Measuring Fault Resilience in Neural Networks

Joel Tobias Ausonio

A dissertation submitted to the
University of Reading for the
Degree of Doctor of Philosophy

Date of Submission: December, 2017

Declaration: I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Joel Tobias Ausonio

Abstract

In an extension to research into modeling a biological network of neurons this expands the basic characteristics of an Artificial Neural Network (ANN) computational model to measure functional compensation exhibited by a biological neural network during damage or loss of structure. Whilst current research has highlighted the availability of various technologies and methods relevant to this area of study, none provide a sufficient description as to how fault tolerance is measured nor how damage is evaluated. Such metrics must be consistent, reproducible, and applicable to a plethora of neural network architectures and techniques. Furthermore, measuring fault resilience of biologically inspired ANN architectures provides insight into how biological networks are able to exhibit this amazing ability.

This research brings together previous works into a comprehensive damage resilient ANN framework as well as, and more importantly, provides consistent measurement of fault tolerance within this framework. The proposed set of fault resilience metrics provides the means to evaluate the efficacy of networks which are subjectable to damage. These metrics and their source algorithms rely on the modification of various statistical methods and observations currently used for network training optimization.

Contents

List of Figures

List of Tables

List of Algorithms

Publications by the Authors of This Thesis

- Towards Optimizing the Selection of Neurons in Affordable Neural Networks, 2014 Ausonio et al. (2014)

Acknowledgements

This dissertation has been one of the most difficult and rewarding things I've done in my life to date. I made many sacrifices and experienced my fair share of moments when I thought I would never finish. This was a journey I did not take lightly and still underestimated. There are many people without which I would not have succeeded.

First and foremost I would like to thank my supervisors Professor Richard J. Mitchell and Professor William Holderbaum. Richard, your patient persistence and depth of knowledge kept me on track throughout the long process of a part-time PhD. I cannot thank you enough for knowing exactly when to reach out to me via email and check in on me. It is as though you knew exactly when life was getting in the way of progress and gave me just enough of a push to continue. I would not have made it through the most difficult parts of this programme without you.

William, your excitement about my subject matter along with the hours you spent on video calls with me trying to get me heading in the correct direction were priceless. Thank you for all of the hard work and late nights you put into ensuring my success. I will miss the times we would finish our technical discussions and just talk about life and family.

To Iain, my friend. Thank you for giving me a sounding board for bouncing ideas. Thank you for being supportive. Thank you for acting like I had already finished this programme and trying to relieve the stress I held onto while doing it. But, most of all, thank you for dropping off my dissertation for submission. You're a life saver.

To my parents Salvatore and Kathy, thank you letting me hide in your house

to work on this thesis, night after night. There were quite a few nights where I would get nothing done and just stare at the screen. I'm sure you knew. Thank you for everything. You are the best parents I could ever ask for.

To my twin brother, Matthew. We've both had a lot of changes in the past seven years but we were always able to stay connected. You let me tell you the hard parts of this programme but also made me remember the good parts. If I were ever to consider quitting I'm sure you would have stopped me. Thank you for listening.

To Aurea, my daughter. I love you. You are my sunshine. Daddy is done with his work now. We can make robots and play games all you want. Feel free to finish the dissertation you started while I worked on mine. If I remember correctly it had to do with micro controllers. I will help you in any way I can. I hope you can see what I've done and realize that you can do anything you put your mind to.

To Lucas, my son. I love you. You missed most of this time in our lives but you've joined us at the best part. When you're old enough I'll tell you all about this work I did and what it means to me.

Salena, my love. I would not have started nor finished this if it were not for you telling me time and again that I could do it. You never gave up on me. You always believed in me and never doubted me. For every hour of work I put into this programme you put two into making sure I could do so. Your strength carried me over the finish line. Thank you for being my best friend. I love you more than anything.

