula below:

$$SE \propto \lim_{A_{\text{norm}} \rightarrow 0} \text{Ideal SE}(1 - A_{\text{norm}}), \quad (3.1.a)$$

$$\dot{Q}^{\text{shield}} \propto \lim_{A_{\text{norm}} \rightarrow 0} A_{\text{norm}} * \dot{Q}^{\text{air}}, \quad (3.1.b)$$

where A_{norm} is the normalized area of the apertures, \dot{Q}^{air} is the ambient (air) rate of heat transfer, and \dot{Q}^{shield} is the heat transfer out of the shield. Equation 3.1.a is simplified form

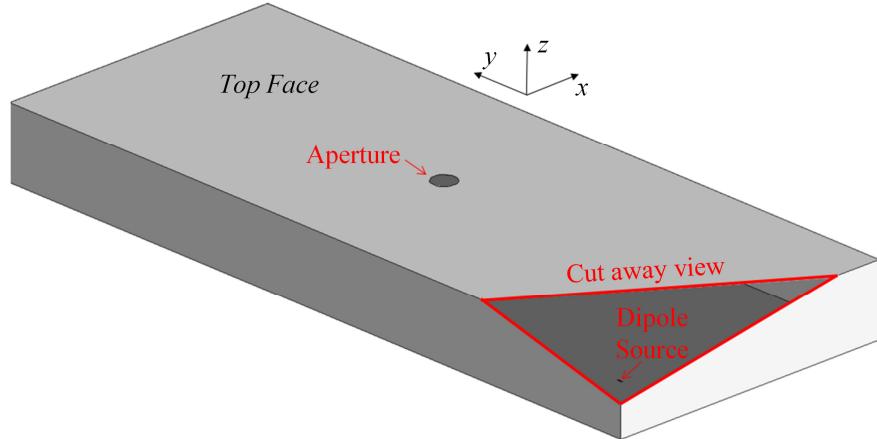


Figure 3.2: Preliminary one arbitrary aperture diagram with a cutaway view of the source oriented in the y-direction.

of Newton's Law of Cooling [61]. As Equation 3.1.a and Equation 3.1.b show, as A_{norm} goes to zero, the SE becomes more ideal; however, the air mass flow rate drops to zero, thereby increasing the temperature. Thus, there is a desire to balance the normalized area and the SE. There are methods to control emissions, such as increasing the number of apertures and decreasing the diameters of apertures, or material changes [60] [61] [62] [63] [64] to increase the SE while keeping A_{norm} constant; however, this is beyond the scope of this thesis. Shown in Figure 3.2 is one of the arbitrarily chosen aperture simulations utilized in the preliminary experiments. The other configurations with arbitrary aperture locations were 4-, 6-, 8-, 9-, 14-, and 16-aperture configurations. The planar surface with the apertures cut out is referred to as the *top face*. The PEC enclosure is 12 x 30 x 3 cm, and it encloses a 1.55 GHz dipole source with a magnitude of 1V oriented in the y-direction. The measurement point varies with preliminary experiment; however, it is measured at one of the two locations: 1) near field (6 cm, 15 cm, 6 cm), or 2) far field (6 cm, 15 cm, 103 cm).

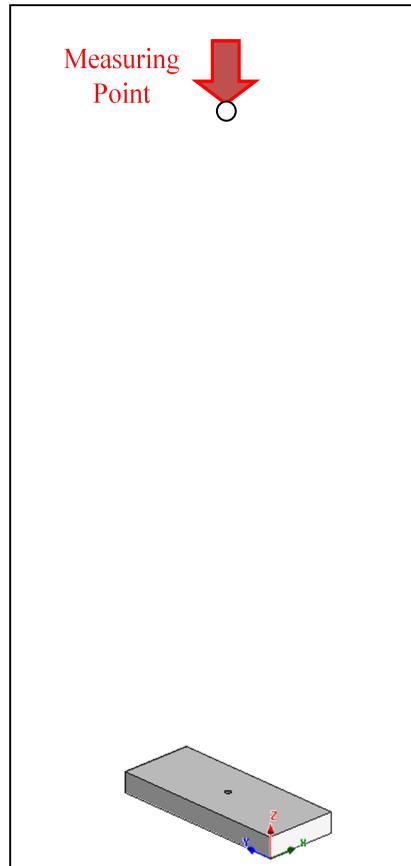


Figure 3.3: Far field measuring point for a one-aperture preliminary investigation on how a source's position affects radiated electric fields.

The measuring point, as shown in Figure 3.3, and direction are also dependent on preliminary experiments with the value being either $|E_x|$ or $|E_y|$, where $|E_x|$ is the absolute value of the radiated field with respect to the x -axis. A summary of the measured d , $|E_y|$ is found in Table 3.1.

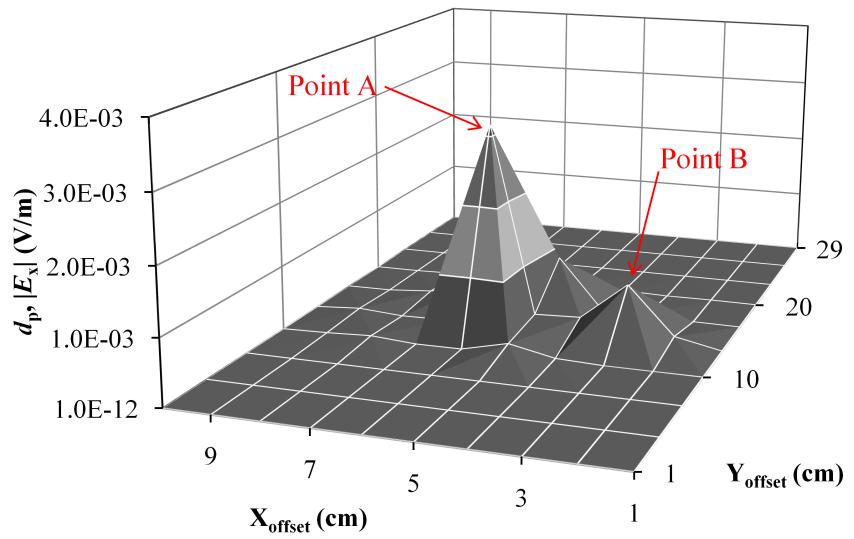
Table 3.1: Summary of d_p for $|E_y|$ values for preliminary investigation on arbitrary aperture geometries with source location variations.

Number of Holes	Minimum $ E_y $ (dB V/m)	Maximum $ E_y $ (dB V/m)	Average $ E_y $ (dB V/m)	Standard Variance of the $ E_y $ (dB V/m)	Typical ANN Use (Train or Test)
1	-270.9	-134.3	-205.0	32.1	Train
2	-231.0	-134.5	-180.6	22.2	Test
4	-224.7	-134.7	-172.3	22.8	Train
6	-202.1	-134.0	-161.0	15.5	Train
8	-188.6	-134.2	-155.3	11.8	Test
9	-199.3	-133.3	-157.1	15.5	Train
14	-171.6	-134.0	-149.1	8.2	Test
16	-182.1	-132.3	-148.5	11.1	Train

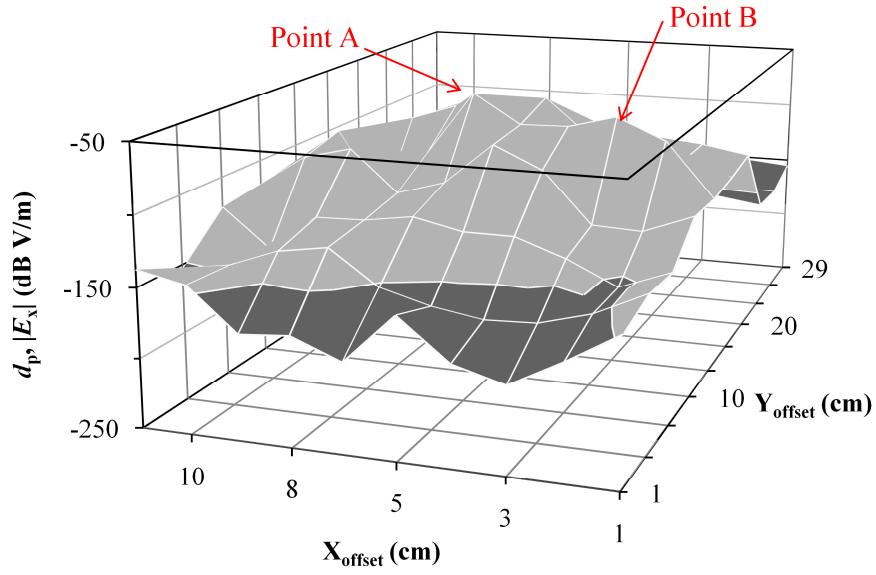
The minimum radiation is a function of source position within a cavity; however the maximum radiation for a given aperture configuration is very similar, within about one dB V/m, as shown in Table 3.1. Also, the geometric configuration for apertures two, six, and fourteen was used interchangeably in training and testing sets. All other configurations were used for training sets.

3.1.1 Various Tests to ANN I/O Structure – Magnitude vs. dB

Preliminary experiments for case study one were done such that the radiated fields were measured several centimeters from the *top face*. It was later determined from the ANN results that rapid changes, such as those found in magnitudes shown in Figure 3.4, in the near field had derivatives that changed too rapidly for 3-layer MLP networks. In Figure 3.4 (a) and (b), points A (6, 13 cm) and B (3, 13 cm) have the same source location; however, the magnitude $|E_x|$ is reported in dB in Figure 3.4 (b). From Figure 3.4 (a), it can be concluded that the values in the training set for d_p changed too rapidly, and this will be shown in subsequent subsections.



(a)



(b)

Figure 3.4: Near field, one-hole aperture for the preliminary d_p values experiment (a) V/m (b) same data but in dB V/m.

A preliminary estimation had an f_{ann} trained to estimate a near field estimate of $|E_x|$ (V/m) from a two-aperture geometry. The f_{ann} had 20 hidden sigmoid neurons trained utilizing a QNM algorithm. Shown in Figure 3.5 is a contour graph of the percent error comparing d_p values to f_{ann} estimates as a function of X_{offset} and Y_{offset} .

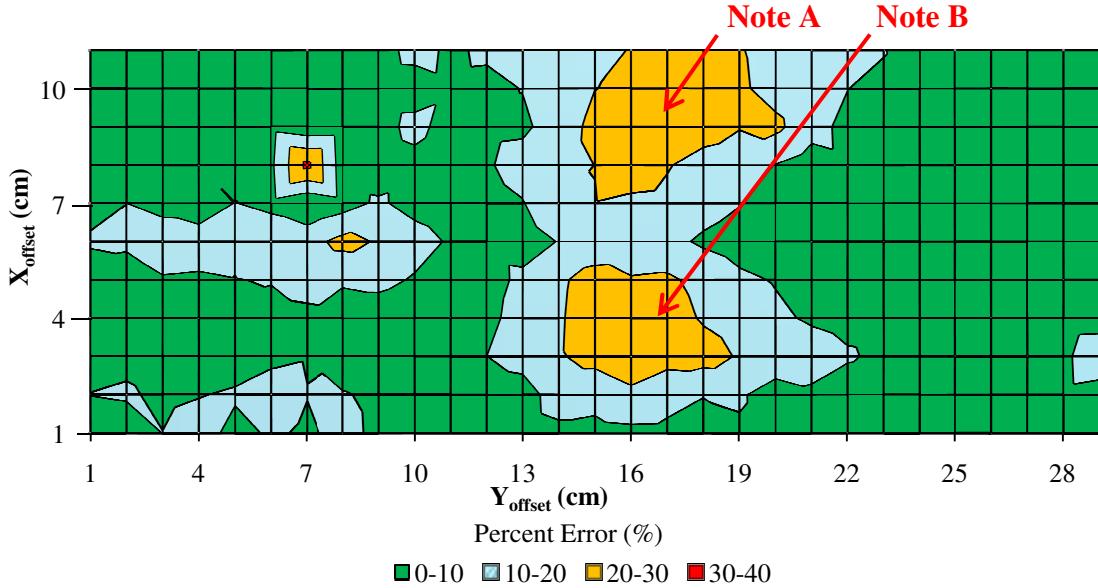


Figure 3.5: Percent error surface of $|E_x|$ as calculated from a sigmoid neuron type ANN in the near field.

As illustrated in Figure 3.5, the data changes sporadically in several regions, such as Note A and B. Given the large derivatives for $d^2|E_x|/d^2x$ or $d^2|E_x|/d^2y$, the trained ANNs had an unacceptable average and max E_{train} and E_{test} . This is similar to Figure 3.4; however, it does model a different geometry. The significance of Figure 3.5 is that if f_{anns} try to model phenomena that are almost discontinuous, the f_{ann} 's error increases. For this example, the average E_{train} was 3.531%, and the E_{test} was 12.025%. This is further examined in Section 3.1.3. As shown in Balanis [65], the near field values change extremely rapidly. Recognizing that the near field could not be modeled with current ANN training techniques, a decision was made to conduct a preliminary simulation in the far field.

3.1.2 Various Tests to I/O – Near vs. Far Field

The far field simulations had several advantages; however, they did increase simulation complexity in HFSS. The advantages to the far field were smaller derivatives for $d^2|E_y|/d^2x$ or $d^2|E_y|/d^2y$ such that the values $d|E_y|/dx$ appeared less sporadically, *i.e.*, continuous across the input data space. For instance, Figure 3.4 (b), which is a one-aperture

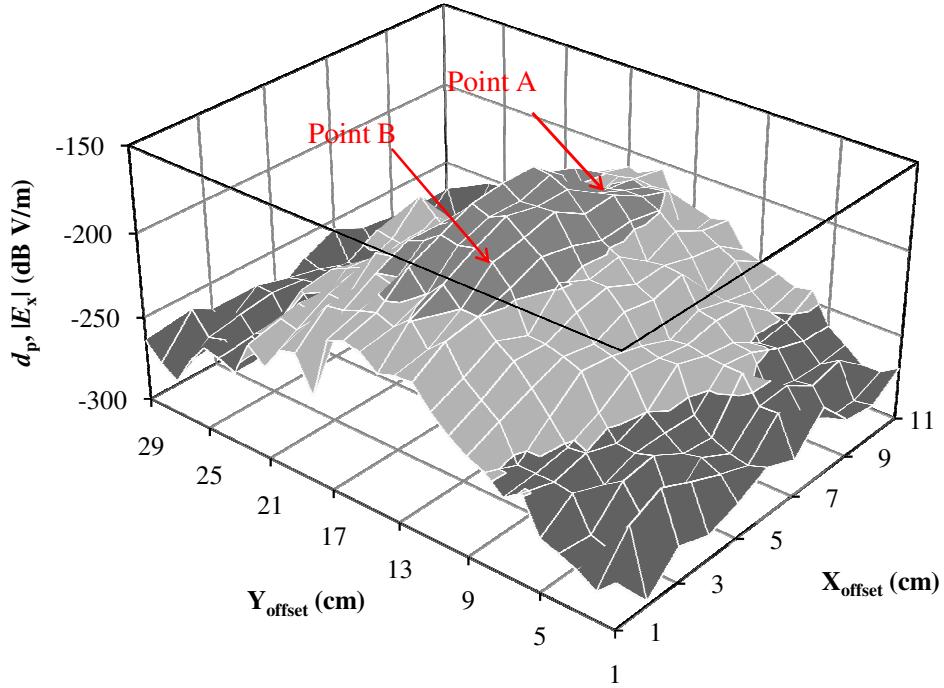


Figure 3.6: Far field d_p values, $|E_x|$, for a one-aperture preliminary experiment.

geometry, was re-simulated, but in this case, the measured values are in the far field, and this is shown in Figure 3.6. The comparison of the f_{ann} created for the near and far fields is shown in Figure 3.7. The near field f_{ann} had twenty sigmoid hidden neurons trained by a QNM algorithm utilizing a smaller N_{train} and N_{test} (by about two hundred and twenty samples) than the far field f_{ann} . The far field f_{ann} had forty sigmoid hidden neurons trained by QNM. Preliminary experimental results based off of Figure 3.6 are presented Figure 3.7. Note that the oscillations in Figure 3.7 are not significant; rather, they are a coincidence from the data set creation.

Figure 3.7 is not a direct comparison due to different directions of measurement and different sizes of training and testing sets. This is acceptable because Figure 3.7 shows how much different the f_{ann} s were from the d_p values for a given sample point.

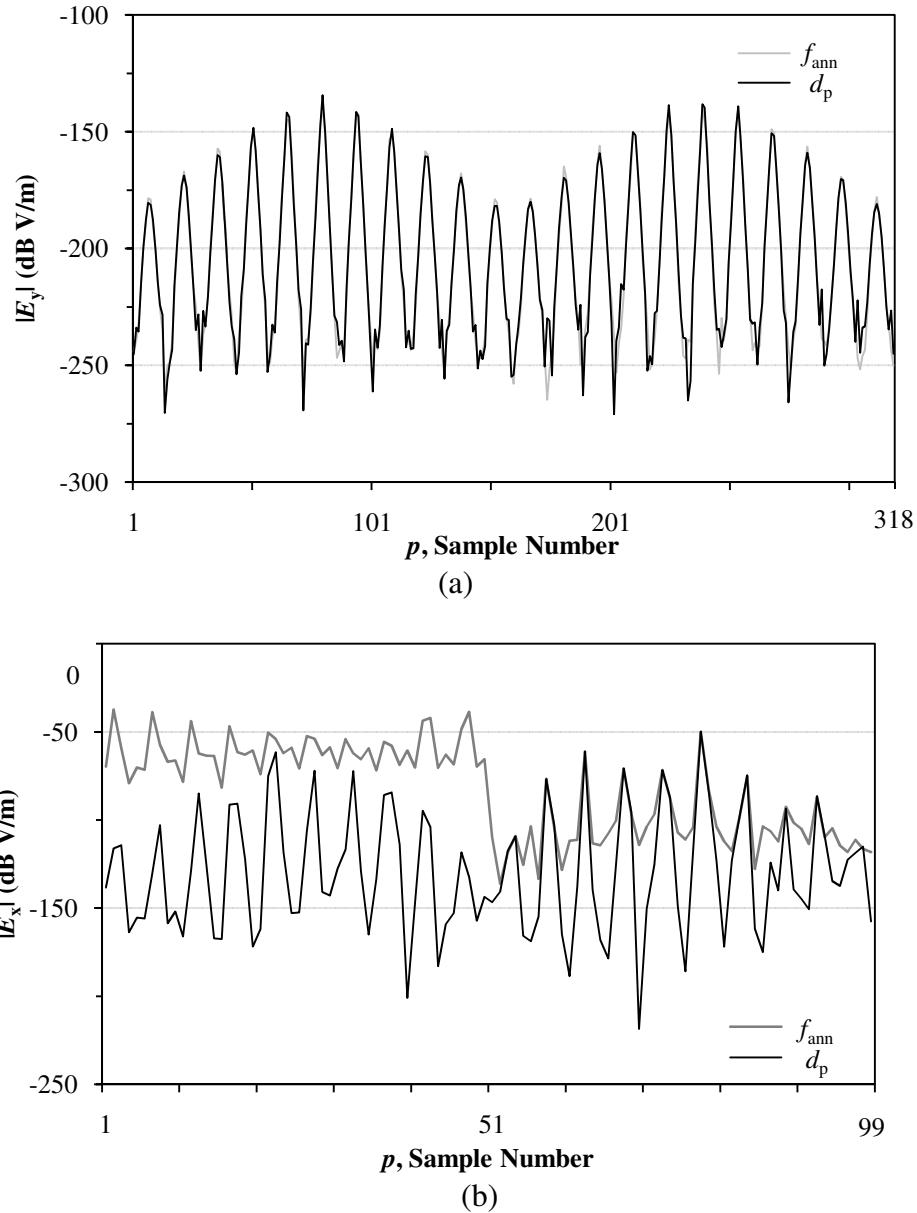


Figure 3.7: Comparison of f_{ann} estimates of $|E_y|$ where (a) is the far field and (b) is the near field.

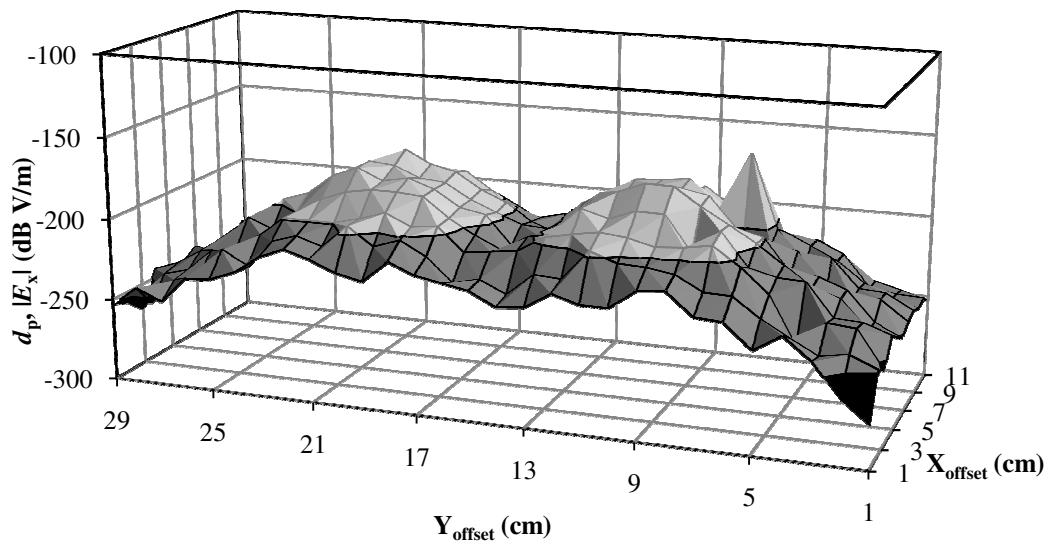
The sample points in Figure 3.7 had the training, validation, and testing data sets combined to show the f_{ann} estimates for all calculated d_p values. The difference between part (a) and (b) is that part (b) was the first preliminary experiment done, and as such, fewer samples were collected as shown on the axis. The values were measured in the $|E_x|$ direction, and part (a) had an additional input, $|E_0|$. Part (a) benefits fully from the recommen-

dations established in this section. The difference in E_{test} is significant: part (a)'s $E_{\text{test}} = 2.74\%$ and part (b)'s $E_{\text{test}} = 31.26\%$, which is more than an order of magnitude reduction in error.