Chapter 1

Introduction

Considering the ability for a biological system to recover from faults, it is believed that studying and mimicking this quality in an Artificial Neural Network (ANN) would not only alleviate problems in distributed ANN applications but may also shed light on how a biological system provides this crucial ability. The ability to optimize biologically-inspired solutions is currently not possible due to a lack of consistent methods for appraising the efficacy of said solutions in the presence of damage.

Previous and current work in the area of self-healing ANNs focuses either on application-specific designs to overcome faults (e.g. algorithm specific improvements) or the introduction of a generic "watchdog" component responsible for actively monitoring, diagnosing, and reconfiguring the network (Jin, 2010). This active system monitor would still require prior knowledge of the problem domain and/or would typically become a bespoke solution for the network it is meant to diagnose. Neither are considered generic frameworks nor would they necessarily remain faithful to the original biological inspiration of the brain (Al-Zawi et al.,

2009).

Further, a common framework for providing damage resilience to ANNs must take into account saliency of the networks subcomponents (i.e. neurons) when attempting to recreate lost connections and preserve previously learned responses. Neural network pruning and optimization techniques are crucial in developing this framework (Silva et al., 2005)(Khabou and Gader, 2000)(Yuan et al., 2010). The focus of this research is to bring together various methodologies to provide a common framework for measuring fault recovery and resilience within a biologically inspired self-healing ANN architecture. Additionally, this framework will help to alleviate concerns and shortcomings in current self-healing ANN approaches. In providing a set of fault tolerance measurements, past and future research into fault resilience ANNs can be compared, leading to meaningful relative evaluations of efficacy. A metric-based comparison leads to the possibility of fault tolerant artificial neural networks being consistently optimized, resulting in improved future research in this field.

1.1 Biological Inspiration

The brain's ability to reorganize and reinforce its functions is paramount to its ability to produce learned responses. As the biological brain loses neuron cells and the connections they have made with surrounding cells (through either damage or decay) the inherent knowledge is ultimately lost. Yet, from a system perspective a brain can still produce similar output in spite of these changes within its foundation. Studies into brain neurogenesis, syntapogenesis, and sprouting show that new cells are being made and fresh connections

established between sensory neurons, motor neurons, and neurons within the hippocampus. The same neuroplasticity responsible for the ability to learn and develop is also the basis for damage recovery (Clergue and Collard, 1998)(Mulligan et al., 2010)(Michel and Collard, 1996)(Stroemer et al., 1995).

Similarly, when an ANN is damaged after training it can still produce some output (depending upon its design) but its behavior thereafter is not well understood nor guaranteed. Damage to an ANN in this case refers to either missing connections between input-hidden/hidden-output nodes or the absence of hidden nodes altogether. When considering a distributed ANN where each node may exist on separate hardware and connections between them cannot guarantee a quality-of-service this structural loss presents a real problem (Steinder and Sethi, 2004)(Tang et al., 2005)(Zadeh and Seyyedi, 2011)(Lee et al., 2011). This topological consideration is also prevalent within deep neural networks, or stacked neural networks, where each neuron in this context is a processing unit which can be lost. Currently, there is no common set of measurements available which aim to understand how fault resilient an ANN is, nor how well it may "recover" following either active or passive attempts at self-healing.

1.2 Fault Diagnosis Foundation

Starting with a review of basic, black box, system-level, fault diagnosis against various applications and ANN implementations, the goal is to aggregate commonalities into a fault diagnosis framework for use in this research. This thesis details the pros and cons associated with numerous fault diagnosis techniques and what they mean to this research by highlighting those which are considered

to be improbable within context of the biological inspiration mentioned earlier. In reviewing biological inspirations, and varying fault diagnosis implementations (both related to neural networks and not) a foundational set of features is collated as the basis for fault tolerant neural network research. This foundation includes a set of fault resilience measurements which, in reviewing existing fault tolerant neural network research, is absent from this field of study. By and large, the research reviewed as part of this thesis lacks any meaningful, quantifiable comparison of fault resilience amongst methods. In some cases, detection and localization of faults are completely omitted from research efforts (Jin and Cheng, 2011)(Jin, 2010).

From there, the Affordable Neural Network (AfNN) method (Uwate and Nishio, 2005) is framed around the ability for individual neurons to participate within ANN learning and generalization behaviors. AfNNs provide a structurally redundant and passively fault tolerant framework with learning rates comparable to traditional Multilayer Perceptrons (MLPs). Further, fault diagnosis methods are applied to consistently quantify the level of damage the system takes when damaged using the same error measurements captured during training as part of the back propogation of error. These methods provide the underpinning for a common set of fault resilience measurements and are tested in this thesis using a selection of ANN classification and regression problems. Comparisons between regular ANNs and ANNs utilizing the presented fault diagnosis methods is performed and the implications discussed.

Designed originally to emulate the firing patterns of the human central nervous system the AfNN technique provides a biologically inspired design for MLPs, and, as mentioned, a truly passive fault diagnosis framework to build from. The way they work is by, first, providing a pool of neurons in one or more hidden

layers. The size of the pool is selected at network design and is equal to or greater than the number of neurons needed for the chosen application. During training each neuron is individualy evaluated for inclusion into the processing of the networks' input; the result of said evaluation results in a subset of the neurons actually pariticpating in the processing. This evaluation essentially acts as a binary switch, one for each neuron, turning on or off that neurons' activity for a period of time. The method of evaluation is defined as the affordability scalar in section 3.1.

1.3 Use of Entropy and Epochs

The calculation of information entropy within the field of machine learning is mostly limited to providing a cost term for training optimization (Karystinos and Pados, 2000)(Silva et al., 2005)(Khabou and Gader, 2000)(Yuan et al., 2010). However, since this calculation is already being passed back through the network, in the case of back propogation based learning, it can readily be used to not only affect network weights but also to quantify the value of each unit within an ANN. Similar to how Mean-squared Error (MSE) is used, entropy can now be harnessed to evaluate the "importance" of individual neurons and, therefore, the impact of losing a neuron and its connections due to damage.

Measurement of epochs is a unique concept in that learning systems will attempt to avoid overfitting by limiting the number of epochs to train against in conjunction with either a target error or error difference thresholds (Haykin, 1994). What are not used thus far are the remaining epochs (i.e. the epochs not used) as a measurement of network training efficiency. For instance, if a network

is trained and damaged then retrained to the previously acheived MSE within an epoch ceiling then, holding all else constant, two network configurations can be compared with respect to how many epochs they used to acheive their targets as a means to understanding retained value of the network between damage onset and retraining.

1.4 Summary of Thesis Contributions

As will be detailed in the review of associated literature, no fault tolerant neural network study currently provides a multi-faceted approach to measuring fault resilience within an ANN framework. Furthermore, no fault tolerant neural network study meets the requirements of the foundational feature set for this area of study, as presented in this thesis. Specifically, this thesis proposes that no fault tolerant neural network research previously undertaken and reviewed herein provides a truly passive fault resilient method and no method sufficiently quantifies the level of fault a system incurs.

By using AfNNs (Uwate and Nishio, 2005) as the basis of comparison with respect to fault resilience measurements, the research presented in this thesis is consistent with the biological inspirations whilst providing a foundation for evaluating the resilience of this and similar frameworks in the future. Using error, entropy, and epoch based calculations the contributions of the research presented here is aimed at providing definitions for these fault resilience measurements and evaluating them against well known, publicly available data sets. Aggregating all of these elements, the following points summarize the contributions of this thesis:

1 Error-based resilience measurements have been produced and justified, which are designed around the concept of neuronal redundancy. This use of error-based measurement is novel because it is specific to how network error is affected by loss of structure.

2 Entropy-based measurements will be derived for further measuring saliency of redundant neuronal units.

3 Metrics for quantifying fault rehabilitation through the use of error, epochs, and entropy, within a number of control settings, will be provided, through experimental results.