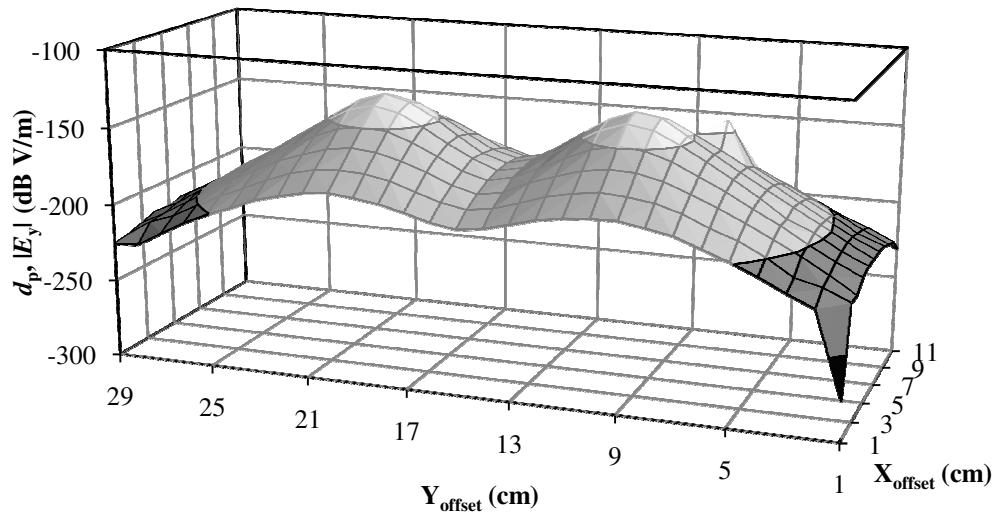
An advantage to a far field measurement is that far fields measurements are typically more utilized in high-frequency measurements. A disadvantage is that the computation load increases when the radiation boundary is extended to the far field. In general, this means that there is a significant increase in the number of tetrahedrals in the mesh, which increases the size of the matrix that *HFSS* has to invert for each adaptive pass. However, as the data show, modeling radiated emissions from the near field to the far field yielded ANNs with a significantly lower average and maximum training/testing errors for a given aperture configuration.

3.1.3 Various Tests to I/O – Measurement Direction

Figure 3.8 is the d_p values for two different components, specifically in the Figure 3.8 (a) x -direction and Figure 3.8 (b) y -direction for an arbitrary two-aperture configuration. The surfaces presented show that Figure 3.8 (b), in the y -direction, has a more continuous surface, and as such, it became of interest to examine how training in different directions affects the error of EM ANNs. The E_{test} results of training ANNs for arbitrarily chosen aperture geometries for 1, 4, 6, 9, and 16 apertures is shown in Table 3.2.



(a)



(b)

Figure 3.8: Comparison of $|E_x|$ and $|E_y|$ from the physical solver, d_p , where (a) is $|E_x|$ and (b) is $|E_y|$.

Table 3.2: Preliminary investigation comparison of $|E_x|$ and $|E_y|$ for creation of sigmoid ANNs.

Hidden Neuron Type	Training Direction	Training Epochs	Average $E_{\text{train}} (\%)$	
Sigmoid	x -dir	1610	3.53	
Sigmoid	y -dir	505	2.07	
Hidden Neuron Type	Testing Direction	Average $E_{\text{test}} (\%)$	Worst Case $E_{\text{test}} (\%)$	Correlation Coefficient
Sigmoid	x -dir	12.024	38.007	0.995
Sigmoid	y -dir	14.6986	34.644	0.995

The data in the table supports the general recommendation that ANNs generally have lower error when d_p appears more continuous. This assertion is from $|E_y|$'s E_{train} being smaller than $|E_x|$'s, where in this investigation N_{train} is about twelve hundred samples. This is much larger than N_{test} , which is about three hundred samples. Also, the worst case error is smaller in the y -direction.

3.1.4 Investigation on Network Type – Hidden Neuron Types

There was a question of which ANN hidden neuron type would be best suited for EMC modeling during preliminary investigation. Utilizing the one-aperture configuration in the far field, $|E_x|$ data is shown in Table 3.3. The ANN structure was 50 hidden neurons trained with a QNM algorithm with inputs being the source location, and the output was the near field $|E_x|$ (V/m).

Table 3.3: Preliminary investigation on different neuron types.

Row Number	Neuron Type	Validation		
		Average Error (%)	Max Error (%)	Correlation Coefficient
1	<i>Arctangent</i>	20	60	0.96
2	<i>Ckt Simulator</i>	15.8	42	97
3	<i>Constant</i>	1.29e4	2.19e4	NA
4	Decay Sinusoidal	NA	NA	NA
5	<i>Dip Gaussian</i>	185	1207	0.4
6	Hyperbolic Tangent	NA	NA	NA
7	Linear	NA	NA	NA
8	Mexican Hat	NA	NA	NA
9	Multilinear 6	NA	NA	NA
10	Multi Sigmoid	NA	NA	NA
11	Multi Symbolic	NA	NA	NA
12	Multi Symbolic Fixed 1	NA	NA	NA
13	Multiply 2	NA	NA	NA
14	Neural Model	NA	NA	NA
15	<i>Normalize</i>	15.8	43.1	0.97
16	Positive Constant	NA	NA	NA
17	Positive Linear	NA	NA	NA
18	<i>Quadratic</i>	11.29	35.5	0.98
19	Relay	NA	NA	NA
20	<i>Sigmoid</i>	108	506.7	0.511

Table 3.3 shows the effect that the neuron's activation function has on validation error, where NA means that the ANNs would not train properly or have large errors in the hundreds, such as in row three. It should be noted that the results are based on ANNs trained on near field measurements and thus at points seemed sporadic/discontinuous, which is one possible rationale for the large errors. Upon review of this data, several observations can be inferred. A variety of the neuron types available in *NeuroModeler* require advanced experience that is beyond the scope of this thesis, such as the Mexican hat neuron type, because these types require advanced training techniques. There were six different neuron types that are italicized which had further investigations done, and it was found that arctangent and sigmoid neurons are best suited for EM ANNs. Sigmoid neurons were used as the primary hidden neuron type for case studies due to their consistent lower error and ease of training.

3.1.5 Investigation on Single Training Algorithms

Another variation to the training process was changing the types of training algorithms, as shown in Table 3.4. For this table, the X_{offset} and Y_{offset} were given as inputs to a twenty-hidden-arctangent-neuron ANN to estimate far field values of $|E_x|$ for a one-aperture configuration.

Table 3.4: Preliminary investigation on training algorithms for a one aperture geometry.

Row Number	Training Algorithm	Average Error (%)	Validation Max Error (%)	Correlation Coefficient
1	Backpropagation	30.54	71.66	0.9877
2	Sparse Training MLP	6.516e7	1.340e4	0.9916
3	Conjugate Gradient	2522.3	6497.4	0.9922
4	Quasi Newton	4.69	38.75	0.9994
5	Huber Quasi Newton	5.58	35.56	0.9992
6	Autopilot	5.06	20.36	0.9994
7	Simplex	7.13	26.82	0.9988

Table 3.4 shows the results for various training algorithms. Row number two had an extremely large error; the ANN was not properly trained. It should be noted that advantages for some of the algorithms, *i.e.*, computational time, memory usage etc. Advantages for the algorithms in rows one and two are that the algorithms could train faster and reduce memory usage. This was not tested/verified in this section. Rows four through seven were within three percent of each other. This is significant in that different training algorithms trained the same network with a substantial variation to ANN error.

3.1.6 Investigation on Multiple Training Algorithms

This subsection examines how different combinations of training algorithms running sequentially can have an effect on the error of the ANN. The ANNs were trained with 1-, 4-, 6-, 9-, and 16-aperture geometries and tested with an eight-aperture geometry. The ANNs created had sixteen, twenty, thirty, forty, and fifty hidden sigmoid neurons, trained

with QN, QNM, CQN, and CQNM algorithms. The f_{ann} s that had the lowest average E_{test} are shown in Table 3.5.

Table 3.5: Preliminary data results for different combinations of training algorithms from eight-aperture simulations.

Row Number	Different Combinations of Training Algorithms.	Number of Hidden Neurons (Sigmoid)	Lowest Avg. Error ANNs		
			Average $E_{\text{test}} (\%)$	Max $E_{\text{test}} (\%)$	Correlation Coefficient
1	Quasi Newton	16	4.04	15.57	0.9991
2	Quasi Newton MLP	16	5.80	16.59	0.9983
3	Conjugate + Quasi Newton	20	4.23	11.10	0.9991
4	Conjugate + Quasi Newton MLP	16	4.88	15.88	0.9987

The size of the training and validation sets were approximately six hundred points and had three hundred points, respectively. The average E_{train} and E_{valid} was within approximately one percent. As Table 3.5 shows, the lowest average E_{test} for different algorithms was similar, within two percent, suggesting that there may be a pattern. This was further examined with variations to the number of hidden neuron types, and the results are graphed in Figure 3.9.

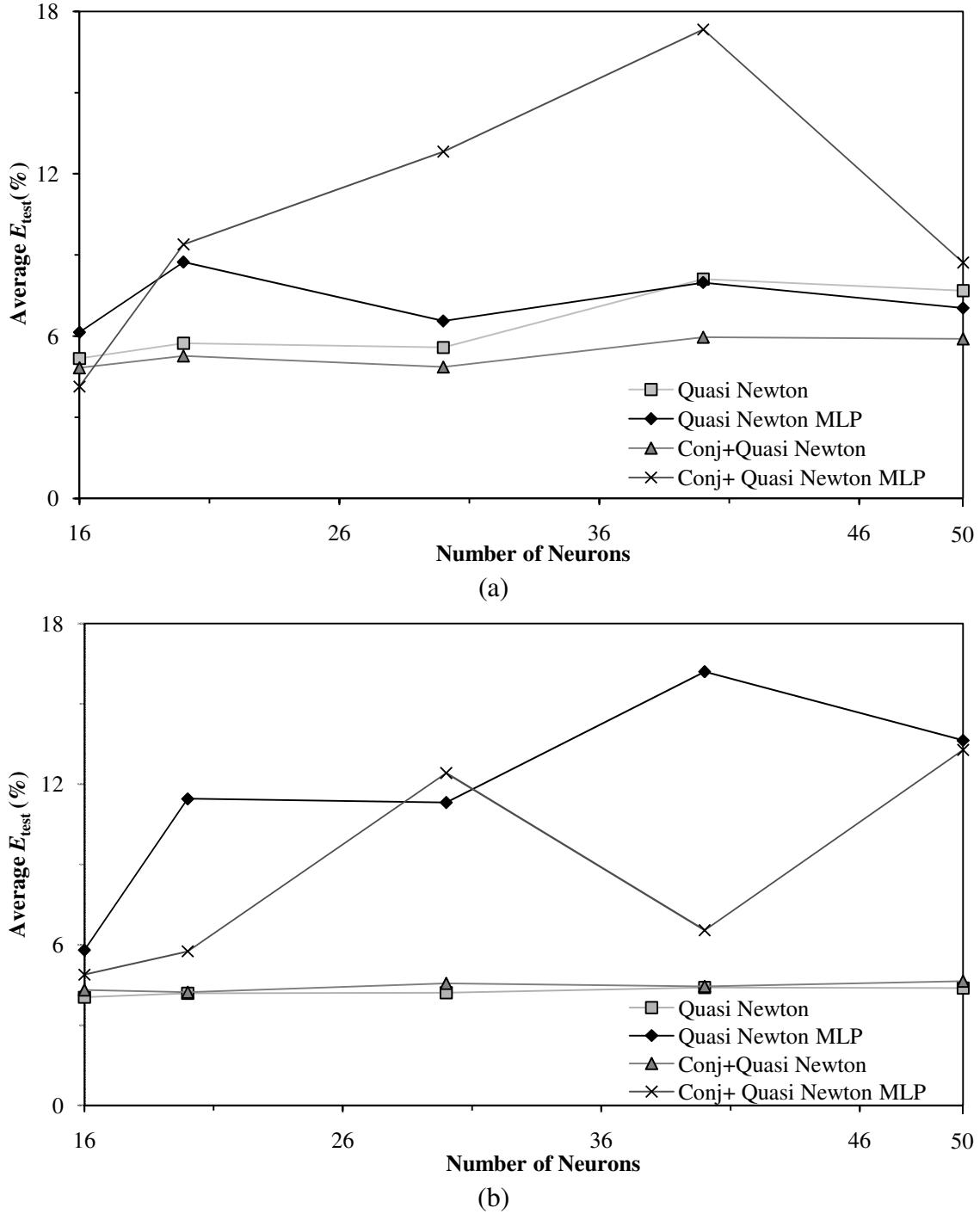


Figure 3.9: Average E_{test} results from an eight-aperture investigation: (a) arc tangent hidden neurons (b) sigmoid hidden neurons.

The significance of Figure 3.9 is that there is no discernable trend in the bands. For instance, single (QN, QNM) vs. multiple algorithm (CQN, CQNM) approaches have what appears to have at least two bands that almost overlap; however, it's a different combina-

tion for the different neuron types. There were twenty distinct f_{ann} s made and tested for Figure 3.9's creation. This figure shows how variations to network parameters affect the E_{test} values. ANN creation for case studies have variations to the number of hidden neurons and training algorithms. This is similar to this subsection, with the differences being f_{ann} iterations, the starting number of hidden neurons, the stopping number of hidden neurons, and the incremental increase of hidden neurons.

3.1.7 Size of Training Set on Error

One area of interest is to see how few training points are required to create an accurate EMC ANN. A training data set with d_p values for 1-, 4-, 6-, 9-, and 16-aperture geometries, with four inputs and one output, was created. The inputs were X_{offset} and Y_{offset} , $|E_0|$, and the number of apertures and the output was $|E_y|$ (dB V/m). The test set was from the two-aperture geometry and contained approximately three hundred data points. The preliminary experiment results of doubling N_{train} by combining the training and validation sets are shown in Figure 3.10.

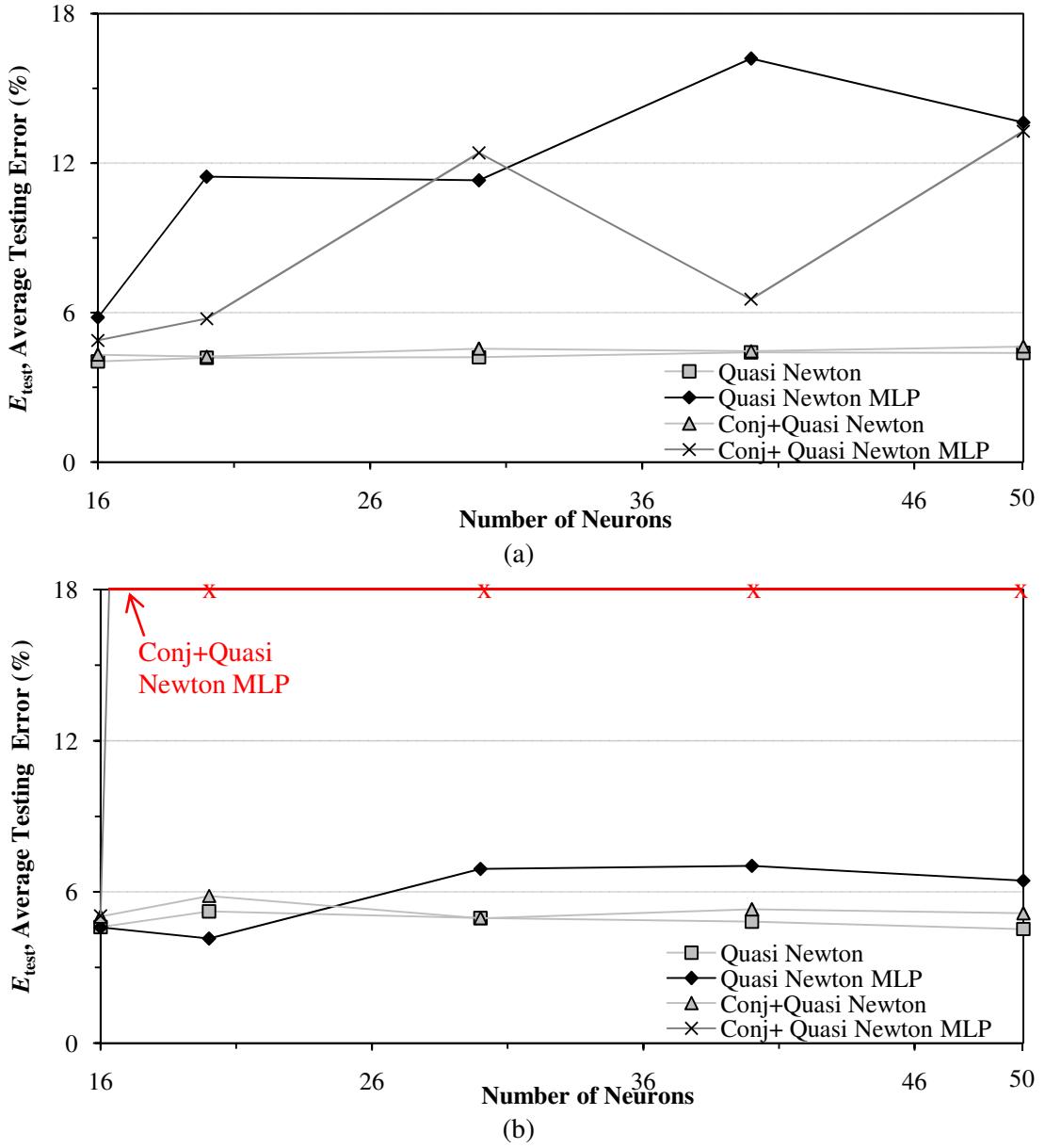


Figure 3.10: Preliminary investigation of the effect N_{train} has on the average E_{train} and E_{test} , determined that (a) N_{train} has seven hundred samples and (b) N_{train} has fourteen hundred samples.

In Figure 3.10, part (a)'s lowest average E_{test} was 4.15%, and for Figure 3.10, part (b)'s lowest average E_{test} was 4.04%. Even though E_{test} was smaller for fewer training points, the E_{train} was smaller for the larger N_{train} , which had an E_{train} of 3.36%, whereas the smaller N_{train} set had an E_{train} of 3.877% and an E_{valid} of 4.01%. To summarize, the experiment

tested the recommendation that more training points reduced ANN error, and the data supports this recommendation.

3.1.8 Cropping Input Data Space on Error

There is an interest to determine whether the regions of lesser significance for EMC structures have a significant effect on E_{test} . This was examined by cropping the data set for X_{offset} 's regions from 1-3 cm and 9-11 cm as well as cropping the data set for Y_{offset} 's regions 1-6 cm and 25-29 cm in the training, validation, and testing sets. The train data set consisted of 1, 2, 4, 9, 16 apertures, and the test set consisted of a six-aperture configuration. The ANN estimation for the six-aperture configuration using a cropped input data set is shown in Figure 3.11.

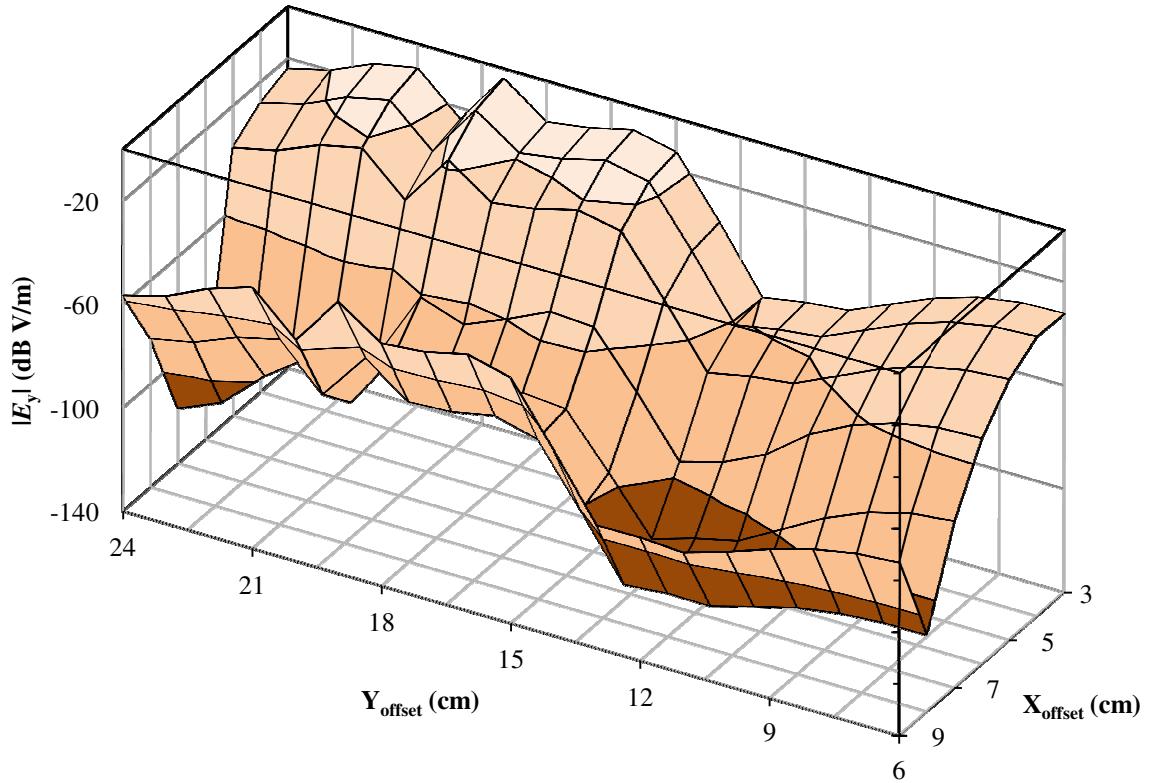


Figure 3.11: Preliminary cropped data set simulation, where the f_{ann} estimates the six-aperture cropped data set results.

Table 3.6 is the error result when Figure 3.11 is compared to the corresponding d_p values.

The f_{ann} had arctangent hidden neurons instead of sigmoid neurons.

Table 3.6: Comparison of ANNs created by comparing cropped training data sets to non-cropped training data sets.

Data sets	Measurement direction	Epochs	Average $E_{\text{train}} (\%)$	
Not cropped	x	1003	3.5	
Cropped	x	1420	4.5	
	Measurement Direction	Average $E_{\text{test}} (\%)$	Worst Case $E_{\text{test}} (\%)$	Correlation Coefficient
	x	25.44	79.9	0.98
Cropped	x	132.34	222.4	0.89

Table 3.6 shows a comparison of the cropped data set with the non-cropped data set; the average E_{test} significantly increased by about five hundred percent. The increased error is important, and the fringe regions of the input data should be included.

3.1.9 Investigation on Changes to Inputs (specifically $|E_0|$)

One of the most experimental aspects of EMC ANN creation is determining the proper ANN inputs. In the preliminary experiments, several variations of inputs were experimented upon. One of the most significant inputs that reduced the error of the ANN was the unshielded (removing the *top face*) radiation for a given source location. This quantity is referred to as $|E_0|$. A training set was created of 1-, 4-, 6-, 9-, and 16-aperture geometries and was tested with two and eight apertures. The results can be seen in Table 3.7.

Table 3.7: Comparison of E_{test} results of ANNs trained with and without $|E_0|$.

Configuration		Average $E_{\text{test}} (\%)$	Max $E_{\text{test}} (\%)$
Two Aperture E_{test}	With $ E_0 $	13.6	38.1
	Without $ E_0 $	15.2	39.9
Eight Aperture E_{test}	With $ E_0 $	6.7	6.1
	Without $ E_0 $	16.5	17.4

Table 3.7 shows a remarkable reduction in error for the eight-aperture preliminary experiment by approximately ten percent. This preliminary experiment stressed the importance of proper ANN input and output selection. Another preliminary experiment examined the effects of including $|E_0|$ as an input and adjusting the size of the training data sets and testing data sets (from Table 3.7) so that similar data sets used in Table 3.8. However, in Table 3.8, the training set was split so that a validation set exists.