4 Analysis on the effects that data set generation, and attribute distribution therein, have on the weight distributions within an MLP and, subsequently, the fault resilience measurements presented.

1.5 Organization of Thesis

Chapter two provides an in-depth review of relevant literature, framing this research by highlighting relevant areas of work by others and underlining current gaps therein. Chapter three reviews the AfNN technique and how it can be used as a basis for a structurally redundant and fault resilient ANN with which fault recovery measurements can be made. Chapter four provides error-based fault recovery measurements which focus on the inherent value of individual neurons and how the loss of these neurons affects the overall "health" of the ANN. Chapter five takes this further and produces two new measurements, based on calculations of entropy and epochs, which complement the findings

from chapter four. Chapter six discusses how data set make-up can affect not only the ability of an ANN to train but also to sustain damage. Finally, chapters six and seven provide conclusive statements and thoughts on future research, respectively.

Chapter 2

Fault Resilience in Neural Networks - Related Research

Fault resilience measurements of Artificial Neural Networks is an area of study which, as detailed in this chapter, is rather limited. However, each instance of previous research share common foundations in various subjects. Section 2.1 focuses on a method by (Uwate and Nishio, 2005) for providing structural redundancy to an Artificial Neural Network (ANN) called the Affordable Neural Network (AfNN). Next, and to better understand and frame the novel contributions of this thesis, section 2.2 presents foundational concepts related to the study of neural network fault resilience. Following that, section 2.3 details research specific to either methods for introducing fault tolerance within ANNs or the analysis of fault tolerance within ANNs. Finally, section 2.4 details the core hypotheses presented in this research as the basis for scientific experiments herein.

As foundational subjects are discussed the descriptions will highlight how and

why they are considered relevant to the area of fault tolerant ANN research. Subsequently, constraints and considerations are captured and examined. The analyses and descriptions of previous fault tolerant ANN implementations is reviewed and shortcomings noted in comparison to how they relate to the underlying subject matter.

This review of previously published literature will support and make evident the problem statement which is the subject of this thesis: namely, that there exists no set of measurements designed to capture the various facets of fault tolerance in ANNs. For instance, previous research lack a truly passive framework and, therefore, are both not biologically inspired and not computationally acceptable (i.e. attempting to scale these methods exceeds computational capacities). Alternatively, they make claims regarding natural fault tolerance of neural networks trained using modified data sets and utilizing various neural network architectures but are unable to provide a consistent set of metrics which quantify this resilience effectively. In addition, these studies also suffer from their own designs in that the fault resilience is solely dependent on the localization and rehabilitation of fault units which suffers from temporal delay in detection and action which are not acceptable in real-world applications.

The AfNN technique was intended to provide an Multilayer Perceptron (MLP) model with the ability to more closely mimic the neuronal firing patterns exhibited by a biological brain. In doing so, they indirectly provide a new foundation for structural redundancy which can be analysed within the area of fault tolerant ANN research. With both the AfNN method and the lessons learned from previous research regarding the measurement of faults within neural network systems, these tools provide the framework for solving the problem statement of this thesis.

2.1 Affordable Neural Networks as Foundation for Fault Resilience Measurements

Considering the various fault tolerant neural network approaches, and also the goals of this thesis to provide fault resilience measurements for MLP applications, the work by (Uwate and Nishio, 2005) provides a method with which a number of the foundational features, listed in section 2.2, of fault resilience are present. Using what they call AfNNs Uwate and Nishio are neuron pool selection to produce subsets of participating neurons, within the hidden layer, at each presentation of data to an MLP. The benefits of this approach include learning rates comparable to a classic MLP whilst providing structural redundancy by training duplicate, highly salient neurons within a biologically inspired design. Most importantly, this redundancy is purely passive in nature in that loss of neurons need not be detected for the network to continue operation; the lost units are simply not available for selection.

The AfNN method also helps in overcoming shortcomings noted in (Bugmann et al., 1992) and (Damarla and Bhagat, 1989) where large pools of neurons led to low convergence rates and poor accuracy. Affordability, as described by Uwate and Nishio, helps not only avoid this problem but also to reduce the symptoms of being stuck in local minima. Similarly, works by (Deodhare et al., 1998) and (Neti et al., 1990) note that distribution of weight saliency is key to fault tolerance; this distribution is a direct consequence of the affordability method, as neurons are selectively participating during each propagation of weight updates. Per (Chu and Wah, 1990), temporal delays in executing active fault resilence frameworks introduces failure in the active fault diagnosis frameworks. The

benefit of using Uwate and Nishio's model is that temporal delays are never incurred. A lost neuron is simply not available for selection during data presentation and back propagation. A diagram of the AfNN architecture is presented in figure 2.1

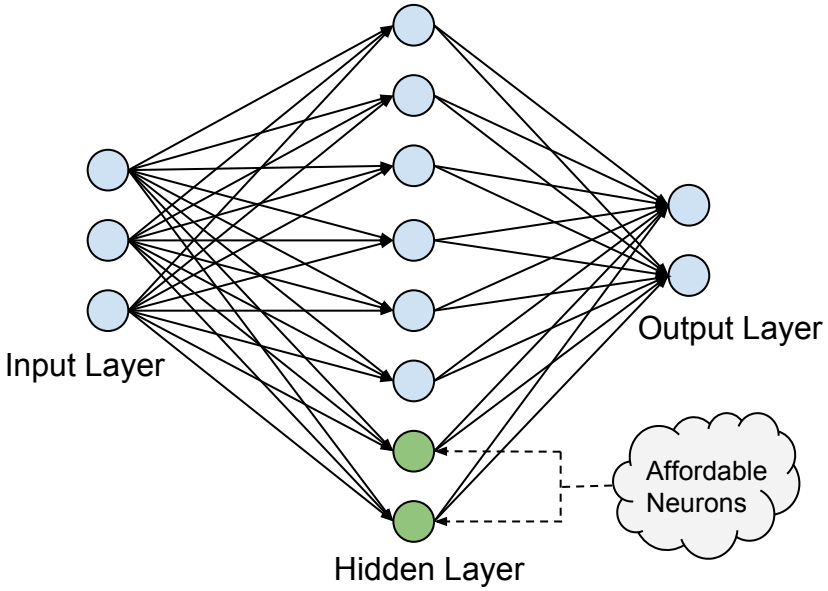


Figure 2.1: Network model with affordable neurons as proposed by (Uwate and Nishio, 2005).

Considering the numerous fault resilient neural network implementations reviewed in section 2.3, the affordable neural network technique provides a unique opportunity in that biological inspiration, and passive redundancy are all present. A set of fault resilience measurements built on top of and measured against the AfNN meets all of the principle features of a fault tolerant neural network as detailed in this thesis.

However, the study performed by (Uwate and Nishio, 2005) has a number of shortcomings that need to be addressed before proceeding with the AfNN as

a basis for fault resilience measurements. Namely, the study presented does not sufficiently describe the affordability methods capability to train nor how it works.

2.2 Foundational Research

This section captures the inspirations, problem statements, and basic methodologies related to the research of fault tolerant neural networks. First a summary regarding the importance of the biological inspirations and studies related to this research is presented. Brain damage studies, in particular, reinforce the foundation for neuroplasticity and neuronal reorganization. ANNs used in brain studies are presented in order to highlight existing fault resilience of standard implementations. It is also important to detail how first-order biological systems relate to high order, emergent, brain recovery. From this analysis the techniques and constraints of fault diagnosis approaches is revealed and made available for comparison later on. Lastly, a review of research related to data sets and network optimizations focusing of the efficacy within ANNs is discussed. These are often overlooked subjects within the fault tolerant ANN research gathered in section 2.3 which has real implications on the effectiveness and relevance of those measurements.