Table 3.8: Comparison between ANNs trained with differing numbers of inputs.

Inputs		Average $E_{\text{test}} (\%)$	Max $E_{\text{test}} (\%)$	Correlation Coefficient
X_{offset} Y_{offset} Number of Apertures	Train Set	2.41	16.84	0.9997
	Validation Set	2.52	21.92	0.9996
	Test Set	12.40	28.56	0.9922
X_{offset} Y_{offset} Norm. Area $ E_0 $	Train Set	3.36	15.52	0.9994
	Validation Set	3.53	21.86	0.9996
	Test Set	30.59	65.78	0.9967

Table 3.8 illustrates the E_{test} for some ANN structures. This table shows that the addition of more inputs does not guarantee E_{test} reduction.

3.2 Case Study One: Source Position Varied within a Shielding Enclosure

This case study examines how a source's position within a cavity and the number of apertures affects the radiated emissions from an enclosure. This is accomplished by sequentially adding additional apertures to the front face of a rectangular cavity. The variations to the source location mimic variations of a noisy component placement within the shielding enclosure.

3.2.1 Procedure of Case Study One

For each geometry variation in *HFSS*, the FEM mesh was rebuilt. The meshes were refined at least five times and have two consecutive passes under the maximum delta energy. The meshed region is the volume enclosed by the radiation boundary, and the radiation boundary was always at least $\lambda^{\text{solution}}/4$ from the nearest edge of the simulation, where $\lambda^{\text{solution}}$ is the *HFSS* solution wavelength. Typically, these solution settings for the *HFSS* mesh are set in order to provide acceptable accuracy for tolerable computational load. Typically, during an adaptive pass, regions of large changes in derivative changes are probable regions of higher error density. Subsequent adaptive passes, therefore, have a larger number of increases to tetrahedral density. The solution frequency for the meshes was set for 1.55 GHz. The *HFSS* simulation settings were set such that simulations would have at least five adaptive meshes evaluated and the maximum delta energy was below 0.08. After a preliminary solution formation and data verification, the parametric simulations were completed.

Figure 3.12 illustrates a rectangular cavity with seven circular apertures on the top face. The cavity dimensions are $30 \times 12 \times 3 \text{ cm}^3$, and the circular apertures are 1 cm in diameter. The cavity encloses a Hertzian dipole (oriented along the y -axis) in the $z = 1.5$ cm plane. The configuration is chosen to represent an enclosed source (*e.g.*, an FPGA) within a PCB shield. The apertures represent the ventilation/access holes on the shields. For the PCB-shield structure illustrated in Figure 3.12, two EMC design scenarios may arise. First, suppose that the previous design configurations of the structure involved all but aperture number four on the cavity. For some reasons, such as thermal constraints, it may be required to add an additional aperture (aperture number four) on the cavity.

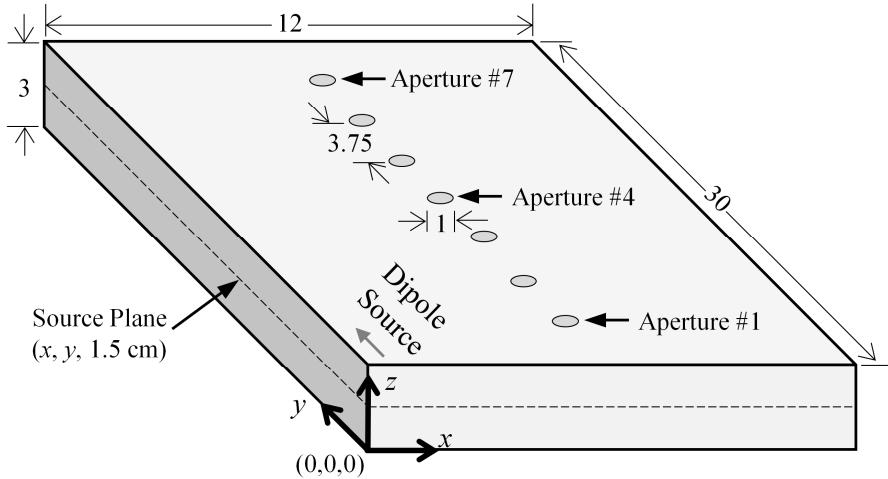


Figure 3.12: Overview of a PCB EMI source location varied within a shield; all dimensions are in centimeters.

Adding an aperture affects the EMC performance of a structure since it is well known that apertures are a good source of EMI coupling inside/outside cavities [55]. In this context, an EMC design choice is whether to carry out the entire system-level analysis with the new configuration or rely on data available from a previous configuration. Similarly, another EMC design decision pertains to re-orienting, or shifting the enclosed EMI source inside the shield. This is a very important decision since the EMI emissions from the cavities are strongly dependent on the source location. As discussed above, such scenarios are difficult to model using available tools due to large time requirements. In this context, the ANN models become ideal candidates for such EMI/EMC design scenarios. The proposed ANN model is illustrated in Figure 3.13. As shown, the model predicts the radiated emissions for a given set of inputs (number of apertures on the cavity, source location inside the cavity, and the radiated electric field $|E_0|$ with top face of the cavity open).

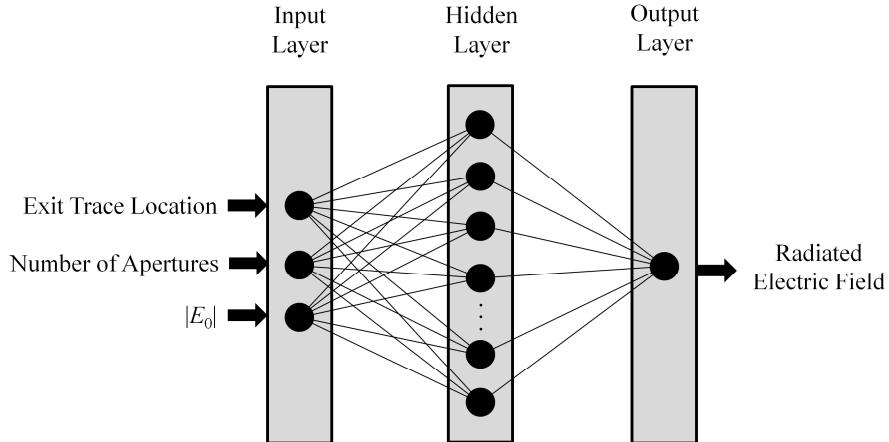


Figure 3.13: ANN model estimation for $|E_y|$ for the structure in Figure 3.12.

A simpler model could have been chosen requiring only two inputs, *i.e.*, the number of apertures on the cavity and the source location; however it was found that the accuracy of the resulting model was relatively very poor. Additional information provided by $|E_0|$ improved the accuracy considerably.

The training data for the proposed model was obtained using *HFSS* for various source locations (in the $z = 1.5$ cm plane) inside the cavity. Specifically, the magnitude of the radiated electric field was calculated 1 m above the top face of the cavity. The cavity walls are Perfect Electric Conductors (PECs) and have zero thickness. Moreover, for each source location, parametric simulations were carried out starting with only aperture number one on the top face and then repeating the simulations after sequentially adding other apertures, except aperture number four.

Figure 3.14 summarizes the ANN creation process involved in creating ANNs with sigmoid-type hidden neurons. The number of hidden neurons was varied with a set increment. For the ANNs created for this case study, the number of hidden neurons was varied from 10 to 28 in increments of 2 hidden neurons.

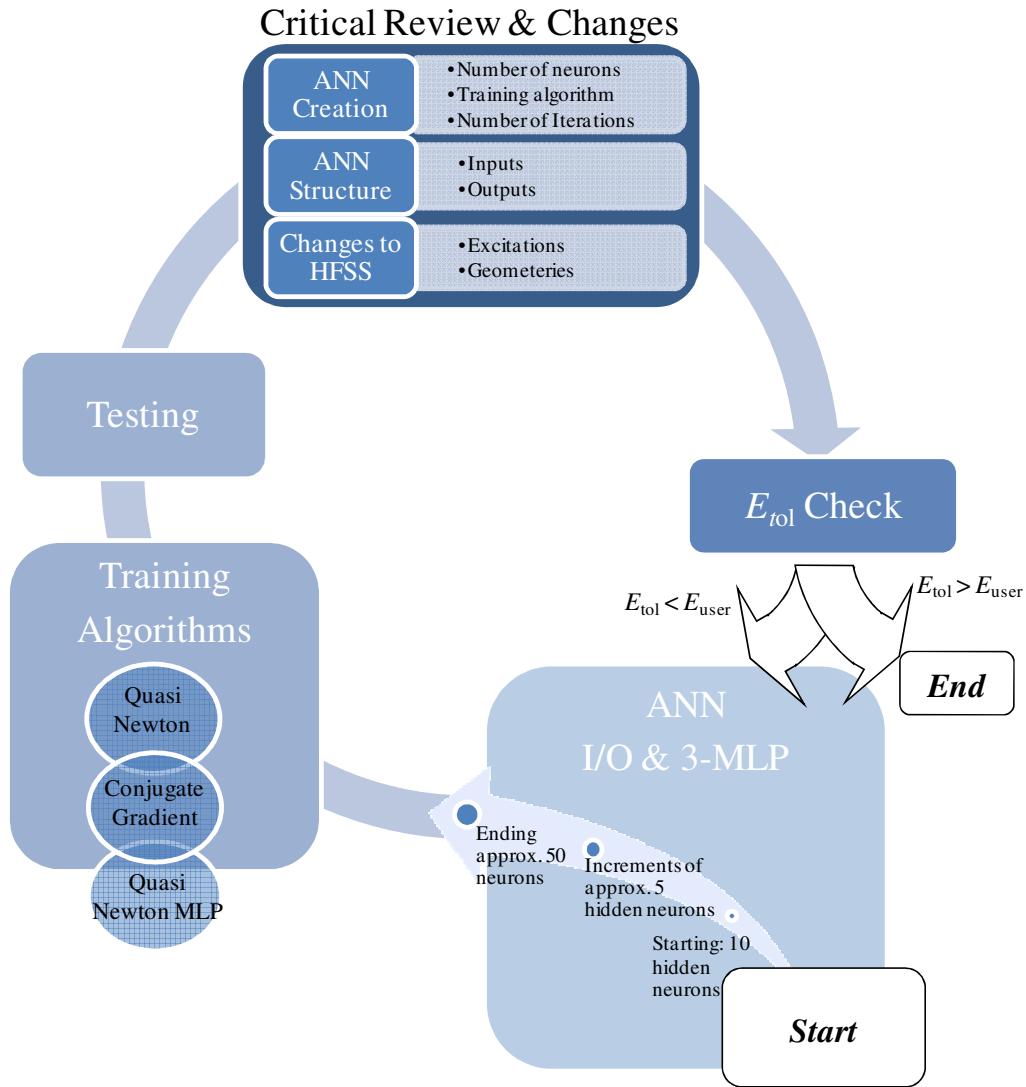


Figure 3.14: ANN creation model that shows the importance of parameter selection.

The training algorithms utilized were QN, QNM, CQN, and CQNM. The ANNs had inputs of the source location, number of apertures, and $|E_0|$, and an output of $|E_y|$ (dB V/m). Each increment had 10 iterations of f_{ann} s created, meaning that for the exact same input files and network structures, the initial weights and biases were reset and the ANNs were re-trained. The ANNs were trained utilizing QN, QNM, CQN, and CQNM, resulting in over three hundred and fifty distinct ANNs being created.

3.2.2 Results of Case Study One

Once the ANN model was trained, it was tested to measure the radiated fields for the additional aperture number four on the top face for various source locations inside the cavity. The results are plotted in Figure 3.15 (a) and (b) and show a remarkable agreement with *HFSS* simulations.

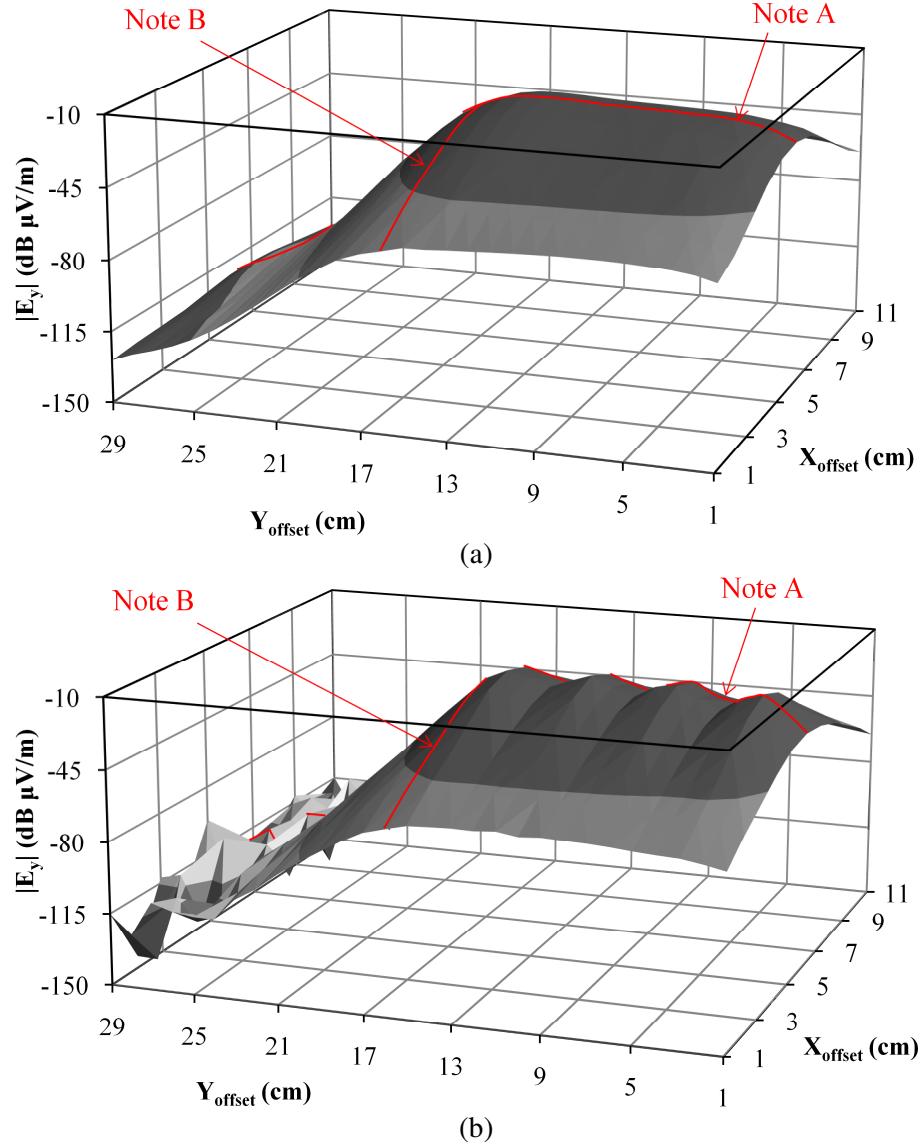


Figure 3.15: Radiated emissions for adding the fourth aperture on the PCB shield: (a) ANN estimation, (b) full-wave calculations.

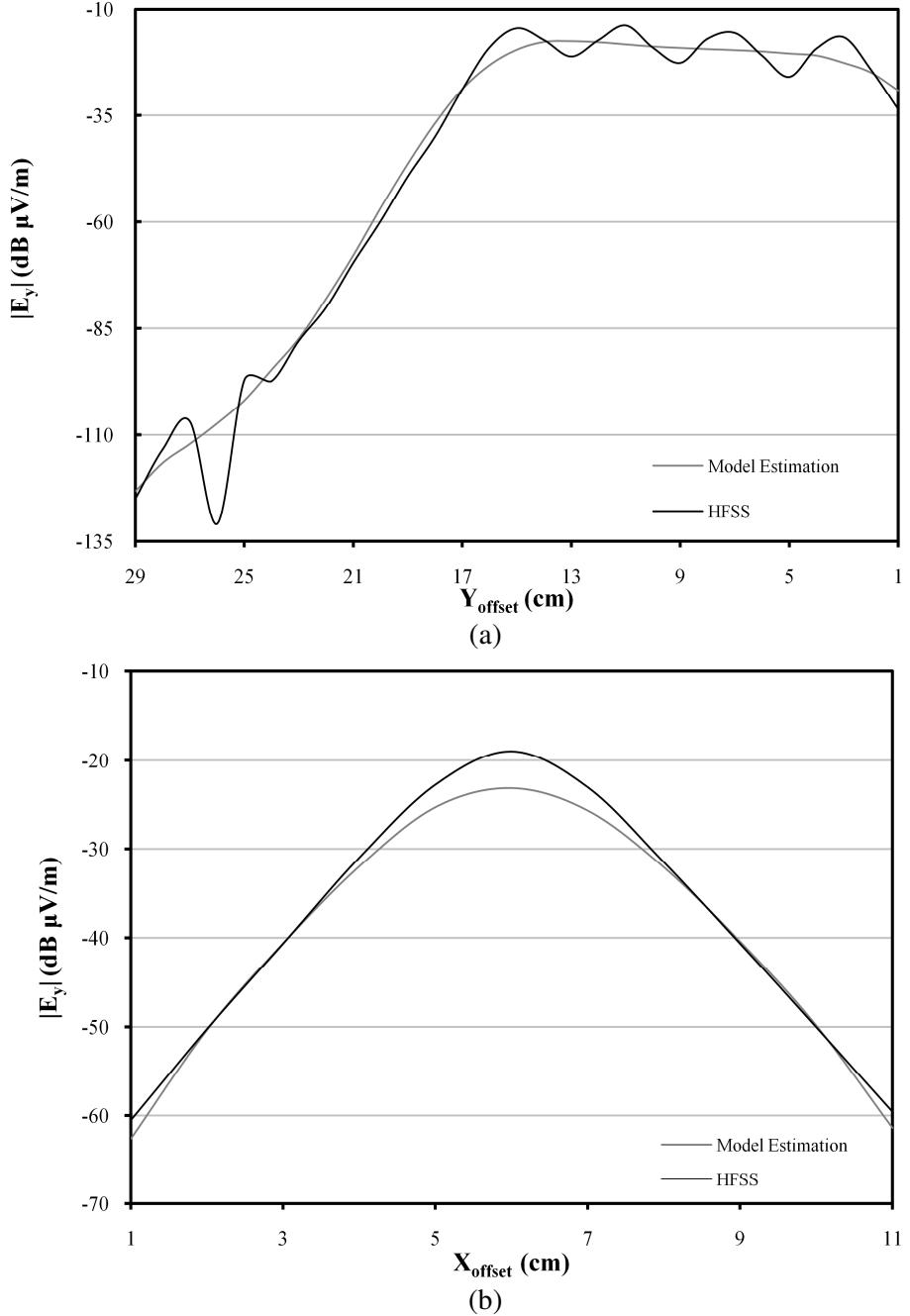


Figure 3.16: Radiated emissions for a four aperture PCB shield: (a) Y_{offset} slice from Figure 3.15 the ANN $|E_y|$ estimation, (b) X_{offset} slice from Figure 3.15, the full-wave calculations.

Specifically, Figure 3.15 (a) plots ANN estimates when the source is swept inside the cavity in the $(x, y, 1.5 \text{ cm})$ plane. Figure 3.16 provides a closer look at the estimated and calculated values. In Figure 3.16 (a), the source is moved along the y -axis in the line $(x,$

16, 1.5); in Figure 3.16 (b), the source is moved along the x -axis in the line (6, y , 1.5). In both figures, a close agreement is depicted. The large discrepancy in the calculated and estimated values in Figure 3.16 (b) from $y = 25$ to 29 cm is probably due to numerical errors, as the magnitude of the electric field is very small in this region.

The training of the ANN model was carried out using an automated ANN modeler, *NeuroModeler* [51] and utilized the recommendations given in Chapter 2. This resulted in the creation of two hundred and thirty distinct f_{ann} s. Of the ANNs created, a ten-sigmoid-hidden-neuron ANN trained with QNM was found to have the lowest average E_{test} values of all the f_{ann} s created for this case study. The f_{ann} s with the lowest E_{test} had an E_{train} value of 1.619% and an E_{test} value of 1.544%.

Since the ANNs presented are trained utilizing randomized initial weights and biases, this makes ANN creation a stochastic process. The f_{ann} s created utilizing ten hidden neurons with a QNM training is shown in Figure 3.17, which compares f_{ann} iteration vs. E_{train} and E_{test} . Given the nature of random initializations for weights and biases within the ANN networks, there were noticeable variations between the iterative ANNs. The s.e. (standard error) is consistently larger for the average E_{test} values, with an SE magnitude of approximately 0.05 when compared to the average E_{train} value's SE, with a magnitude of approximately 0.01. The range of E_{test} as the iteration is varied has an average E_{test} from 1.54% to about 1.85%, and the average E_{train} had an approximate range of 1.58% to 1.63%.

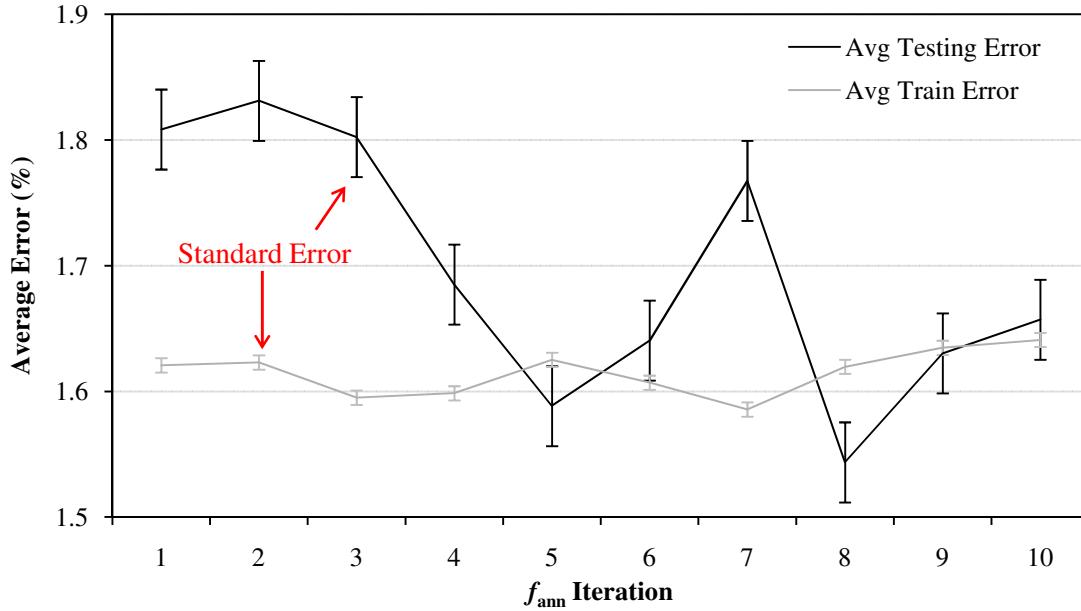


Figure 3.17: Comparison of the average E_{test} of f_{ann} as a function of perturbed weights-biases.

To surmise, this case study examined how an f_{ann} can estimate radiated emissions of a dipole's source as its position is varied in a cavity as well as variations to the number of apertures. The significance of this case study is that in a PCB design, a source's position inside a cavity can be varied as well as the number of apertures on the shield, and that information is passed to f_{ann} , which can accurately and quickly estimate the radiated emissions of the source.