2.2.1 Brain Damage Studies

Work by V.S. Ramachandran presents the core inspiration for this research. In his book *Phantoms in the Brain* (Ramachandran and Blakeslee, 1998) he covers a plethora of brain damage studies and the resultant system-level side effects.

In the chapter about neglect Ramachandran mentions neurogenesis (growth of new/replacement neurons) and synaptogensis (neuronal connection reconfiguration and creation) as essential functions in brain damage recovery. These relationships (neurons and their connections) already exist in typical ANN implementations, making this research a logical progression in brain biomimicry. Use of ANNs in various fields and areas of application produces the need for robust and damage resilient ANN implementations.

To reinforce the comments made by Ramachandran, there exists evidence in other studies of neurogenesis in mouse and human brains (Rahman et al., 2002)(Mulligan et al., 2010). Also of relevance is evidence of natural neuronal decay. These abilities are noted as being crucial to the brain's ability to provide adaptation and consistent processing power in addition to brain damage recovery.

Considering these abilities it is not entirely surprising that research in natural brain damage recovery has led to studies in assisting people undergoing neuroanatomic healing. Andersen et. al. (Andersen et al., 2012) detail a number of activities in artificially accelerating natural brain damage recovery. This is important because it provides initial results regarding the ability to mimic high level brain damage recovery using software models.

2.2.1.1 Brain Behavior Emulation in Simulated Systems

Naturally, given these areas of research, and the implications they can have on ANN implementations and applications, use of ANNs in studies of the brain are of interest, along with a look into inherent fault resilience of some ANN architectures. As these studies aim to mimic brain damage or diagnose behaviors exhibited by damage sufferers they also describe ANNs as sharing the same side

effects of a compromised topology. These areas of research tie together the ideas of biological brain behaviors and the behaviors of simulated software and/or hardware implementations.

RajaRajan (RajaRajan, 2011) shows how brain damage can be diagnosed using neural networks. The important contribution from this study is the statement that, although computer-based neural network implementations are not nearly as complex as their biological counterparts, they still share some of the system-level emergent behaviors. In other words, fault resilience is a byproduct of the complex topologies of both systems (even if the artificial implementations are significantly less complex).

More importantly, according to RajaRajan, the complex topology and inherent fault resilience of neural networks are based on the overlapped nature of functional specialization. This means that structural redundancy, shared amongst multiple specializations, leads to inherent fault resilience (Hu and Hirasawa, 2000). However, if overlapped modalities and dynamic, redundant topologies are to be implemented then a new problem arises: how to manage this dynamic topology is by separating the expected topology from the neuronal units that are a part of it. This structural redundancy, as the basis for fault resilience, is also expected to aid highly distributed networks (Federici, 2005). These findings suggest that redundancy and the ability to manage it by growing, shrinking, and reorganizing its units results in a network which can potentially mold itself to various applications and overcome faults. The agent-based approach presented by (Federici, 2005) provides a means of implementing the self-management of each neuron within the context of neuronal selection. This very claim regarding utilization of an agent-based approach to fault rehabilitation forms the basis of most existing fault tolerant neural network implementations.

2.2.2 System-Level Fault Diagnosis

Consider a black-box system used to provide a basic input-output relationship. When the system is presented with a random sampling of input data from a known dictionary it will produce a deterministic output in response. This description is directly applicable to both software and hardware applications. Civil structures can also be subject to this analogy if, for instance, you consider weight loading an input and quantifiable structural resistance and integrity as expected outputs. Even a biological system, like the brain, can be described in this way. Now, consider that any one of these systems (based on its architecture, design, and implementation) is subject to internal faults which can alter its ability to consistently and reliably meet its original design goal(s).

The study of system fault diagnosis is centered on mitigating the risk associated with these potential faults through detection, triage, and rehabilitation. Further, these actions attempt to preserve the black-box methodology by abstracting the existence of this diagnosis away from those that would provide input or retrieve output from it. The following sections outline common fault diagnosis approaches against varying applications and differing designs for implementation. Applicable to this research is how they all lend to a common fault diagnosis framework and provide technical ingenuity related to our subject matter. These works also highlight caveats and considerations discovered during previous research and implementations. Afterwards we begin to cover works related to ANNs in union with fault diagnosis techniques. Finally, self healing as a subject matter is discussed which is based heavily in the fault diagnosis methodologies presented previously. Self healing, in itself, does not supply the solution to a biologically inspired, damage resilient ANN implementation but

does lead us towards more areas of study which are directly applicable.

2.2.2.1 Use of Statistical Methods in Fault Diagnosis

Detection of faults within a communications system can be highly contextually dependent. For instance, (Steinder and Sethi, 2004) implements Bayesian reasoning using belief networks to determine where and if, within a TCP stack, a fault may have occurred in a digital transmission. This determination is dependent on a prior knowledge of system fault causation and a cause-effect relationship dependency model. This technique focuses on the presentation of a symptom as how it relates to an event rather than how it relates to system state, which shows insufficient per (Steinder and Sethi, 2004). Some algorithms reviewed require all symptoms of an event to be understood while others do not. Though multiple algorithms are introduced (Steinder and Sethi, 2004)(Tang et al., 2005)(Nickelsen et al., 2009)(Zadeh and Seyyedi, 2011), the sequence of actions described for each is common:

Initialization Initialize symptom observations to "not observed"

Symptom Observation and Analysis For every symptom observed thereafter, analyze the symptom and relate the observation to all parent and child system components. This step utilizes a belief network designed using prior knowledge of the system under test.

Fault Selection and Localization Each system component calculates the probability that a particular fault event has occurred given the set of symptoms

The study by Steinder et. al. and others (Tang et al., 2005) (Nickelsen et al., 2009) (Zadeh and Seyyedi, 2011) are relevant because they highlight a common

method for fault diagnosis in engineering principles that is not necessarily biologically inspired given our current understanding of brain damage recovery mentioned earlier. In particular, it is important for this research to move away from a fault diagnosis approach which is reactionary or tries to categorize the overall system state as a means for diagnosis. Rather, to remain biologically inspired and mimic the elasticity of the brains functions an applicable fault diagnosis technique and the measurements of such a system must aim to provide emergent damage recovery from a lower level redundancy without using bespoke active recovery. Considering the overlapping of modalities onto one network by reusing subcomponents it becomes incredibly difficult to design a system-level fault diagnosis framework which aims to monitor, diagnose, and repair neurons while considering the large number of combinations in which each neuron can be used. Further, temporal delays in action using this paradigm suffers. If a fault diagnosis solution is meant to be generic and account for rehabilitation of neurons and their connections without knowledge of the functions they partake in (i.e. the problem space they are being applied in) then any attempt to consider system state and events as means for fault diagnosis would not be possible.

Taking this thread of subject matter further, work is being undertaken which aims to marry use of belief systems with a passive, structurally redundant, fault diagnosis (Tang et al., 2009). This is important for two reasons, as mentioned in the paper.

First, it highlights shortcomings with regard to an active fault diagnosis architecture (in this case, using a belief system to determine action). The most important shortcoming is the temporal correlation between current system state and previously observed symptoms. In a biological system, where an acceptable state is not necessarily represented by absolute accuracy or efficiency, the time

spent disambiguating noise from performance degradation due to structural loss attributes to a temporal delay between symptom detection and fault diagnosis action(s). Additionally, in an ANN this implies that the system has diverged from the original fault state through continuous data presentation and generalization. The implication is that, in the least, a short term state cache would need to be maintained and correlated to original symptom detection.

Second, the use of a belief system is highly dependent upon previously known faults and their symptoms during design/implementation. This is not possible in an ANN because the units which exhibit fault (neurons) do not contain context of the system state explicitly unless altered to do so (Bolt, 1992). The fact that two neurons with the same design and activation models can present the most and least salient elements in a trained neural network (Cun et al., 1990b) implies that a symptom-fault relationship at a neuronal level provides no credence to a system-level fault resilience.