3.3 Preliminary Investigation for Case Study Two

As mentioned earlier, with shielding technology there is a tradeoff between thermal concerns and radiated emissions. This was initially examined in Section 3.1; however, in this section, large variations to the number of apertures will be examined where the aperture geometries are arbitrarily chosen. These arbitrary changes to the number of apertures were made such that a 30 x 30 x 3 cm PEC shielding enclosure with 0.5 cm diameter apertures were equally spaced such that there were 19 rows/columns of apertures on the

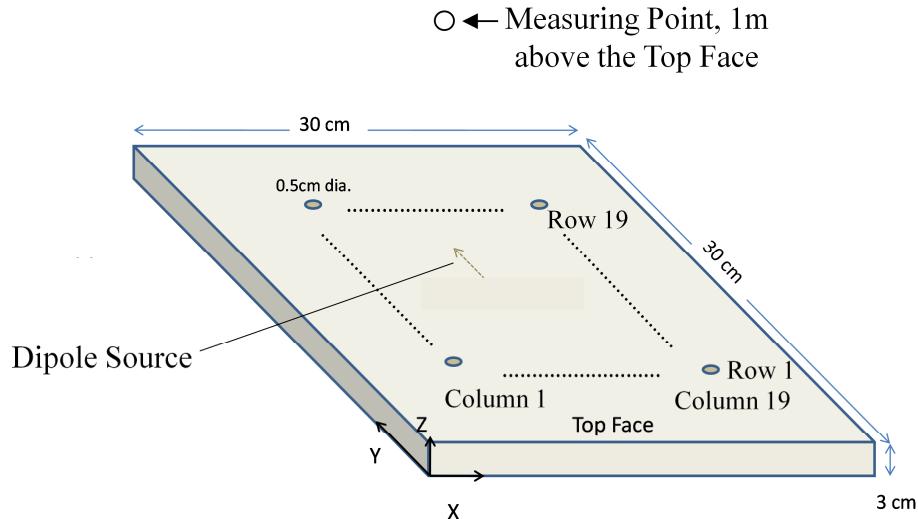


Figure 3.18: Preliminary simulation for Case Study Two.

top face of the shielded enclosure. The apertures were 1.5 cm from the edges and off center of the apertures. Inside the enclosure there was a 1.55 GHz dipole source located on the xy -plane where $z = 1.5$ cm. This can be seen in Figure 3.18. Then arbitrary rows/columns of apertures were removed and then simulated in *HFSS*, and $|E_y|$ dB V/m was recorded. For instance, initially the project was simulated with all holes, then row one of apertures was removed and simulated, then row nineteen of apertures was removed and simulated, then column one of holes was removed and simulated, then column nineteen was removed and simulated, and so forth. Another variation done was to remove rows one and nineteen the same way; however, columns two and eighteen were removed instead of one and nineteen. Another variation done was to remove rows and columns starting at two and eighteen. This resulted in a training set consisting of 1, 2, 4, 6, 8, 9, 12, 16, 20, 25, 36, 50, 81, 100, 171, 190, and 361 ($N_{\text{train}} = 17$) apertures and a testing set consisting of 15, 45 ($N_{\text{test}} = 2$) aperture geometry. The f_{ann} results can be seen in Figure 3.19.

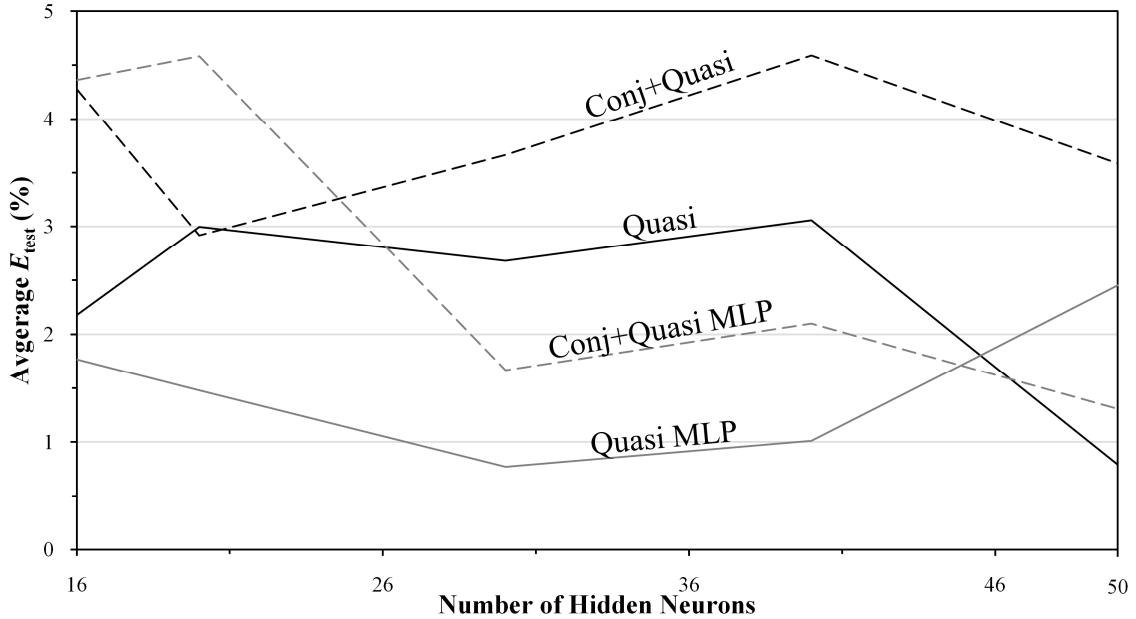


Figure 3.19: Preliminary results on arbitrary aperture addition.

As the results in Figure 3.19 show, the thirty sigmoid hidden neurons network trained by Quasi MLP had the lowest average E_{test} error for the arbitrary hole addition experiment, where the average E_{test} was 0.7718% and the average E_{train} was 7.84%.

3.3.1 Investigation on the Significance of an Outlier

Reviewing the line “HFSS with error” in Figure 3.19, there was an assumption that at the six-aperture sample, there is a d_p calculation error. The lowest E_{test} from Figure 3.19 is the line with the label “With 6hole ANN,” and this ANN was trained with the outlying d_p value. A preliminary experiment was conducted to train an ANN without that erroneous d_p (6-aperture) value, and the results were compared. Also shown in Figure 3.20 is the correctly calculated d_p value, labeled as “HFSS without Error.” This point was not used for training any ANNs, and it is included to show what the correct d_p value is. The ANN structure in the figure has thirty hidden sigmoid neurons with three inputs and one output.

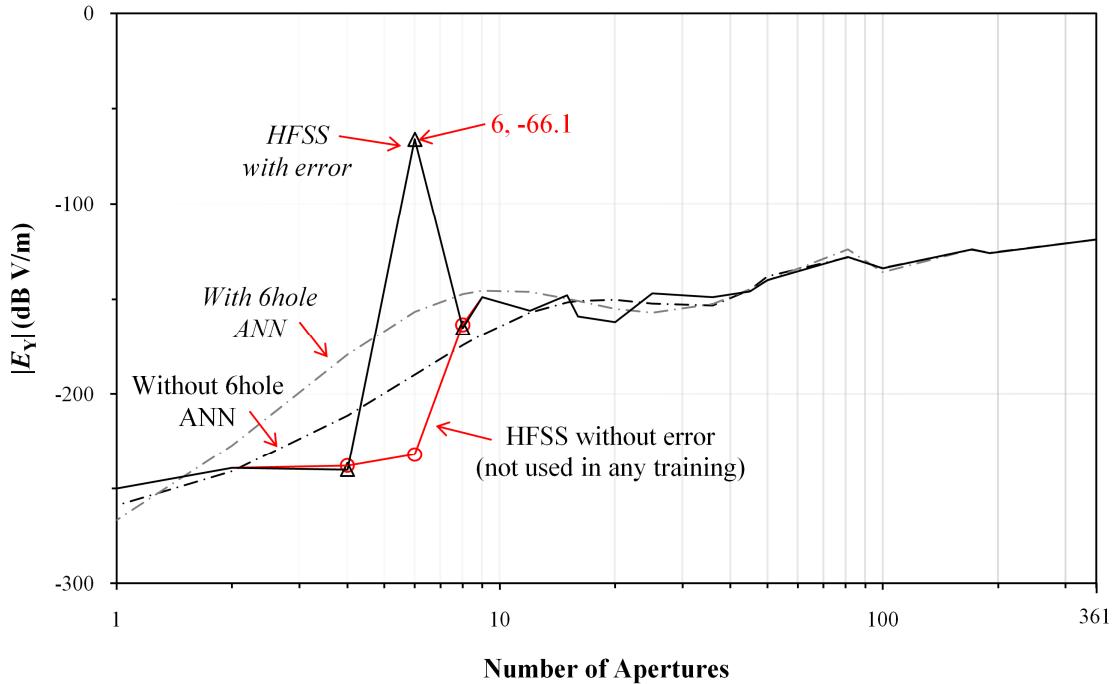


Figure 3.20: Preliminary experiment that highlights the importance of training points in modeling error from the ANNs with the lowest average E_{test} in Figure 3-19.

As the results show, training without the sixth aperture had an average E_{train} : 4.849%, and an average E_{test} : 1.741%. Training with the sixth aperture had an average E_{train} : 7.846% and an average E_{test} : 0.772%. The ANN inputs were total number of apertures, normalized area of apertures, and unshielded radiated value. The ANN outputs were $|E_y|$.

The discontinuities found in *HFSS* simulations could be attributed to a variety of reasons, such as insufficient meshing for the geometry, resonance within the cavity etc. The *HFSS* error in Figure 3.20 for six apertures was caused by a systematic error, particularly a failure in specifying the correct density of tetrahedrals within the *HFSS* mesh. The correctly simulated values are labeled “*HFSS without error*.” In Figure 3.20, the *HFSS* error was used in both training sets to show the importance of the outlier and how closely ANNs can model a rapid change. As shown in Figure 3.20, there was a large $d(\mathbf{x}_p)^2/d^2(\mathbf{x}_p)$ value, where \mathbf{x}_p is the six-aperture geometry. The f_{ann} error was larger than it could have

been, so the sixth aperture was removed from the training set, and the ANN was re-trained. Without the sixth aperture, the training error was reduced; however, the testing error was increased. Given that the N_{train} is much greater than N_{test} , removal of the outlier is generally favored in reducing the overall error of the f_{ann} . User expertise is critical in determining if an outlier is significant or not.

3.3.2 Experimental Evidence on Extrapolation vs. Interpolation Training

The inclusion of samples where $p = 1$ and N_{train} was stressed in Chapter 2. This section provides experimental evidence examining thirty-two discrete sets of training data, sixteen of which have training samples where $p = 1$ and N_{train} (*interpolating*), and the other sixteen that do *not* have $p = 1$ and N_{train} (*extrapolating*). The results are shown in Figure 3.21. The f_{ann} s created had thirty sigmoid hidden neurons trained with QNM. Examined in the figure is evidence for the recommendation that as N_{train} goes to zero, E_{test} goes to a large value. As Figure 3.21 illustrates, after a certain number of training points were moved into the testing set, E_{test} becomes intolerable. This happens approximately after the removal of eight training samples; however, the E_{train} becomes smaller. This reduced E_{train} is not an accurate indicator of f_{ann} estimation given that N_{train} is significantly smaller than N_{test} . For this preliminary investigation, any simulations below eight training samples resulted in unacceptable testing error. As shown, the training error improves, in general, when more data points are increased. Figure 3.21 (a) shows training sets based on keeping the extremes of the input space included for training, whereas part (b) does not.

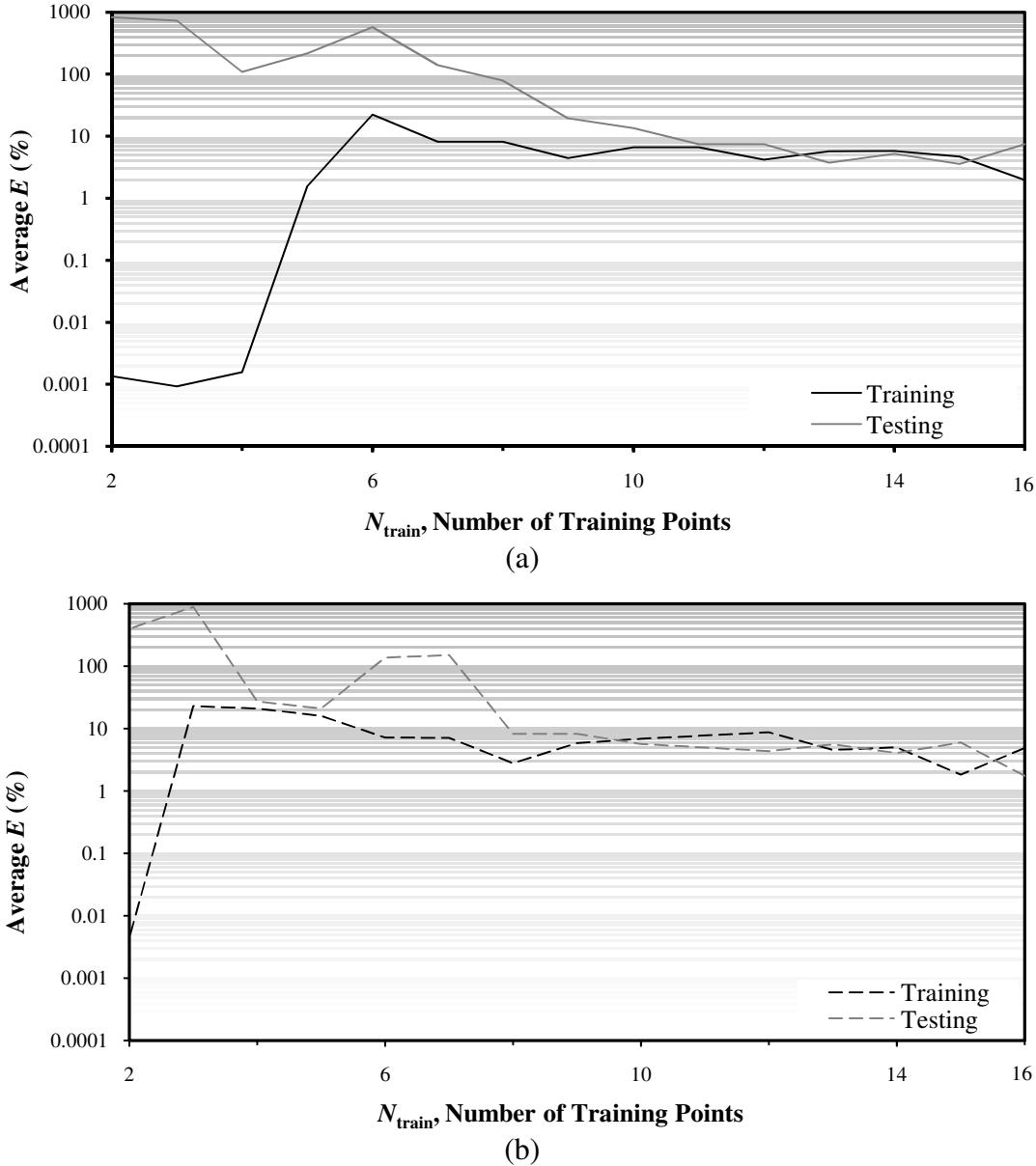


Figure 3.21: Investigation results showcasing how moving training points into testing points affects average E_{train} and E_{test} : (a) an interpolating training data set, (b) an extrapolating training data set.

Given the large number of combinations that can be made with even a small data set size in the order of twenty-five samples, this recommendation is a very lenient guideline. Examining $N_{\text{train}} = 11$ to 16 in more depth is shown in Table 3.9.

Table 3.9: Comparison of a preliminary investigation on variations to the number of apertures as a function of N_{train} .

N_{train}	Interpolating data sets		Extrapolating Data sets	
	E_{train} (%)	E_{test} (%)	E_{train} (%)	E_{test} (%)
16	1.97	7.45	4.85	1.74
15	4.70	3.59	1.82	6.00
14	5.77	5.20	4.97	4.06
13	5.70	3.72	4.53	5.56
12	4.18	7.40	8.65	4.34
11	6.61	7.40	8.44	5.27

For example, looking at the average E_{test} in Table 3.9, where $N_{\text{train}} = 16$, the extrapolation average E_{train} was 1.741%, and the interpolation average E_{train} was 7.449%. There is a difference of 5.709%, showing that the guideline can be wrong. For Figure 3.21, the E_{test} for interpolation is smaller than extrapolation only four times out of the fourteen training sets.

In summary of the preliminary investigations, several things were observed: 1) Data set creation recommendations are not definite, but example dependent; 2) data outlier points that have a very large second-order derivative have a substantial effect on the overall error for the ANN; 3) although ANNs can estimate radiated emissions reasonably when holes were arbitrarily removed, as demonstrated earlier from Case Study One (variable source location inside a shield), a more structured format leads to the lowest average error for Case Study Two. This is shown in the next section.

3.4 Case Study Two: Large Variations to the Number of Apertures on a Shield

For design scenarios more complicated than the previous case study, the ANN modeling approach offers significant computational advantages. For example, a range of apertures

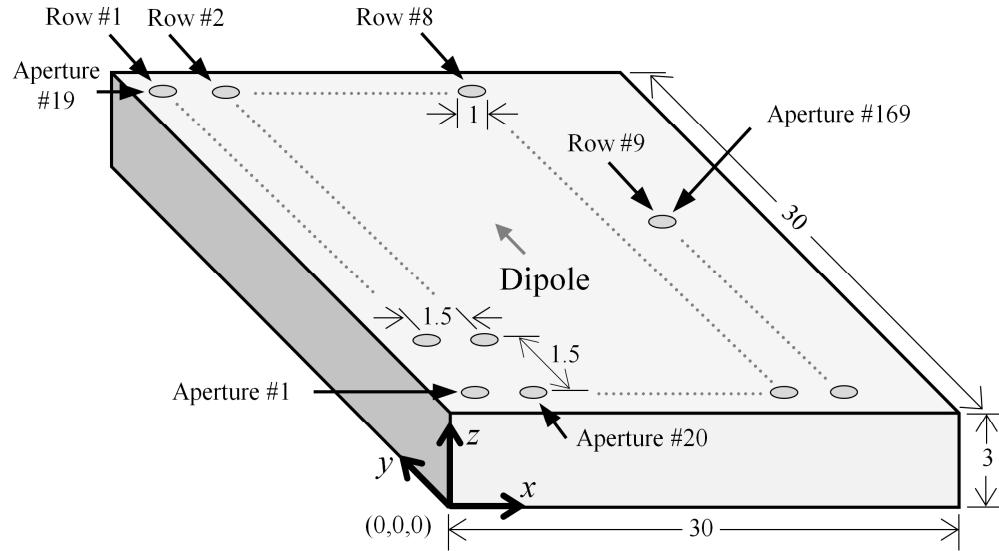


Figure 3.22: Overview of large variations to sequential variations to the number of aperture enclosing a dipole source at its center.

on a shield can be efficiently evaluated for radiated emissions using ANNs. This range of apertures can be demonstrated by considering the structure illustrated in Figure 3.22. As shown, an EMI source is enclosed at the center of a rectangular cavity. The cavity dimensions are selected as $30 \times 30 \times 3$ cm and the circular apertures are chosen as 1 cm in diameter.

3.4.1 Procedure of Case Study Two

The proposed model is represented in Figure 3.23. As shown, the model has three inputs: 1) number of apertures, 2) normalized area of apertures, and 3) the radiated electric field $|E_0|$ with the top face of the cavity removed. The desired output: $|E_y|$ is measured 1m normal to the top face. The normalized area of apertures means the total area of the apertures divided by the total surface area of the cavity's top face. For collecting the training data for the proposed model, the following algorithm was generated:

- Radiated fields were calculated 1 m above the cavity with only aperture number one located at 1.5, 1.5, and 3 cm on the top face.

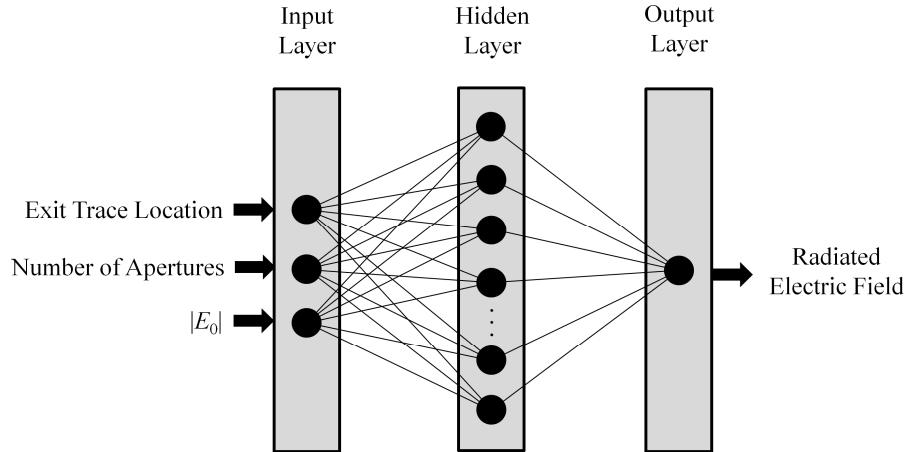


Figure 3.23: ANN model for estimating the radiated emissions for a given number of apertures on the shield.

- Next, aperture number two was added 1.5 cm off from aperture number one along the y -axis, and simulations were repeated.
- This procedure was repeated until nineteen apertures were added in row number one on the top face. Then another row of apertures, row number two, displaced 1.5 cm along the x -axis, was started.

This sequential addition of holes was carried out up to one hundred and sixty-nine apertures. Out of these one hundred and sixty-nine apertures data points, one hundred and fifty-two samples were employed for training the ANN models, and seventeen samples were used for testing.

3.4.2 Results of Case Study Two

A comparison of the results for the ANN creation in this case study is shown in Figure 3.24 (a), where the f_{ann} values shown are for the iteration that had the lowest average E_{test} values. This figure also compares the created ANNs' average E_{test} for this case study to its preliminary investigation.

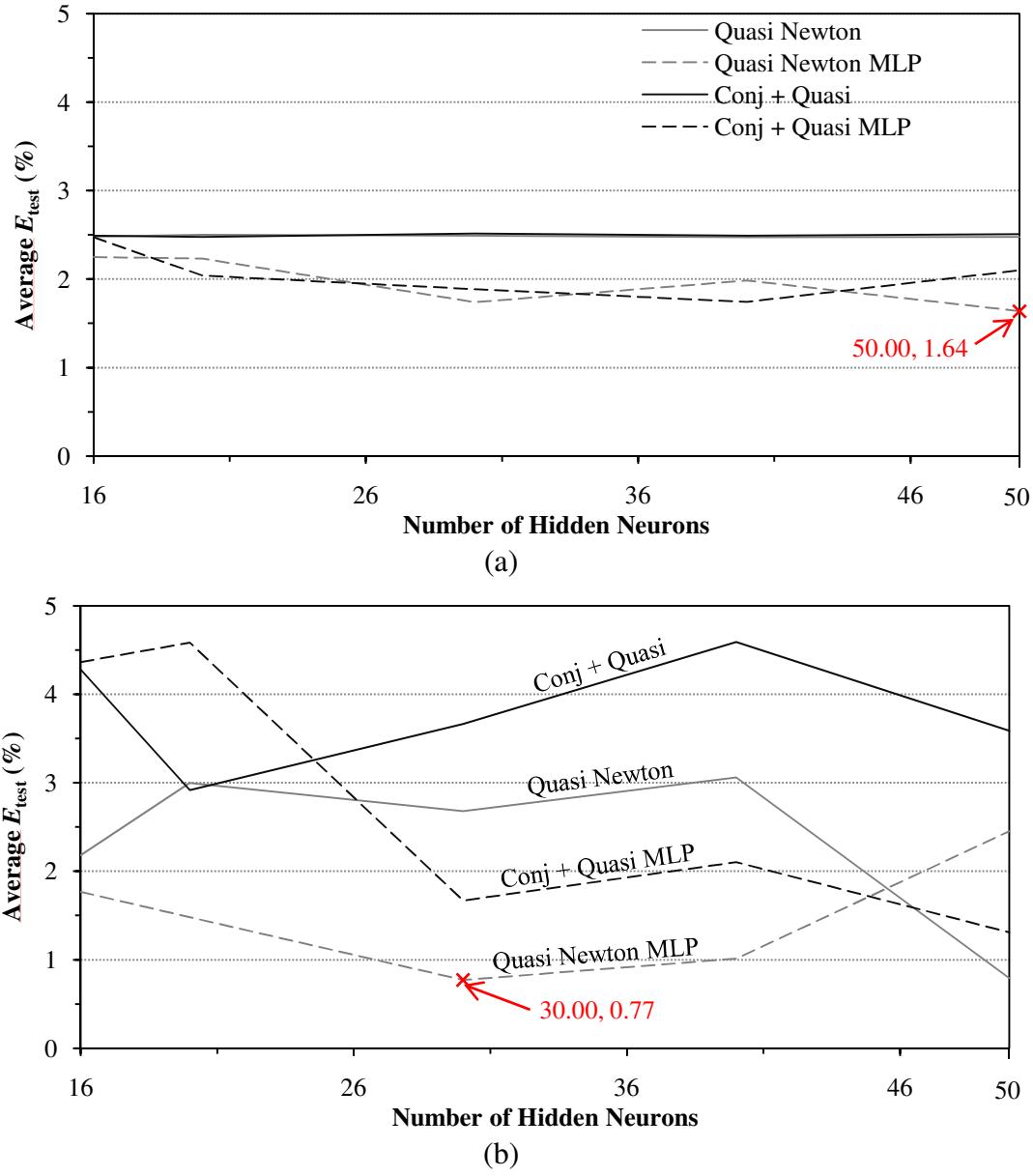


Figure 3.24: Comparison between Case Study Two's (a) sequential aperture addition average E_{test} results and (b) arbitrary aperture addition average E_{test} results.

As shown in Figure 3.24, the ANN training algorithms have an effect on E_{test} . The variations for all the algorithms in part (a) and (b) of Figure 3.24 stayed within approximately one percent as the number of hidden neurons varied. The lowest average E_{test} is lower in (b) because N_{test} in part (b) is much smaller than in part (a) ((a) $N_{\text{test}} = 17$ (b) $N_{\text{test}} = 2$). For example, the arbitrary geometries and test samples were randomly selected, and the over-

all error E_{test} is lower than that obtained using a sequential collection of data. Examining the many available ANN combinations is an important step in reducing the overall process of testing ANN's error. This is significant in the sense that the graph illustrates the effect training algorithms have on average testing error as well as the effect that the number of hidden neurons has. From the data, Quasi Newton MLP algorithm and Conjugate gradient combined with a Quasi Newton MLP algorithm consistently had the lowest average testing error for a given training and testing set.

Based on Figure 3.24, it can be concluded that the overall accuracy of the model depends not only on the number of training points but also on the particular set of data points used for training. Figure 3.25 shows the 50 hidden sigmoid neuron trained by QMN. The note on the figure, for the 131-aperture configuration, converged too early. When HFSS converged to tolerances, the correct value is -40.08 dB μ V/m. Figure 3.25 shows a remarkable accuracy between the full-wave solvers and the ANN estimates.

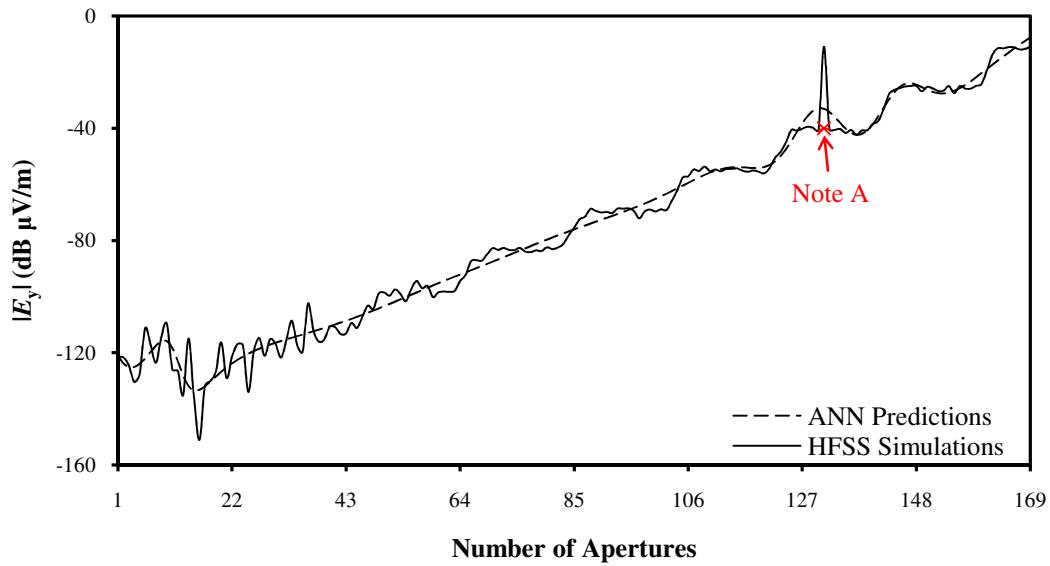


Figure 3.25: Estimated radiated emissions versus number of apertures.

Given the nature of ANNs, there is likely to be an ANN which will have a lower training E_{test} ; however, if the created/existing ANN already satisfies the error tolerance, it

is sufficient. This section does show that developing neural models needs constant improvement. In a product line, as more and more data become available, the trained models can be re-trained for added accuracy.

3.5 Preliminary Investigation for Reverse Modeling

The procedure for reverse modeling is outlined in Section 2.6. The preliminary $f_{\text{ann},c}$ was tested on a variety of p values for the testing set based on the experimental investigation in Chapter 4 (see Section 4.1.1). For the specifics of the f_{ann} creation, or for the geometry, see Section 4.4. What is significant from the preliminary training is that the f_{anns} and the data sets were utilized for the initial investigation on the reverse modeling algorithm. The $f_{\text{ann},ic}$ was trained utilizing d_p values from the training set. The $f_{\text{ann},jc}$ is chosen by the $f_{\text{ann},ic}$ with the lowest E_{train} . The preliminary results of running the algorithm utilizing the $f_{\text{ann},jc}$ is shown in Table 3.10.

Table 3.10: Reverse modeling preliminary investigation.

Inputs(\mathbf{x})				$F_{\text{ann}}(\mathbf{x})$ (dB V/m)	$F_{\text{ann}3c}(\mathbf{x})$ (dB V/m)	d_p (dB V/m)
Frequency (GHz)	X_{offset} (cm)	$ E_0 $ (dB V/m)	# of Holes			
1.079	5.4	46.398	9	31.427	32.91	33.02
1.079	4	46.406	9	31.423	32.90	32.79
1.132	9.9	48.994	9	31.427	33.30	33.15

The algorithm utilized in Table 3.10 had consistent E_{tol} and Δy values for all the rows. As shown for these specific points, there was a significant reduction in the error for those specific sample points. There was a four- to five-percent reduction in the error of the f_{ann} by using $F_{\text{ann},c}$, where $F_{\text{ann},c}$ is the value returned from the reverse modeling algorithm.

3.5.1 Comparison of Error Equations

Also completed during the modeling of the preliminary investigation was a comparison of RMSE and *NeuroModeler*'s error equation. The results of the comparison are

shown in Table 3.11.

Table 3.11: Comparison of *NeuroModeler* and RMSE equations.

		Train Average (%)	Test Average (%)
Chapter 3's	<i>NeuroModeler</i>	1.819	8.399
Case Study 1	RMSE	2.404	10.131
Chapter 3's	<i>NeuroModeler</i>	2.708	2.515
Case Study 2	RMSE	2.216	1.840
Chapter 4's	<i>NeuroModeler</i>	5.399	5.599
Case Study	RMSE	5.349	5.479

The error equations are defined in Chapter 2; RMSE and *NeuroModeler*'s training and testing error are similar in magnitude, within approximately one percent of each other. Numerically, the difference between the error equations comes from the denominator of RMSE utilizing the d_p value for p , whereas *NeuroModeler* utilizes the max/min d_p values from the training set.

3.6 Reverse Modeling of Case Study Two

To improve the accuracy of the developed model while using less training points, the reverse modeling algorithm can be employed. The reverse modeling algorithm was applied fully to case study two. To this end, a new f_{ann} that was trained from thirty-five training points and the developed model was tested using seventeen testing points. Similar to the preliminary investigation, the $f_{\text{ann,je}}$ was chosen from the $f_{\text{ann,ic}}$ that had the lowest E_{train} where the reversed value came from d_p . Highlighted in Figure 3.26 is the first loop of the $F_{\text{ann,c}}$ for the sample of one hundred sixty-eight apertures. The graph is labeled showcasing the $f_{\text{ann,je}}$ calculated values for x_j and the x_{jc} (and the E_{tol}) as illustrated in the Figure 3.26. In the figure, the y_1 and y_2 values are shown from step five in Chapter 2's reverse modeling pseudo code.

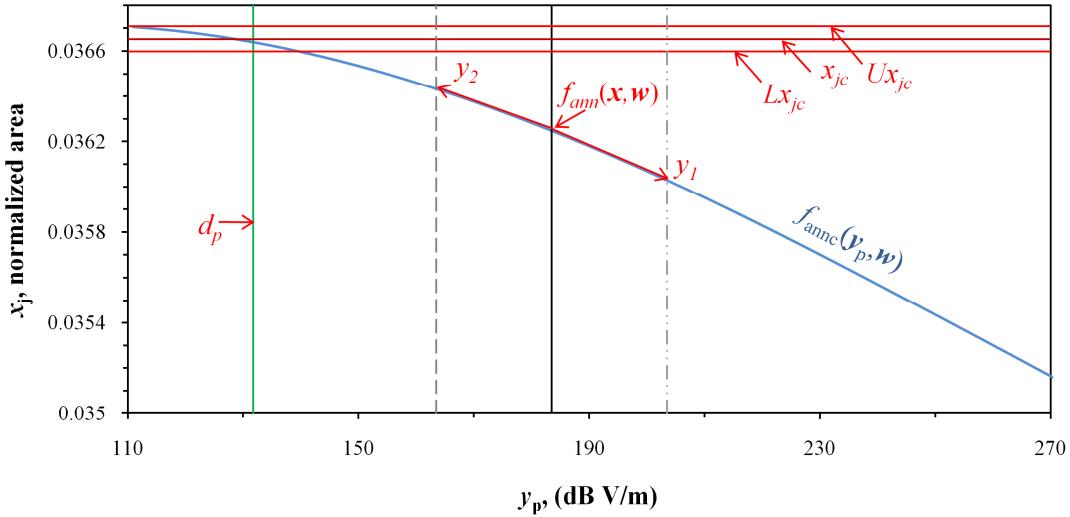


Figure 3.26: Example of step five of the reverse modeling, $F_{\text{ann,c}}$ modeling of the estimation for 168 apertures.

To surmise, y_2 's estimate for x_j is much closer to x_{jc} than y_1 's estimate. Therefore, the next iteration will begin from y_1 's estimate for y_1 . The E_{tol} effect on the reverse model is shown in Figure 3.27. This is for the 168-aperture test as well as a comparison between two reverse modeling techniques and a root finder algorithm [66]. Note that the $f_{\text{ann,jc}}$ was the same for all three estimates.

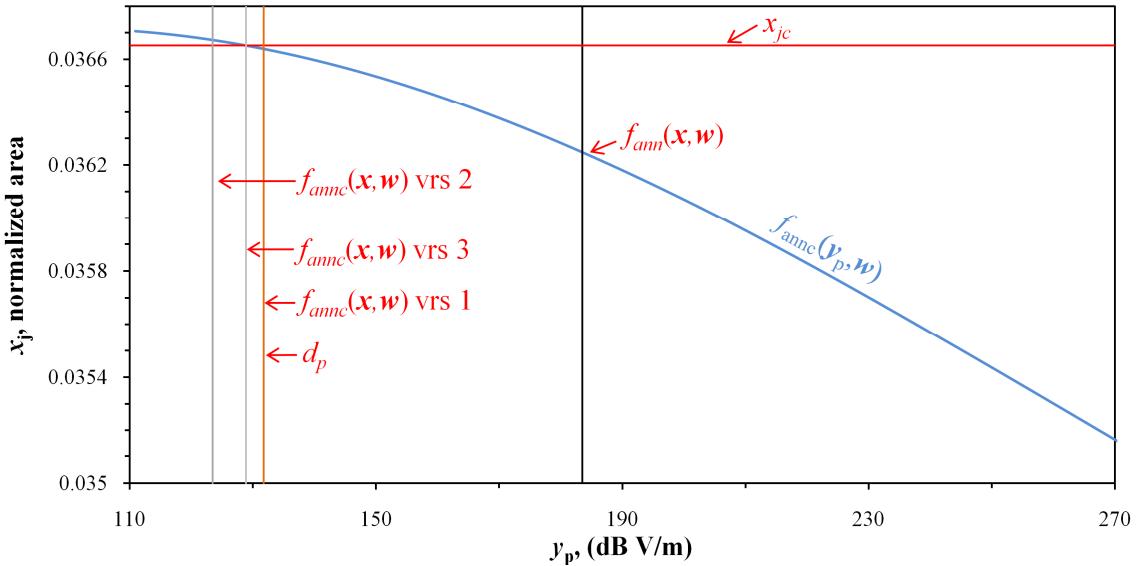


Figure 3.27: Comparison of different versions of $f_{\text{ann, jc}}$ parameter values for $F_{\text{ann,c}}$ modeling of the estimation for 168 apertures.

As shown in Figure 3.27, the $f_{\text{ann},\text{jc}}$ that had tailored Δy and E_{tol} had the lowest error value. For Figure 3.27, the notation of $f_{\text{ann},c}(\mathbf{x}, \mathbf{w})$ vrs 1 was the final estimate ($F_{\text{ann},c}(\mathbf{x}, \mathbf{w})$) and had fixed values where $F_{\text{ann},c}(\mathbf{x}, \mathbf{w})$ could loop fifteen hundred times before it moved on, $\alpha = 20$, and $E_{\text{user}} = 0.1472$. The notation of $f_{\text{ann},c}(\mathbf{x}, \mathbf{w})$ vrs 2 was the final estimate of $F_{\text{ann},c}(\mathbf{x}, \mathbf{w})$ which had variable values for the following: (1) the number of times the algorithm could loop before it moved on, (2) α ; and (3) E_{user} . The notation of $f_{\text{ann},c}(\mathbf{x}, \mathbf{w})$ vrs 3 was the final estimate ($F_{\text{ann},c}(\mathbf{x}, \mathbf{w})$) and instead of utilizing the pseudo code, the $F_{\text{ann},c}(\mathbf{x}, \mathbf{w})$ used a root finder [66].

Table 3.12: Comparison of reverse modeling techniques.

	Training average Error (%)	Testing average Error (%)
f_{ann}	17.460	17.166
$f_{\text{ann},\text{jc}}$	3.02e-2	2.12e-2
$F_{\text{ann},c}^1$	-	11.0724
$F_{\text{ann},c}^2$	-	9.76e-2
$F_{\text{ann},c}^3$	-	5.080

¹ $F_{\text{ann},c}$ had a fixed values for: 1) max of fifteen hundred loops before it moved on, 2) $\alpha = 20$, 3) $E_{\text{user}} = 0.1472$.

² $F_{\text{ann},c}$ had a variable number for the max # of loops, alpha, and E_{user} for each data point.

³ $F_{\text{ann},c}$ instead of using Section 2.7, used a zero root method [66].

As the data in the table shows, the correction algorithm can have differing amounts of error for the different functions. Any of the variations listed in the table reduced the f_{ann} error. The results for $F_{\text{ann},c}^2$ are plotted in Table 3.13 and show that the based reverse modeling approach is very accurate when data points are few. Based on the use of 35 training samples, a significant improvement in the model accuracy as compared to the standard ANN approach was achieved.

Table 3.13: Comparison of d_p values to $f_{\text{ann}}(\mathbf{x}_p, \mathbf{w})$ and $F_{\text{ann,c}}(\mathbf{x}_p, \mathbf{w})$ in the proposed model of Figure 3.23.

Number of Apertures	Normalized Area of the Apertures	Radiated Electric Field $ E_y $		
		HFSS	ANN	Reverse Model
7	0.00152716	237.14	224.68	237.07
13	0.00283616	255.06	224.69	254.83
28	0.00610865	241.08	223.22	240.49
34	0.00741765	237.24	222.51	237.00
45	0.00981748	231.15	218.72	231.48
51	0.01112647	219.68	216.11	219.64
69	0.01505347	204.87	210.57	204.86
73	0.01592613	203.38	209.96	203.44
84	0.01832596	198.13	208.13	198.23
98	0.02138028	189.64	205.20	189.67
102	0.02225295	189.17	204.96	189.22
111	0.02421644	174.75	201.02	174.64
120	0.02617994	176.08	197.64	176.03
133	0.02901610	160.56	194.91	160.56
146	0.03185226	145.14	188.38	145.26
153	0.03337942	146.65	189.97	146.68
168	0.03665191	131.79	183.49	131.74

This study was carried out to find the accuracy of the developed model for a reduced number of training points.

3.7 Comparison of Full-Wave Solution to ANNs

A comparison of the computational overhead required by the proposed ANN models with the full-wave simulations is given in Table 3.14.

Table 3.14: Comparison of computational time and memory requirements for ANNs and HFSS.

		Time per calculation (s)	RAM (Mb)
Case Study One	f_{ann}	0.0224	31.0844
	HFSS	77	759.67
Case Study Two	f_{ann}	0.0224	31.379
	HFSS	79.5	901.56

As can be noted, the trained models are approximately three orders of magnitude faster than HFSS on a given computer. This does not include the time required by HFSS for changing the geometrical details in the computational models. Similarly, the memory re-

uirements for the proposed models are approximately 24 times smaller than *HFSS*. The time per f_{ann} calculation was approximately 22ms, and the memory usage for *NeuroModeler* v1.5 was approximately 31Mb. *HFSS* time per calculation varied with each geometric configuration, and in these case studies, the time for the simulations took on the order of minutes. The memory usage was in the hundreds of mega bytes.

3.8 Summary

To summarize Chapter 3, there were two case studies presented: (1) source variations within a shielding enclosure, and (2) large variations to the number of apertures on a shield. For the first case study, there were many preliminary investigations performed. These preliminary investigations focused on (1) the validity of the case study, (2) how magnitudes affect error, (3) how continuous training data affects error, (4) variations on hidden neurons, (5) how training algorithms affect error, (6) variations to the size of the data sets, (7) variations to the data sets, and (8) the significance of the ANN inputs. Case Study One found that ANNs can successfully estimate radiated emissions for a given source location within a cavity. The second case study's preliminary investigations examined the following: (1) the validity of the study, (2) the affect of a single training data sample on error, and (3) the significance of interpolating training data sets on ANN error. Case Study Two found that ANNs can successfully estimate the radiated emissions from an enclosed source when there are large variations to the number of apertures. Case Study Two also had the reverse modeling algorithm applied, and it was found that the algorithm could reduce ANN error. Finally, the computational resource requirements of ANNs and *HFSS* were compared for Case Study One and Case Study Two, and it was found that ANNs require far less memory and computational time than *HFSS*.

Chapter 4

EM-ANN Modeling of PCB Cavity Structures

The location of connectors exiting a shielded enclosure can have a significant effect on the magnitude of the radiated emissions. As this case study illustrates, the magnitude of the radiated emissions is a function of both frequency and position of the trace on a PCB structure. In this study, an ANN model is created that illustrates the ANN's capability and ability to model EMC-type problems that include frequency as an input. Additionally, this study shows the advantages of unique ANN inputs, such as $|E_0|$, first introduced in Chapter 3.

4.1 PCB Cavity Case Study: Variations of a Trace Position on a PCB

The case study models a partial shield to predict the EMC impact of changing a PCB trace exit location in the shield. The ANNs made in this study utilized *AutoIt!* for ANN creation.

4.1.1 Procedure of PCB Cavity Case Study

As shown in Figure 4.1, this case study on PCB trace modeling is an idealization based on an enclosed source within a shield on top of a PCB. From user experience, the PCB can be represented by a PEC layer with a typical fr4 weaved board thickness (on the order of approximately 30 mils). Real PCB shields seldom cover an entire PCB surface due to

packaging, weight, and cost restrictions. Mostly, shields are custom designed to cover only a device or a particular area on the PCB. Similarly, during an optimization or troubleshooting process in an EMC lab, the shields are usually hand-made to fit a particular configuration. Using ANN models, this trial-and-error process can be made significantly fast.

An important consideration in the design of such shields is exactly where the shields must be placed on a board since radiations from a PCB and shield structure depend on source location and traces running out of the shields. This problem can also be stated to select the optimized points of exit for a PCB trace.

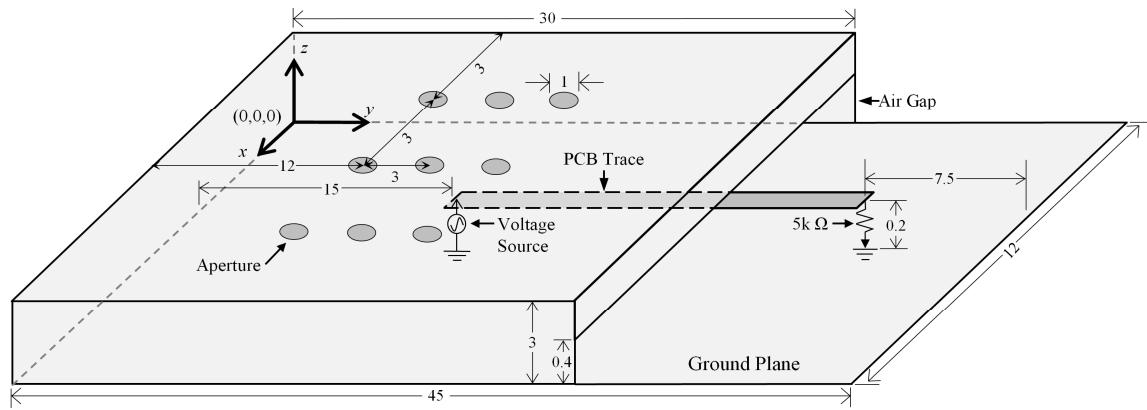


Figure 4.1: A voltage source driving a PCB trace terminated with a load and covered by a partial PCB shield. Dimensions are in centimeters.

As shown in Figure 4.1, a PCB shield encloses a voltage source connected, through a trace, to a resistive load ($5\text{ k}\Omega$) located outside the cavity. The trace is partially covered by the shield. The shield is connected to the ground plane on three sides, and there is a four mm air gap on one side to allow PCB traces to exit. The proposed ANN-based model for the problem is illustrated in Figure 4.2. As shown, the model contains four inputs: (1) X_{offset} , (2) number of apertures on the cavity, (3) frequency, and (4) $|E_0|$. The output was $|E_y|$ (dB V/m) and measured one meter above the *top face*.

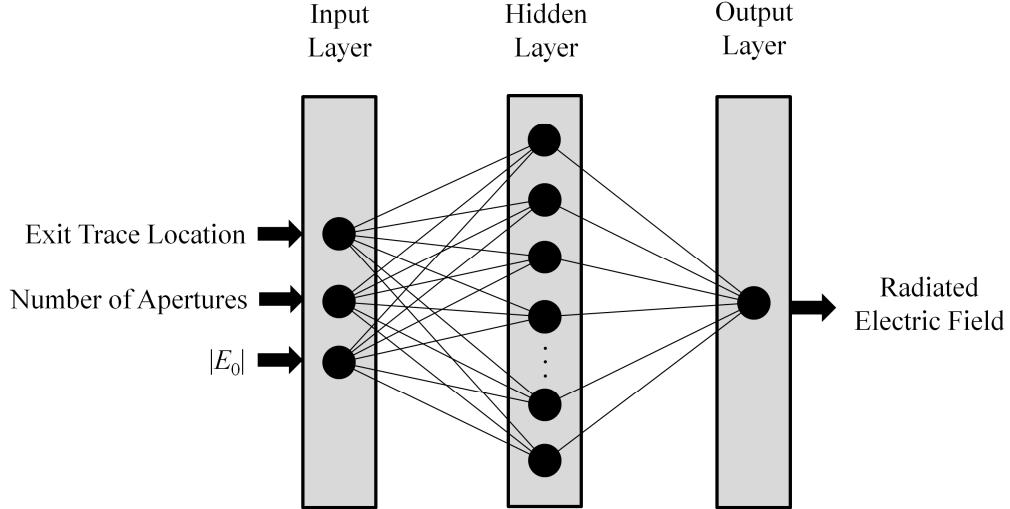


Figure 4.2: ANN model for the structure in Figure 4.1.

The training data for the proposed model was collected by moving the source-trace-load setup along the x-axis in 1 mm increments. For each position, the radiated fields were calculated 1 m above the top face of the shield. This was repeated for frequencies from 1 GHz to 1.3 GHz in the increments of 26 MHz. The fields for $|E_0|$ were calculated with the *top face* removed.

The f_{ann} s created had 10 to 40 hidden sigmoid neurons in increments of 5 neurons, created using QN, QNM, CQN, and CQNM with each f_{ann} configuration created 10 distinct times. This resulted in the creation of approximately 250 f_{ann} s.

4.1.2 Results of PCB Cavity Case Study

Following the procedure outlined, $N_{\text{train}} = 1,508$ training samples were obtained. The model had $N_{\text{test}} = 39$ samples, and the results are plotted in Figure 4.3 and Figure 4.4. The f_{ann} model had 35 hidden sigmoid neurons that were trained using a QNM algorithm.

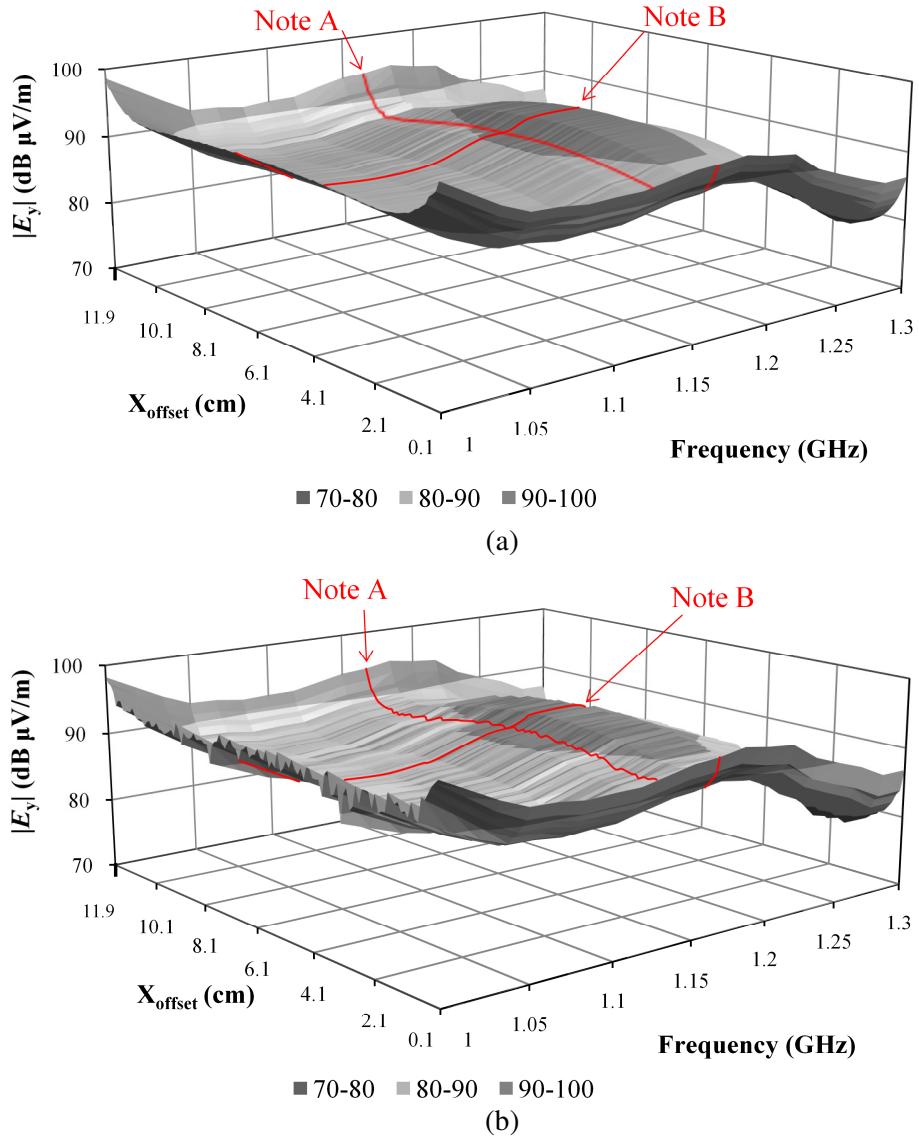


Figure 4.3: Radiated emissions from the PCB shield structure (a) model estimation, (b) *HFSS* estimation.

A close agreement can be seen among the calculated and estimated values. In fact, in Figure 4.3 (b), the largest difference between the ANN estimate and the *HFSS* calculations are 0.74dB as the source's location is varied. Similarly, in Figure 4.3 (b), the maximum error is 0.53 dB as the frequency is varied from 1 GHz to 1.3 GHz. Note A in Figure 4.3 is part (a) in Figure 4.4; similarly, Note B is part (b) in Figure 4.4.

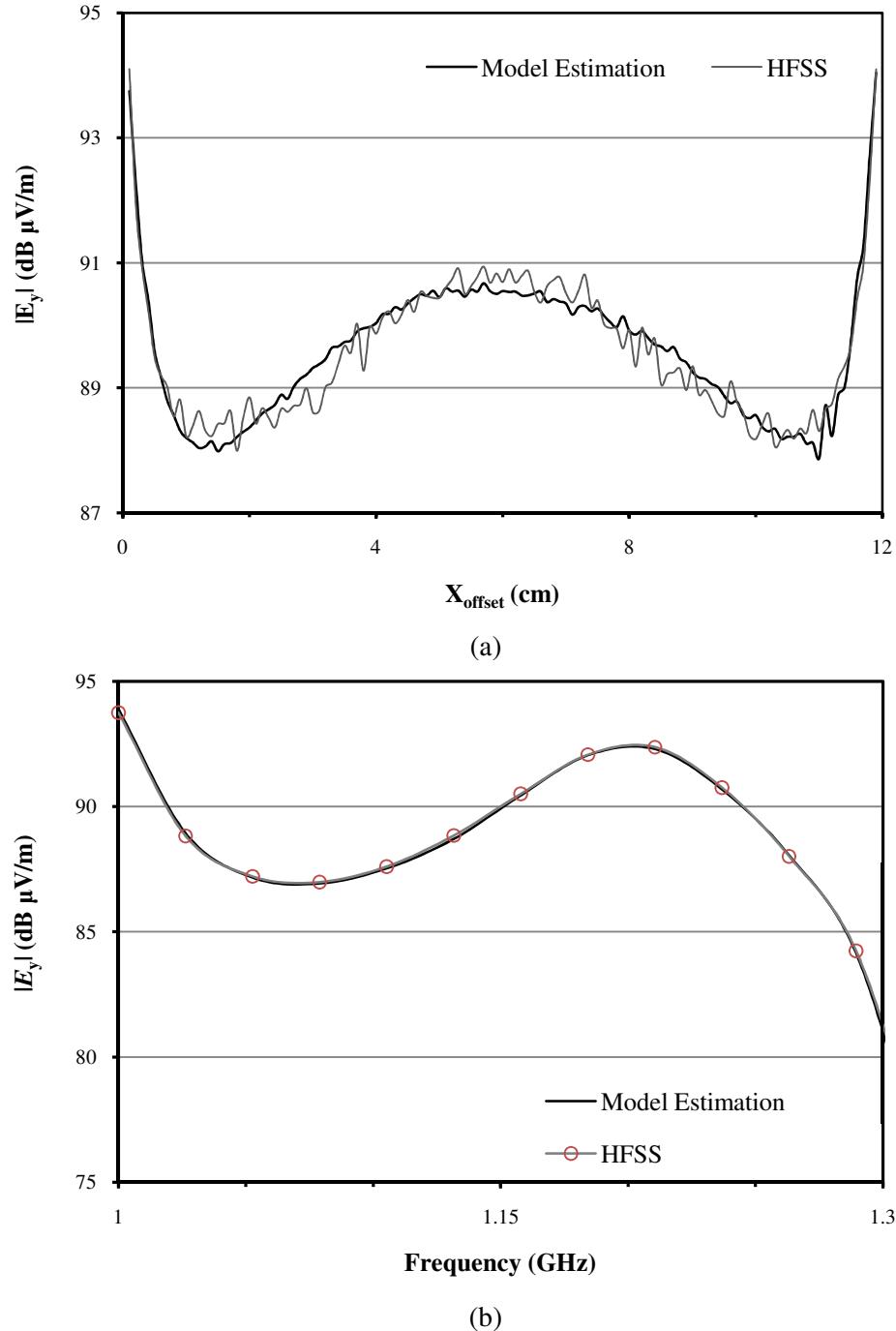


Figure 4.4: Radiated emissions from the PCB shield structure of Figure 4.3. (a) radiated emissions vs. source location, (b) radiated emission vs. frequency.

As seen in Figure 4.4, as the frequency is varied for a X_{offset} value, the f_{ann} model estimate is within 1 dB. However, as the X_{offset} value is varied, the f_{ann} values vary from the d_p val-

ues by approximately 0.2 dB. With the results presented, it is clear that f_{ann} s can estimate the radiated emissions for this configuration.

4.2 Comparison of Full-Wave Solution to ANNs

A comparison of the computational overhead required by the proposed models with the full-wave simulations is given in Table 4.1.

Table 4.1: PCB EMC modeling comparison for time and memory requirements.

		Time per calculation (s)	RAM (Mb)
Case Study	f_{ann}	0.023	30.632
	<i>HFSS</i>	243.3	666.33

As can be noted, the trained models are approximately three orders of magnitude faster than *HFSS* on a given computer. These results in Table 4.1 are similar to Table 3.14.

4.3 Summary

To summarize Chapter 4, there was a case study presented that examined whether ANNs could estimate the radiated emissions from a PCB trace. The shielding on the PCB was similar to the shields presented in Chapter 3; however, this example was unique in that the source enclosed in the shield connected to a trace that extended outside of the shield and terminated on a resistor on the surface of the PCB. The results of the case study found that ANNs could successfully estimate the radiated emissions from a PCB trace as the trace location and frequency was varied. The computational resource requirements of the ANNs and *HFSS* were compared, and it was found that ANNs require far less memory and computational time than *HFSS*.

Chapter 5

ANN Modeling of PCB interconnects

The termination position of PCB interconnects on a board can have a significant effect on the magnitude of the radiated emissions from a PCB. According to [67] [68] [69] [70], the termination position of a PCB interconnect has had a significant effect on radiated emissions from a PCB. Guidelines for PCB termination typically are broadly defined. In general, these broadly defined guidelines are frequency and geometry independent and should be remembered as such. The goal of this chapter is to model PCB radiated emissions from interconnects through using *HFSS*; use that data to train ANNs; and, after training, utilize the trained f_{ann} to estimate the optimal position of a trace termination position. This case study's f_{ann} is further examined for correlations between the N_{train} , number of epochs, E_{train} , E_{test} , memory usage, time per f_{ann} calculation, and time per f_{ann} training session. The optimization technique was to find local minima for a uniform fine mesh for the input space for the f_{ann} . These f_{ann} s can efficiently and quickly estimate the radiated emissions.

5.1 Preliminary Investigation for PCB Interconnect Case Study

Shown in Figure 5.1 is the geometric configuration used in the preliminary investigation for modeling PCB wire interconnects.

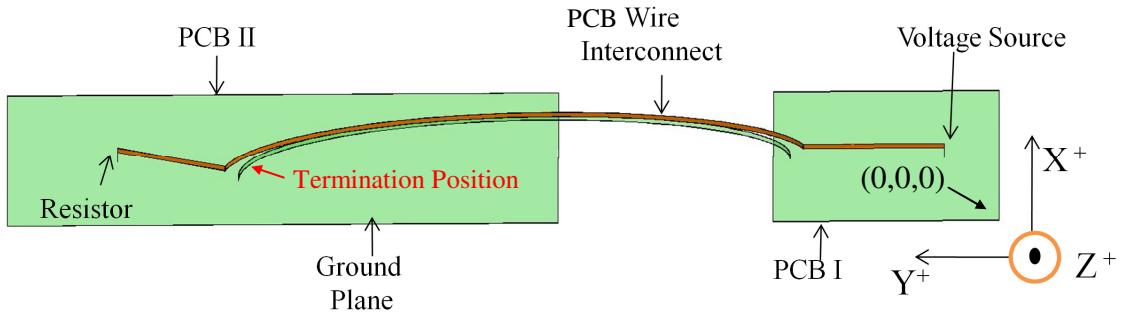


Figure 5.1: Overview of PCB interconnects geometry diagram.

As shown in Figure 5.1, there are two wires (PEC material) interconnecting two PCB planes (PEC) via an equation-based surface, specifically elliptic based such that the arc length of the wire stays relatively constant. PCB I is 3 cm x 2.05 cm x 3.6 μ m, the gap is 2 x 3 cm, PCB II is 3 cm x 5 cm x 3.6 μ m, and the wire has a 3.6 μ m thickness that is 1mm wide. The termination position of the interconnects' location is varied from (0.95 x 5 x 0 cm) to (1.95 x 7 x 0 cm), which is the termination position's boundaries. The termination's location was varied in 1 cm increments as well as the source frequency of the voltage source from 1 GHz to 3 GHz in 50 MHz increments with a 1 V magnitude (the HFSS solution frequency was 3 GHz with a discrete frequency sweep performed). This is outlined in Table 5.1. The arbitrarily small thickness was assigned to the sheets to avoid a non manifold mesher error.

Table 5.1: Preliminary interconnect termination points.

Row Num.	X _{offset} (cm)	Y _{offset} (cm)	Frequency
1	1	5	1GHz to 3GHz in 50 MHz increments
2	2	5	
3	1	6	
4	2	6	
5	1	7	
6	2	7	

The resistor was a $5 \text{ k}\Omega$. $|E_y|$ was measured as the maximum absolute value of the electric field 1 m above the PCBs in respect to the y -direction.

The preliminary investigations on PCB interconnects are outlined in Table 5.1. For these geometries and excitations that had the $|E_y|$ values calculated, the results are shown in Figure 5.2.

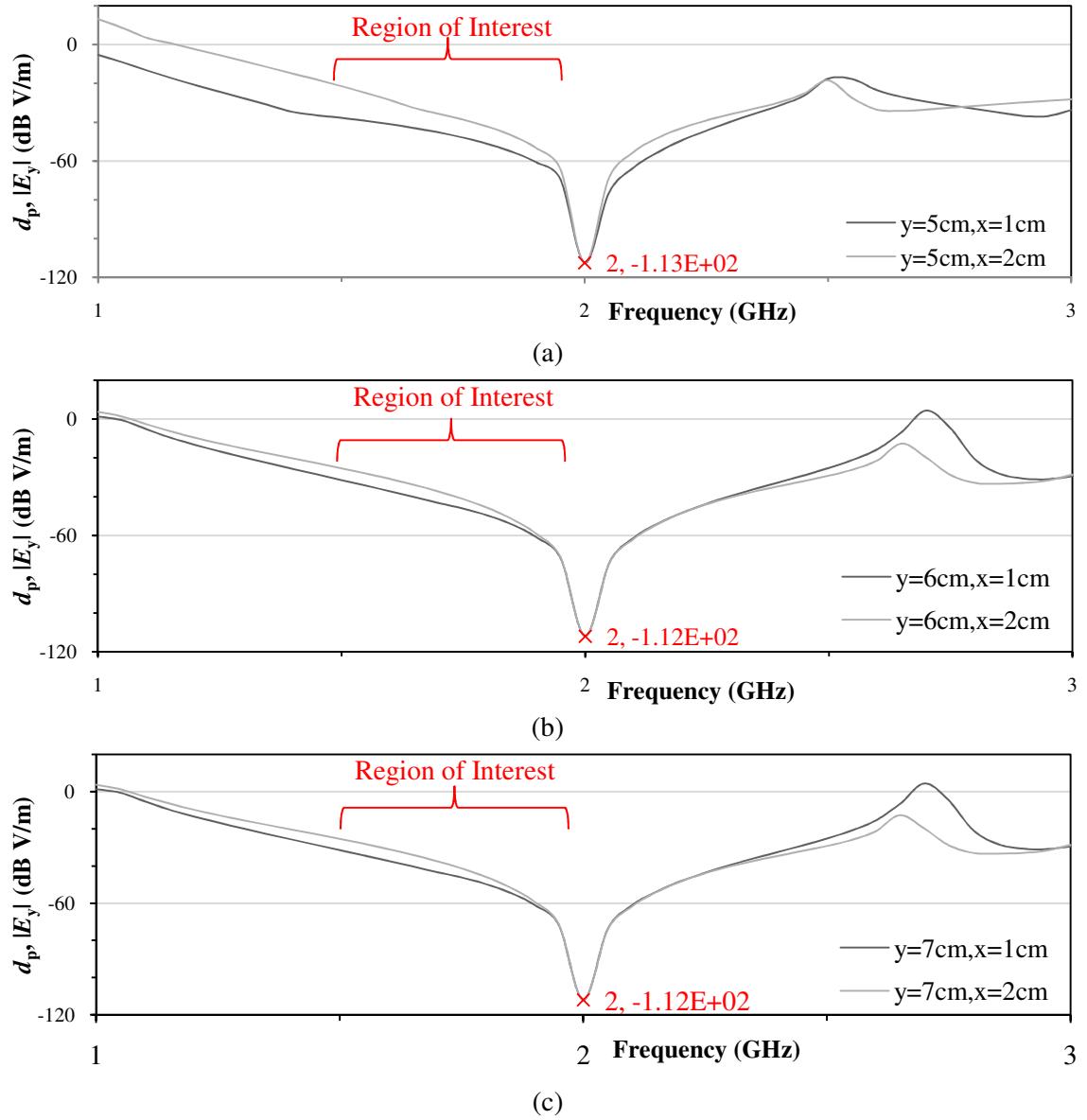


Figure 5.2: Preliminary investigations calculated d_p for the PCB interconnects case study.

Figure 5.2 was reviewed, and it was determined that regions of interest did exist for the simulations, as noted on the figure. Figure 5.2 (a) shows the $d_p |E_y|$ values for rows 1-2 in Table 5.1 and uses circular equation surface for the trace, Figure 5.2 (b) shows the $d_p |E_y|$ values for rows 3-4 in Table 5.1 and uses elliptic equation surface for the trace, and Figure 5.2 (c) shows the $d_p |E_y|$ values for rows 5-6 in Table 5.1 and uses elliptic equation surface for the trace. Figure 5.2 shows that for initial values of d_p , there are frequencies for which the simulation has an optimal trace location that may not be located on the boundary conditions. Examining the graph holistically, min/max magnitudes of the radiated fields are -112 dB to 4.4 dB, respectively, and have a large enough variation to be of interest as an EMC problem as the frequency and position change.

5.2 Case Study on Variations of an Interconnect Termination Point

As mentioned earlier, interconnect termination points can have a significant effect on the magnitudes of the radiated emissions, meaning that if a circuit must use two PCBs, finding an ideal PCB interconnect location could mean a significant reduction in radiated emissions for a given frequency. Although not investigated in this thesis, the converse of this could be correct—*i.e.*, that certain termination points will significantly increase radiated emissions. The design challenge is to decide where to terminate the wire on a PCB such that the far field radiated emissions meet the user criteria.

5.2.1 Procedure of Interconnect Case Study

Given that the preliminary investigation showed the geometry has a solution of interest, the preliminary simulation outlined in Table 5.1 was therefore continued with a finer mesh. There were increments of 0.5 cm for the interconnect's termination position, resulting in a 250% increase for d_p samples available to the training, validation, and testing

sets. Other than the finer mesh, the procedure remained identical to the preliminary investigation. To surmise, there were fifteen geometric configurations simulated with forty discrete frequency points, results in N being approximately six hundred data points. The geometry is shown in Figure 5.1.

After the $|E_y|$ values were simulated, f_{ann} creation began. For this simulation, the data set was divided into $N_{\text{train}} = 605$, $N_{\text{test}} = 5$, and then several more sets were created, shifting d_p points based on frequency from training sets to testing sets such that the increment was approximate 60 points, and that resulted in approximately 9 train/test sets. This can be seen in Table 5.2. The training algorithms used were QN, QNM, CQN, and CQNM. The f_{ann} s were created such that the inputs were frequency, X_{offset} , and Y_{offset} ; the output was $|E_y|$ in V/m; and number of hidden neurons was from 5 to 25 neurons in 5-neuron increments with five perturbed iterations resulting in over nine hundred f_{ann} s being created.

5.2.2 Results of Interconnect Case Study

The results from the ANN creation showed that there were many ANNs successfully created that had an error less than one percent for E_{test} . Not all of these ANNs are shown; however, in Figure 5.3, the best of the eight different data sets are compared. The results of the f_{ann} creation are shown in Table 5.2. As the data in the table shows, the f_{ann} 's created error is tolerable for EMC modeling, specifically with E_{train} s and E_{test} s primarily being at/under three percent.

Table 5.2: f_{ann} creation results for PCB interconnects case study.

f_{ann} Number	Number of Train Samples	Number of Neurons	Training Algorithm Type	Average E_{train} (%)	Average E_{test} (%)
1	90	15	CQN	1.446	3.281
2	165	5	QN	3.204	3.144
3	240	20	CQNM	1.240	2.663
4	315	10	QNM	1.549	3.014
5	390	5	QN	0.942	1.199
6	465	15	QN	0.870	0.328
7	540	5	CQNM	0.712	0.325
8	600	20	QN	0.796	0.201

5.2.3 Investigations on Variations to the Number of N_{train}

As shown in Table 5.2, the size of the training sets has an effect on the average E_{test} .

Graphically, there is a general trend that the E_{test} decreases as N_{train} increases, as shown in Figure 5.3.

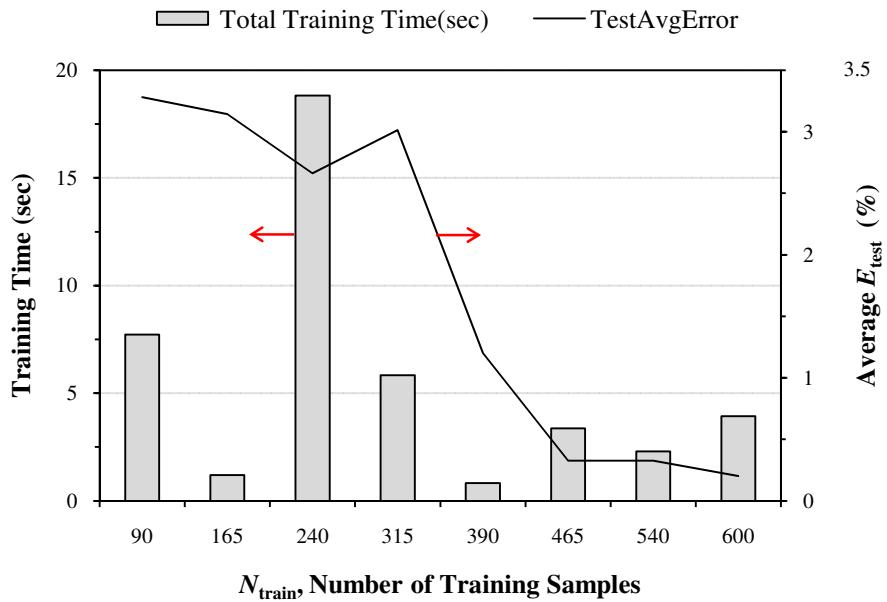


Figure 5.3: Investigation on N_{train} vs. train time and average E_{test} for the f_{ann} s with each data set's lowest average E_{test} .

Also shown in Figure 5.3 are the training times for the f_{ann} s. The expectation was that as the number of training points decreases, the training times should decrease. However,

from inspection, comparing the number of training samples to the training times in this data suggests that the two are not directly related.

Further examining training parameters as a function of N_{train} , Figure 5.4 shows a correlation between the number of training epochs and training times, where the f_{anns} were selected by having the lowest average E_{test} for each data set. Figure 5.4 shows that there is a general correlation between the number of epochs and the training time. The comparison between the two shows that there is a general normalized epoch scale factor which is approximately 1.2 seconds per one hundred epochs.

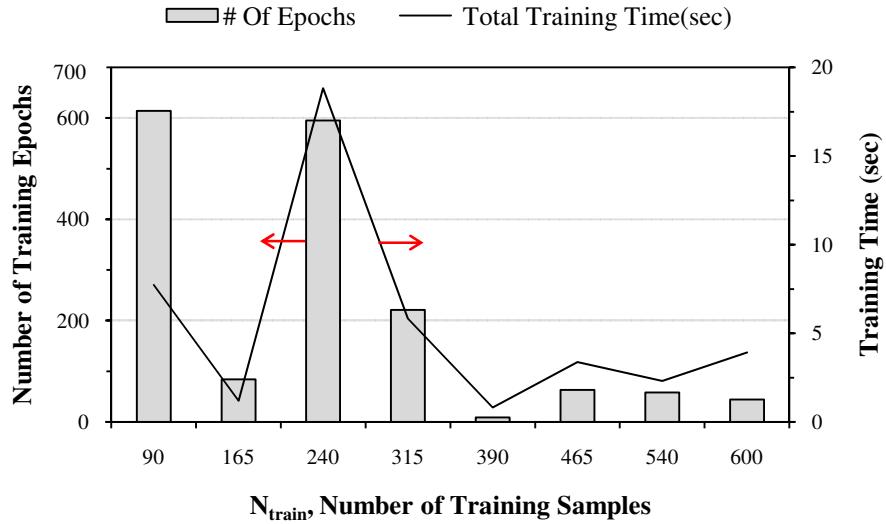


Figure 5.4: Investigation on N_{train} vs. number of train epochs and train times.

The number of epochs vs. the number of training samples has no clear correlation; however, this could be due to the indirect comparison being made between f_{anns} . Similarly, also of significance is that there is no clear correlation for training time vs. the number of training samples.

As shown in the figures, data has been presented that gives some experimental evidence to the ANN recommendations presented in Chapter 2. There appears to be a corre-

lation between training times and number of training epochs and the number of training samples.

5.2.4 Investigations on Repetitive f_{ann} Creation

One area of interest given that ANN creation is a stochastic process is how to see how consistent training times and training error are as a function of the f_{ann} iteration. This is shown in Figure 5.5.

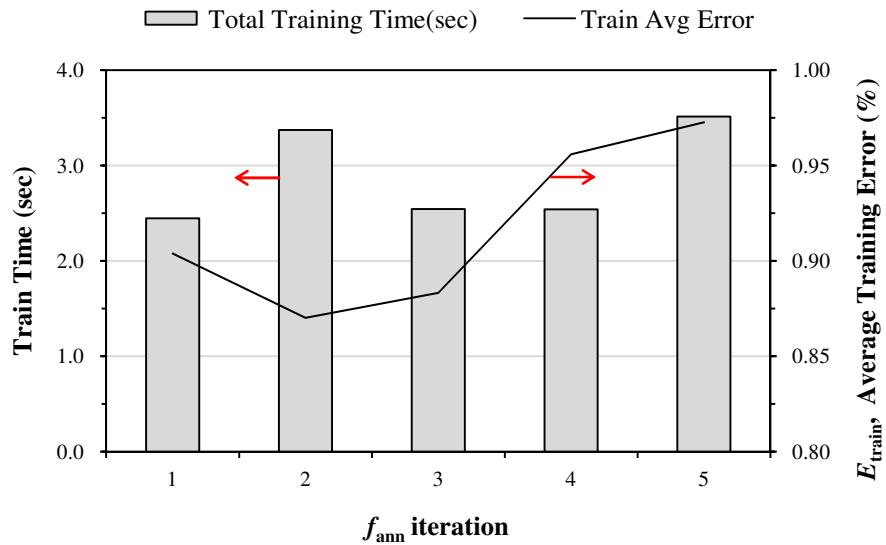


Figure 5.5: f_{ann} iteration vs. train time/average train error for f_{ann} number seven found in Table 5.2.

As shown in Figure 5.5, the training time has some variation, within a range of approximately 1 s for training time. This suggests for this f_{ann} that there is consistency between iterations in training time. E_{train} varies from approximately 0.95% to 0.975%, suggesting that the average E_{train} stays approximately the same as a function of f_{ann} iteration as well. Examining iteration 2, as training time increases, the training error slightly decreases; however, given the small variations to E_{train} , it would seem that there is, if any, for this f_{ann} , little correlation between training time and training error.

Another metric of interest as a function of f_{ann} iteration would be memory usage and f_{ann} calculation time. These are significant in case the user held f_{ann} calculation time and memory usage as significant, *i.e.*, applications where computational times are paramount. This is shown in Figure 5.6.

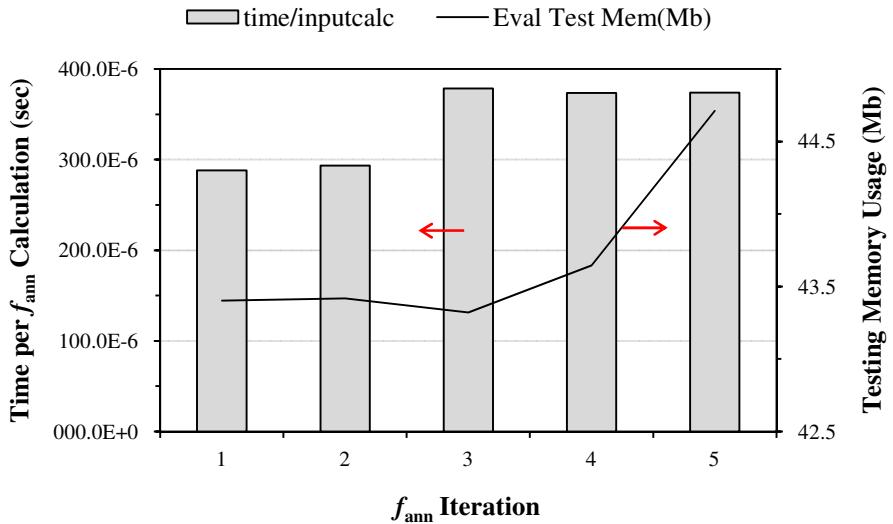


Figure 5.6: f_{ann} iteration vs. time per f_{ann} calculation and testing's memory usage for f_{ann} number seven found in Table 5.2.

The time per f_{ann} calculation varies from approximately 290 μ s to 370 μ s, and the memory usage varies from 43.2 Mb to 44.7 Mb. Again, these standard variations from the mean are 0.40 Mb and 40.6 μ s. The significance of these variations is left to the user's discretion. For this thesis, most of the data collected had similar variations, suggesting that iteration is relatively consistent.

5.3 Optimization of PCB Terminations Utilizing ANNs

In design work, optimizations based on simulation models can provide significant model improvements. Similar to industry constraints for PCB design, the interconnects are given a small region on the total PCB board to terminate, and the locations of the termination

have significant emissions to warrant an investigation in the design in order to make the PCB compliant with local regulations.

In this section, a f_{ann} is utilized to estimate the radiated emissions for a given interconnect position and frequency. The f_{ann} utilized was f_{ann} number seven found in Table 5.2. The optimization technique utilized is a non adaptive method that creates a fine uniform mesh of calculation points for a given criteria, such as frequency, X_{offset} , and Y_{offset} . After f_{ann} values are calculated, the user criteria are then utilized to determine suitable location(s) for the interconnect. This is then considered the optimized location for that given criteria.

In this example, the criteria set was that the optimized termination location was to be found for a given frequency value such that the location gave the smallest radiated emissions. Given that this f_{ann} took about 293 μs to calculate one data point, it means that for a thousand estimations, f_{ann} calculations would take 0.293s to finish. According to the author, a non adaptive optimization technique is suitable given how quickly f_{ann} s calculate one estimation. If the user requires a more efficient algorithm, it is recommended that f_{ann} uses an adaptive process. For this case, the f_{ann} was successfully created, and the optimal termination locations were calculated.

The results of the optimization are shown in Table 5.3. The fine mesh was set for a 2000 (X_{offset}) by 2000 (Y_{offset}) mesh for each frequency point. Included in the table as well are the estimated radiated emissions in dB V/m for consistency; however, the values calculated were in V/m.

Table 5.3: PCB optimization f_{ann} results.

Frequency (GHz)	X_{offset} (cm)	Y_{offset} (cm)	$f_{\text{ann}}(\mathbf{x})$'s $ E_y $ (dB V/m)
1.00	1.00	5.00	-3.47
1.05	1.31	5.00	-7.83
1.10	1.31	5.00	-12.17
1.15	1.32	5.00	-15.73
1.20	1.32	5.00	-18.47
1.25	1.32	5.00	-21.35
1.30	1.32	5.00	-26.84
1.35	1.32	5.00	-59.61
1.40	1.70	5.66	-154.35
1.45	1.90	6.25	-198.66
1.50	1.33	5.42	-131.70
1.55	1.29	5.28	-137.45
1.60	1.53	6.34	-145.78
1.65	1.35	5.59	-125.11
1.70	1.28	5.26	-143.68
1.75	1.91	5.17	-146.04
1.80	1.57	6.49	-133.09
1.85	1.29	5.36	-144.68
1.90	1.34	5.01	-148.57
1.95	1.45	6.11	-148.70
2.00	1.27	5.30	-144.34
2.05	1.83	5.85	-142.34
2.10	1.34	5.62	-150.18
2.15	1.64	6.89	-161.91
2.20	1.95	6.31	-132.56
2.25	1.65	6.18	-149.47
2.30	1.56	6.54	-152.73
2.35	1.50	6.27	-127.95
2.40	1.73	6.62	-139.49
2.45	1.67	6.69	-152.73
2.50	1.80	6.92	-126.59
2.55	1.71	6.97	-122.50
2.60	1.00	7.00	-37.33
2.65	1.00	7.00	-37.33
2.70	1.00	7.00	-37.33
2.75	1.00	7.00	-37.24
2.80	1.00	7.00	-32.33
2.85	1.48	5.00	-25.94
2.90	1.48	5.00	-25.94
2.95	1.48	5.00	-25.94
3.00	1.47	5.00	-25.94

← Test Case

As Table 5.3 shows, the optimal offset values shift as a function of frequency in both the X_{offset} and Y_{offset} locations, and in some cases, the optimal position is not located on the

boundaries. This suggests that inside the data input space there is an optimal position. These samples of interest are highlighted with italics in Table 5.3 rows.

Examining one of these optimized frequency-based termination points in more depth would provide additional insight in the ANN estimation for the optimized position. Looking at 1.9 GHz in further depth, shown below in Figure 5.7 is the f_{ann} estimate as a function of the position.

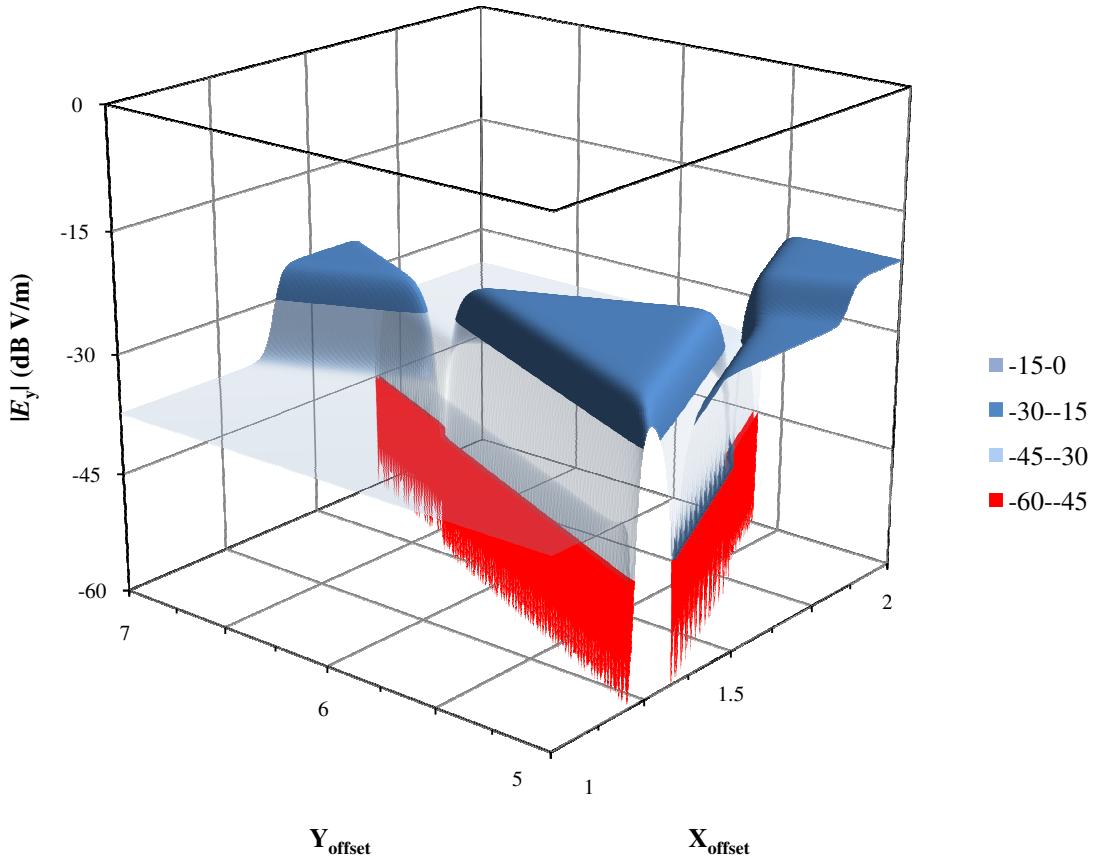


Figure 5.7: The f_{ann} estimate for $|E_y|$ at 1.9 GHz.

As shown in Figure 5.7, the radiated emissions have a minimal value in the region from negative sixty dB V/m to negative forty-five dB V/m; this is defined as the red region. In this region, what appear to be singularities are most probably the discretization of the

hidden neuron functions, meaning that this likely is a f_{ann} systematic error. The radiated emissions will not vary by approximately 20 dB that rapidly in those regions.

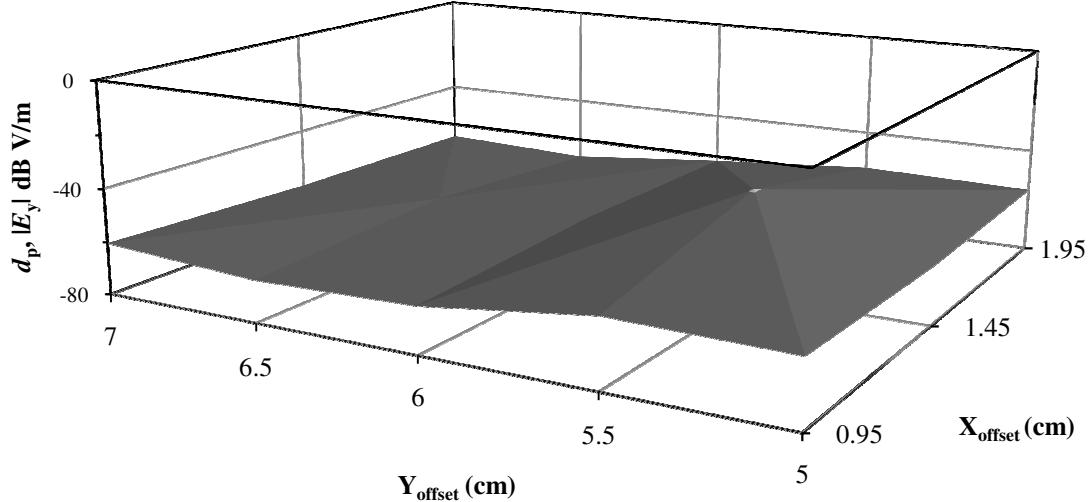


Figure 5.8: Physical model estimates for 1.9 GHz for the PCB interconnect location.

As shown in Figure 5.8, the magnitude of the second derivative with respect to position can be described as continuous in those regions such that the spikes in the f_{ann} look more like troughs in HFSS.

Given that the red regions in Figure 5.7 are probabilistic regions with higher systematic error associated with the ANNs, the author believes that the optimization from f_{ann} s provide a placement guideline because the error could be large at the singularities. This is illustrated by re-graphing Figure 5.7 as a contour graph, as shown in Figure 5.9. The optimal placement, as highlighted in Figure 5.9, was the calculated best interconnect termination position for the trace at 1.9 GHz.

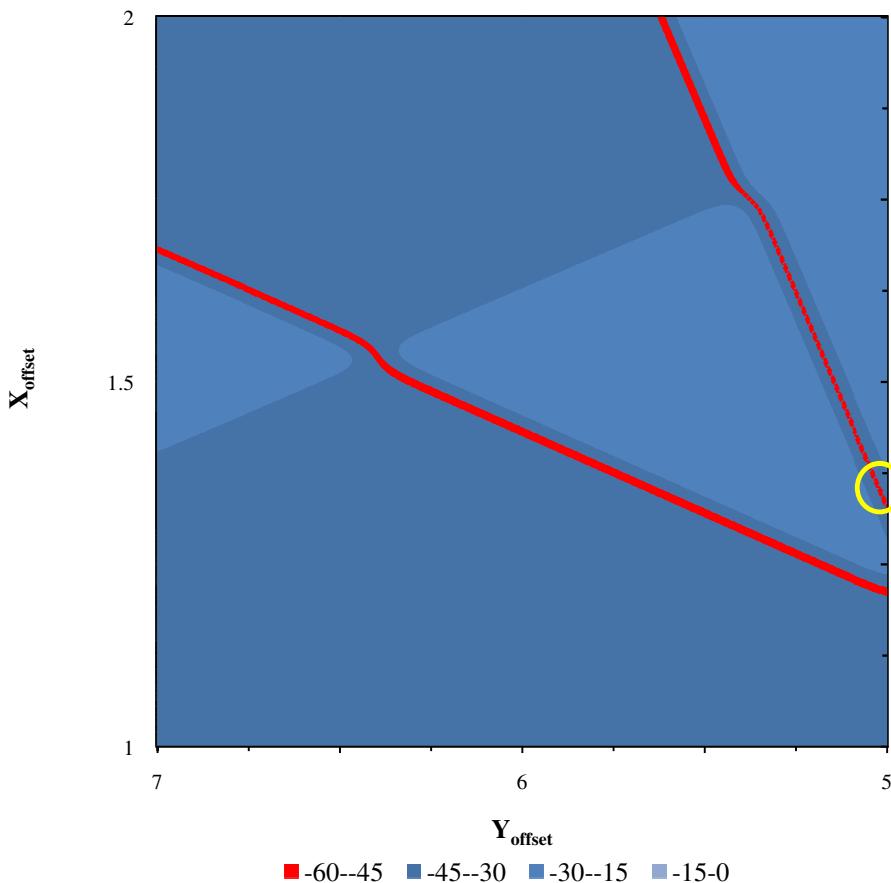


Figure 5.9: Contour graph of the f_{ann} estimate for $|E_y|$ as a function of position where the source frequency is 1.9 GHz.

Given the nature of ANN error, as illustrated in Figure 5.7's discussion, the author surmises that if the user wants to minimize radiation for 1.9 GHz, placing the interconnect anywhere in the -60 to -45 dB V/m regions would minimize radiated emissions.

In summary, this chapter presented an EMC modeling problem of PCB trace interconnects where the optimal termination position on a PCB is unknown. This was accomplished by simulating the problem in *HFSS* and gathering enough data to train f_{ann} s. After f_{ann} creation, one f_{ann} was selected and utilized to infer the optimal position of the interconnect given a solution frequency for a given region on the PCB.

5.4 Summary

To summarize Chapter 5, a PCB interconnect case study was presented. For this case study, a PCB interconnect's termination location on a PCB was varied along with the source frequency, and the radiated emissions were examined. ANNs were found to be successful in estimating the radiated emissions as a function of frequency and the interconnect's termination location. Investigations were conducted that examined the following: (1) N_{train} affects on ANN errors, the number of training epochs, and training times, and (2) the significance of random initialization network values on ANN errors, ANN estimation times, memory usage, and training times. Then an ANN was utilized to find the location of the interconnect for a given frequency which would have the least amount of radiated emissions.

Chapter 6

Conclusions and Future Work

This thesis investigated ANN modeling capabilities for solving common EMI/EMC problems, such as radiated emissions from shielding enclosures, PCB structures, and PCB interconnects. For the cases where data are limited, a reverse modeling-based algorithm was introduced. Also presented was a case study utilizing an ANN for design optimization. Based on the results, it can be concluded that the neural network models may provide very good estimates of “what if” scenarios and, therefore, may be used to make accurate design decisions in an efficient manner. This is particularly useful for products which cannot be modeled using existing tools. However, since accurate measurement data may not be available, the proposed approach could be used to avoid manufacturing multiple prototypes, as is a common practice in the electronics industry.

6.1 Contributions of this Thesis

The contributions of this thesis are to (1) introduce the background of EMC issues, specifically of techniques used to solve EMC issues with their respective advantages and disadvantages; (2) conduct investigations on ANN-based approaches to resolve EMC issues as illustrated by several exemplary structures; (3) illustrate reverse modeling techniques; (4) provide a critical review of the ANN-based approaches as compared to other

techniques, and (5) present a case study of PCB optimization utilizing ANNs for reduction of radiated EM. The research that formed this thesis was also presented at a national conference and submitted for publication.

- Presented “A Neural Network-Based Approach to Estimate the Electric Field from a Source within a Perforated Enclosure” at the Ninth Annual Sigma Xi Student Research Conference on Nov. 14, 2009 in Woodlands, Texas, and received a medal engraved with “Sigma Xi Student Research Conference Superior Presentation” as well as a Certificate of Recognition.
- Submitted and currently in revisions of an article entitled “A New ANN Based Modeling Approach for EMI/EMC Analysis of PCB and Shielding Structures.” This article was submitted to *IEEE Transactions on Electromagnetic Compatibility*, a peer-reviewed journal.
- Submitted and accepted abstract for a conference paper entitled “ANN-Based Electromagnetic Compatibility Modeling of Electronic Systems.”
- Developed an ANN automation tool that makes the creation of ANNs significantly easier while using the *NeuroModeler* tool.

6.2 Future Work

In Chapter 3, two case studies were presented as well as preliminary investigations. The preliminary investigations highlighted some ANN creation-type questions/scenarios and illustrated the practicality of the EMC modeling. The first study examined the radiated emissions from a source within a cavity with sequential variations to the number of apertures. For this study, future work includes incorporation of the reverse modeling technique and optimization protocols with the presented data in order to provide more

case studies for improving ANN quality as well as utilization for design scenarios. The second study in the chapter focused on the large number of variations to the number of apertures. In this study, sequentially hundreds of apertures were added to a shield, and the radiated emissions were estimated by a f_{ann} . Also in this study, a reverse modeling technique was presented that improved a f_{ann} 's error considerably, highlighting the merits of the technique. Future work for this study would be to utilize the normal f_{ann} model and the correction f_{ann} with a design problem, again to further the case studies with design optimization scenarios.

In Chapter 4, a case study was presented showing the radiated emissions from a trace extending from an enclosed source to outside the shield, terminating on a PCB resistor. This explored f_{ann} creation for variations to frequency as well as a distinct geometric example. Future work would be to utilize these f_{ann} s for a design optimization scenario as well as for the reverse modeling algorithm, therefore utilizing it as an additional case study.

In Chapter 5, a PCB interconnect geometry was simulated, and ANNs were created. These ANNs were then utilized for an optimization problem where the termination location of the interconnect for a specific frequency was obtained that minimized radiated emissions. A variety of f_{ann} s were created, examining in greater depth the effects of N_{train} on E_{train} , training times, as well as how closely iterative f_{ann} s had variations in memory usage and calculation times (training and testing). Additional investigations for this study would be to apply a reverse modeling algorithm such that this case study would add additional results for the reverse modeling technique.

For all of the case studies, additional future work would be investigations into increasing the number of effective input parameters. These input parameters could include but are not limited to diameter of apertures, dimensions of the cavity/geometry, excitation type and dimensions, frequency, materials (types, anisotropic properties etc.), number of apertures, number of sources etc. Similarly, the number of output parameters could be increased as well.

To summarize, this thesis shows how f_{anns} can be utilized for modeling EMC-type problems, specifically estimating the radiated emissions of a source from PCB type geometry in four distinct case studies. Also presented was a reverse modeling technique that could improve f_{anns} as well as a case study utilizing the f_{anns} for design optimization based on user criteria. Future work, as outlined, would be utilization of the reverse algorithm and optimization protocols for all the experiments, thereby giving additional case studies.

References

- [1] C. R. Paul, *Introduction to Electromagnetic Compatibility*, Second Edition ed. New Jersey, United States: John Wiley & Sons, 2006.
- [2] D. T. H. Hubing, "Survey of Numerical Electromagnetic Modeling Techniques," Intel Corporation Review and summary on numerical electromagnetic modeling techniques TR91-1-001.3, 1991.
- [3] T. Hubing and A. Orlandi, "A Brief History of EMC Education," in *Proc. of the 16th International Zurich Symp. and Technical Exhibition on EMC*, Zurich, 2005, pp. 95-97.
- [4] D. M. Pozar, *Microwave Engineering*, 3rd ed., B. Zobrist, P. Kulek, and G. Aiello, Eds. Danvers, MA, USA: John Wiley & Sons, 2005.
- [5] T. Hubing, "PCB EMC design guidelines: a brief annotated list," *Proc. IEEE Int. Symp. Electromagnetic Compatibility*, vol. 1, pp. 34-36, Jul. 2003.
- [6] D. M. Hockanson, et al., "Investigation of fundamental EMI source mechanisms driving common-mode radiation from printed circuit boards with attached cables," *IEEE Transactions on Electromagnetic Compatibility*, vol. 38, no. 4, pp. 557-566, Nov. 1996.
- [7] M. Ramdani, et al., "The Electromagnetic Compatibility of Integrated Circuits-Past, Present, and Future," *IEEE Transactions on Electromagnetic Compatibility*, vol. 51, no. 1, pp. 78-100, Feb. 2009.
- [8] E. Sicard, S. Ben Dhia, M. Ramdani, and T. Hubing, "EMC of Integrated Circuits: A Historical Review," in *Proc. of the 2007 IEEE International Symposium on Electromagnetic Compatibility*, Honolulu, 2007.
- [9] J. Lu and F. Dawson, "EMC computer modeling techniques for CPU heat sink simulation," *IEEE Transactions on Magnetics*, vol. 42, no. 10, pp. 3171-3173, Oct. 2006.

- [10] T. V. Yioultsis, T. I. Kosmanis, I. T. Rekanos, and T. D. Tsiboukis, "EMC Analysis of High Speed On Chip Interconnects via a Mixed Quasi-Static Finite Difference-FEM Technique," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1365-1368, Apr. 2007.
- [11] Q.-F. Liu, W.-Y. Yin, J.-F. Mao, and Z. Chen, "Accurate characterization of shielding effectiveness of metallic enclosures with thin wires and thin slots," *IEEE Transactions on Electromagnetic Compatibility*, vol. 51, no. 2, pp. 293-300, May 2009.
- [12] R. Araneo and G. Lovat, "Fast MoM analysis of the shielding effectiveness of rectangular enclosures with apertures, metal plates, and conducting objects," *IEEE Transactions on Electromagnetic Compatibility*, vol. 51, no. 2, pp. 274-283, May 2009.
- [13] A. Kriz and W. Milliner, "Analysis of the CISPR 25 component test setup," *Proc. IEEE Int. Symp. Electromagnetic Compatibility*, vol. 1, pp. 229-232, May 2003.
- [14] H. Rakouth, C. Cammin, and L. Comstock, "The effect of antenna ground impedance on CISPR 25 measurements," *Proc. IEEE Int. Symp. Electromagnetic Compatibility*, no. 1, pp. 453-457, Aug. 2001.
- [15] C. Su, H. Ke, and T. Hubing, "Overview of Electromagnetic Modeling Software," *Annual Review of Progress in Applied Computation Electromagnetics*, vol. 25, pp. 736-741, Mar. 2009.
- [16] Applied Electromagnetics Research Group, "Intermediate Level Modelling Tools for EMC Design," University of Nottingham and York, York, Final Report GR/L-89723 & GR/L-89716, 2000.
- [17] M. Li, et al., "An EMI Estimate for Shielding Enclosure Evaluation," *IEEE Transactions on Electromagnetic Compatibility*, vol. 43, no. 3, pp. 295-304, Aug. 2001.
- [18] M. P. Robinson, et al., "Analytical Formulation for the Shielding Effectiveness of Enclosures with Apertures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 40, no. 3, pp. 240-248, Aug. 1998.
- [19] W. Wallyn, D. D. Zutter, and E. Laermans, "Fast Shielding Effectiveness Predictions for Realistic Rectangular Enclosures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 45, no. 4, pp. 639-643, Nov. 2003.

- [20] T. H. Hubing and M. W. Ali, "EMC Applications of EMAP-2: A 3D Finite Element Modeling Code," in *Proceedings of the 1993 IEEE International Symposium on Electromagnetic Compatibility*, Dallas, 1993, pp. 279-283.
- [21] I. Fatkullin and J. S. Hesthaven, "Adaptive High-Order Finite-Difference Method for Nonlinear Wave Problems," *Journal of Scientific Computing*, vol. 16, no. 1, pp. 47-67, Mar. 2001.
- [22] C. J. Reddy, M. D. Deshpande, C. R. Cockrell, and F. B. Beck, "Finite Element Method for Eigenvalue Problems in Electromagnetics," NASA NASA Technical Paper 3485, 1994.
- [23] J. W. H. Meijs, O. W. Weier, M. J. Peters, and A. V. Oosterom, "On the Numerical Accuracy of the Boundary Element Method," *IEEE Transactions on Biomedical Engineering*, vol. 36, no. 10, pp. 1038-1049, Oct. 1989.
- [24] Q. F. Liu, W. Y. Yin, M. F. Xue, J. F. Mao, and Q. H. Liu, "Shielding Characterization of Metallic Enclosures with Multiple Slots and a Thin-Wire Antenna Loaded: Multiple Oblique EMP Incidences With Arbitrary Polarizations," *IEEE Transactions on Electromagnetic Compatibility*, vol. 51, no. 2, pp. 284-292, May 2009.
- [25] T. Konefal, J. F. Dawson, A. C. Marvin, M. P. Robinson, and S. J. Porter, "A Fast Circuit Model Description of the Shielding Effectiveness of a Box With Imperfect Gaskets or Apertures Covered by Thing Resistive Sheet Coatings," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 1, pp. 134-144, Feb. 2006.
- [26] H. Ke, T. Hubing, and F. Maradei, "Full-Wave Electromagnetic Modelling from DC to GHz using FEM-SPICE," in *Asia-Pacific International Symposium on Electromagnetic Compatibility*, Beijing, 2010, pp. 520-523.
- [27] T. H. Hubing, M. W. Ali, and G. K. Bhat, "EMAP: A 3-D Finite Element Modeling Code for Analyzing Time-Varying Electromagnetic Fields," *Applied Computational Electromagnetics Society Journal*, vol. 8, no. 1, pp. 116-124, 1993.
- [28] H. W. Ott, *Noise Reduction Techniques in Electronic Systems*, 2nd ed. New York, USA: John Wiley & Sons, 1988.
- [29] T. H. Hubing and T. P. V. Doren, "New EMC Design Guidelines for Multilayer Printed Wiring Boards," in *Proceedings of the IEEE Tropical Meeting on Electrical Performance of Electronic Packaging*, 1992, pp. 110-112.

- [30] P. Burrascano, S. Fiori, and M. Mongiardo, "A Review of Artificial Neural Networks Applications in Microwave Computer-Aided Design (Invited Article)," *International Journal of RF and Microwave Computer-Aided Engineering*, vol. 9, no. 3, pp. 158-174, Apr. 1999.
- [31] F. Wang, V. K. Devabhaktuni, and Q. J. Zhang, "A Hierarchical Neural Network Approach to the Development of a Library of Neural Models for Microwave Design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, no. 12, pp. 2391-2403, Dec. 1998.
- [32] P. Dehkhoda, A. Tavakoli, and R. Moini, "Fast Calculation of the Shielding Effectiveness for a Rectangular Enclosure of Finite Wall Thickness and with Numerous Small Apertures," *Progress In Electromagnetics Research*, vol. 86, pp. 341-355, 2998.
- [33] X. Ding, et al., "Neural Network Approaches to Electromagnetic Based Modeling of Passive Components and Their Applications to High Frequency and High Speed Nonlinear Circuit Optimization," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 1, pp. 436-449, Jan. 2004.
- [34] J. W. Bandler, M. A. Ismail, J. E. Rayas-Sanchez, and Q. J. Zhang, "Neuromodeling of microwave circuits exploiting space-mapping technology," *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 12, pp. 2417-2427, 1999.
- [35] V. K. Devabhaktuni, M. Yagoub, and Q. J. Zhang, "A robust algorithm for automatic development of neural-network models for microwave applications," *IEEE Transactions on Microwave Theory and Technique*, vol. 49, pp. 2282-2291, 2001.
- [36] V. K. Devabhaktuni, C. Xi, F. Wang, and Q. J. Zhang, "Robust Training of Microwave Neural Models," *International Journal of RF and Microwave Computer Aided Engineering*, vol. 12, no. 1, pp. 109-124, Dec. 2001.
- [37] F. Wang and Q. J. Zhang, "Knowledge-based neural models for microwave design," *IEEE Transactions on Microwave Theory and Techniques*, no. 12, pp. 2333-2343, 1997.
- [38] F. Wang, V. K. Devabhaktuni, and Q. J. Zhang, "A hierarchical neural network approach to the development of a library of neural models for microwave design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, no. 12, pp. 2391-2403, Dec. 1998.
- [39] A. H. Zaabab, Q. J. Zhang, and M. S. Nakhla, "A neural network modeling approach to circuit optimization and statistical design," *IEEE Transactions on Microwave Theory and Techniques*, vol. 43, no. 6, pp. 1349-1358, Jun. 1995.

- [40] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuits," *IEEE Transactions on Microwave Theory and Techniques*, pp. 2495-2503, Dec. 1996.
- [41] P. M. Watson and K. C. Gupta, "Design and optimization of CPW circuits using EM-ANN models for CPW components," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 12, pp. 2525-2523, 1997.
- [42] G. L. Creech, B. J. Paul, C. D. Lesniak, T. J. Jenkins, and M. C. Calcaterra, "Artificial neural networks for fast and accurate EM-CAD microwave circuits," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 12, pp. 794-802, May 1997.
- [43] Y. Fang, M. Yagoub, F. Wang, and Q. J. Zhang, "A new macro modeling approach for nonlinear microwave circuits based on recurrent neural networks," *IEEE Transactions on Microwave Theory and Techniques*, vol. 48, no. 12, pp. 2335-2344, Dec. 2000.
- [44] J. Xu, M. Yagoub, R. Ding, and Q. J. Zhang, "Neural-based dynamic modeling of nonlinear microwave circuits," *IEEE Transactions on Microwave Theory and Techniques*, vol. 50, no. 12, pp. 2769-2780, Dec. 2002.
- [45] Z. A. Khan, C. F. Bunting, and M. D. Deshpande, "Shielding effectiveness of metallic enclosures at oblique arbitrary polarizations," *IEEE Transactions on Electromagnetic Compatibility*, vol. 47, no. 1, pp. 112-122, 2005.
- [46] X. Dong, et al., "Detection and Identification of Vehicles Based on Their Unintended Electromagnetics Emissions," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 4, pp. 752-759, Nov. 2006.
- [47] S. Koroglu, N. Umurkan, O. Kilic, and F. Attar, "An approach to the calculation of multilayer magnetic shielding using artificial neural network," *Simulation Modelling Practice and Theory*, vol. 17, pp. 1267-1275, May 2009.
- [48] K. Aunchaleevarapan, K. Paithoonwatanakij, W. Khan-Ngern, and S. Nitta, "Novel Method for Predicting PCB Configurations for Near Field and Far Field Radiated EMI Using a Neural Network," *IEICE Trans. Commun.*, vol. E86, no. B, pp. 1364-1376, Apr. 2003.
- [49] I. Chahine, M. Kadi, E. Gaboriaud, A. Louis, and B. Mazari, "Characterization and Modeling of the Susceptibility of Integrated Circuits to Conducted Electromagnetic Disturbances Up to 1 GHz," *IEEE Transactions on Electromagnetic Compatibility*, vol. 50, no. 2, pp. 285-293, May 2008.

- [50] P. Dangkham, P. Sujintanarat, S. Chaichana, K. Aunchaleevarapan, and P. Teekaput, "Recognition and Identification of Radiated EMI for Shielding Aperture using Neural Network," *PIERS Online*, vol. 3, no. 4, pp. 444-447, 2007.
- [51] Q. J. Zhang and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design- from theory to practice," *IEEE Transactions on Microwave Theory and Techniques*, vol. 51, no. 4, pp. 1339-1350, Apr. 2003.
- [52] J. E. Moody, "The Effective Number of Parameters: An Analysis of Generalization and Regularization in Nonlinear Learning Systems," in *Advances in Neural Information Processing Systems 4*. Palo Alto, USA, 1992, pp. 847-854.
- [53] Microsoft. (2007) Add, change, or remove error bars in a chart. Excel Help.
- [54] S. H. Suyu, et al., "Dissecting the Gravitational Lens II Precision Measurements for the Hubble Constant, Spatial Curvature, and the Dark Energy Equation of State," *The Astrophysical Journal*, vol. 711, no. 1, pp. 201-221, Mar. 2010.
- [55] H. A. Bethe, "Theory of Diffraction by Small Holes," *The Physical Review*, vol. 66, no. 8, pp. 163-182, Oct. 1944.
- [56] D. A. Hill, et al., "Aperture excitation of electrically large, lossy cavities," *IEEE Transactions on Electromagnetic Compatibility*, vol. 36, no. 3, pp. 169-178, 1994.
- [57] C. Lertsirimit, D. R. Jackson, D. R. Wilton, D. Erricolo, and D. H. Y. Yang, "EMC coupling to a circuit board from a wire penetrating a cavity aperture," *Proc. IEEE Int. Symp. on Antennas and Propagation*, vol. 1, pp. 375-378, Jun. 2004.
- [58] B. Mohajer-Iravani, S. Shahparnia, and O. M. Ramahi, "Coupling reduction in enclosures and cavities using electromagnetic band gap structures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 2, 2006.
- [59] B. M. Iravani, S. Shahparnia, and O. M. Ramahi, "Coupling Reduction in Enclosures and Cavities Using Electromagnetic Band Gap Structures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 48, no. 2, pp. 292-303, May 2006.
- [60] Laird Technologies. (2010, Jul.) Laird Technologies. [Online].
<http://www.lairdtech.com/Products/EMI-Solutions/EMI-Shielding-and-Gaskets/Board-Level-Shielding/>
- [61] Chomerics. (2009, Sep.) Chomerics Web site. [Online].
http://www.chomerics.com/products/documents/emicat/pg192theory_of_emi.pdf

- [62] C. A. Balanis, *Antenna Theory*, 3rd ed. Danvers, MA, USA: John Wiley & Sons, 2005.
- [63] J. F. Dawson, T. Konefal, A. C. Marvin, M. P. Robinson, and S. J. Porter, "A Fast Multiple Mode Intermediate Level Circuit Model for the Prediction of Shielding Effectiveness of a Rectangular Box Containing a Rectangular Aperture," *IEEE Transactions on Electromagnetic Compatibility*, vol. 47, no. 4, pp. 678-691, Nov. 2005.
- [64] S. Deng, T. Hubing, and D. Beetner, "Estimating maximum radiated emissions from printed circuit boards with an attached cable," *IEEE Transactions on Education*, vol. 50, no. 1, pp. 215-218, Feb. 2008.
- [65] C. A. Balanis, *Advanced Engineering Electromagnetics*, 1st ed. Hoboken, NJ, USA: John Wiley & Sons, 1989.
- [66] MathWorks, Matlab. (2010) Fzero. Help.
- [67] J. L. Drewniak and T. P. Van Doren, "Investigation of Fundamental Mechanisms of Common-Mode Radiation from Printed Circuit Boards with Attached Cables," in *Proceedings of the 1994 IEEE International Symposium on Electromagnetic Compatibility*, Chicago, 1994, pp. 110-115.
- [68] R. L. Hill, T. P. Van Doren, T. H. Hubing, H. J. Drewniak, and F. Gisn, "Common Mode Currents Induced on Wires Attached to Multilayer Printed Wire Boards with Segmented Ground Planes," in *Proceedings of the 1994 IEEE International Symposium on Electromagnetic Compatibility*, Chicago, 1994, pp. 116-120.
- [69] H. W. Shim and T. H. Hubing, "Model for Estimating Radiated Emissions From a Printed Circuit Board With Attached Cables Due to Voltage-Driven Sources," *IEEE Transactions on Electromagnetic Compatibility*, vol. 47, no. 4, pp. 899-907, Apr. 47.
- [70] Y. Fu and T. Hubing, "Analysis of Radiated Emissions From a Printed Circuit Board Using Expert System Algorithms," *IEEE Transactions on Electromagnetic Compatibility*, vol. 49, no. 1, pp. 68-75, Feb. 2007.

Appendix A

AutoIt!* and *NeuroModeler

AutoIt! is a program that was developed for Windows that allows a script to have the ability to control the computer's keyboard and mouse as well as read dialog boxes/labels/text on the user's computer. Utilizing this powerful scripting tool, a script was developed initially that created many ANNs in *NeuroModeler* (v1.5) through *AutoIt!*. This was accomplished by analyzing data presented on the screen and by using conditional statements: modified, trained, tested, and saved ANNs made in *NeuroModeler* according to user specifications. *AutoIt!* was also utilized in recording ANN training and calculation times as well as monitoring *NeuroModeler*'s memory usage.

Code used to make *NeuroModeler!* is available for download on OhioLINK supplemental section. The script file will require the installation of *AutoIt!* version three. In the script, lines 86-88 need to have the training/testing/ f_{ann} names array updated for the ANNs data sets that will be utilized. Also, the arrays need to have the dimension updated as well. After this information is entered, the script can be run by pressing F7. A future update is to further enhance the graphical user interface (GUI) such that the array update can be handled inside of the GUI. Other known issues with the script include early termination due to improperly arrayed variables caused by labels not being displayed when

they are expected to be displayed as well as *NeuroModeler* memory drift. After a long session of creating ANNs, the ANN memory usage as recorded from the Windows' task manager steadily increases, and this is assumed to be occurring due to a large open log file for *NeuroModeler* as well as memory allocation for the program never being reassigned. However, even after one full day of ANN creation, the memory usage of *NeuroModeler* is typically less than one hundred megabytes.

The advantages of *NeuroModeler!* include the following:

- Makes the creation of ANNs less arduous for the user.
 - Saves and stores all files in an intuitive directory scheme.
- Creates a comma-separated file of relevant results for up to 33 fields.
- Creates a highlight page of the results file.