2.2.3 Network Optimization and Neuronal Selection

Having reviewed a biological inspiration, covered both basic and specialized fault diagnosis methods the next step is providing these autonomic neural network units with the tools needed to produce emergent fault resilience.

This tool set consists of network pruning and optimization techniques and builds a foundation for fault resilient metrics. By using calculations, such as entropy, artificial neurons can be measured for their resilience against their peers and within the overall system. The content of these algorithms will be specific to this research but will build upon the successes of similar implementations by others.

Together, the aim is to produce a neuron architecture whereby, during participation in training and generalization of a larger ANN, these agents can manage their own paritipcation in such a generalization whilst the system is able to measure, in multiple ways, how fault resilient a network is and how much of that resilience depends upon a single neuron. The goal being that neuronal measurement of highly salient neurons will provide the most efficient measurement of fault resilience.

2.2.3.1 Network Optimization

Pruning a neural network is an optimization technique used for minimizing the size of a network whilst maintaining maximum performance. It prevents over-fitting of a network by removing redundant neurons or neurons which have essentially been trained against noise within training data. Amongst the various techniques for implementing pruning, use of Shannon's entropy is the best suited for the non-invasive fault resilience this research aims to provide. In other words, whilst other techniques leverage probing or otherwise perform calculations outside the normal operation of a neural network (typically with high temporal and processing costs), the calculation of entropy can be done passively using data already presented to the network units.

Karystinos and Pados (Karystinos and Pados, 2000) present a method for using entropy against the problem of ANN over-fitting. Specifically, methods for using input data sets as a means of breaking finite training data into pieces and using probability densities of those clusters. Whilst this work highlights the benefits of entropy as a tool for categorizing input data clusters it never really optimizes the network structure itself (particularly after it has been trained to a certain

degree). Karystinos and Pados also note that epochs utilized in training relates directly to the optimization of a neural network. In other words, treating epoch utilization as a maximization problem can also be used alongside entropy as a means to reach maximum efficiency of a network.

Silva et. al. (Silva et al., 2005) are able to use entropy as the neuronal cost function within an ANN but admit to the sensitivity of the algorithm's configuration in relation to its successful implementation. In this sense, entropy is no better than other cost functions when it comes to configuration and being contextually agnostic but does exceed in optimization accuracy.

Khabou and Gader (Khabou and Gader, 2000), and Yuan et. al. (Yuan et al., 2010) also use entropy as an efficient cost term in a side-by-side comparison with a "traditional" ANN approach using squared-error. Again, this highlights the power of entropy as means for cost evaluation over methods but which suffers from being a bespoke solution in real-world applications. In addition, Yuan et. al. used Kurtosis as another method for the cost term with limited success. Whilst this research was a less temporally heavy solution than that of Silva, it trades that off for less accuracy in optimization.

Having seen entropy as a cost term it is important to understand how this works. Opposed to a typical cost term (typically mean-squared error) entropy will effectively quantify cost as data compression efficiency. Considering an ANN as a data compression system which builds a deterministic relationship between an input dictionary and an output dictionary, entropy is the measure of uncertainty or randomness of the input pattern given the calculated output. Unfortunately, use of entropy also depends upon knowledge of the data density in order to be most effective (Cun et al., 1990b)(Silva et al., 2005).

It follows that this method need not be limited to high-level cost analysis. In fact, (Cun et al., 1990b) successfully use entropy as a means to calculate neuron saliency in order to prune an ANN to an optimal size. This method also places a heavy temporal and computational burden on the system as retraining needs to occur frequently.

Hassibi et. al. (Hassibi et al., 1993) and Zhao et. al. (Zhao et al., 2010) both build on the work by Le Cun to provide more efficient means of performing network pruning using entropy. Whilst these works omit specifics as to how well their respective algorithms perform and note that not all experiments are successful, this research still believes the use of entropy is an effective way to provide neurons the tools to perform fault resilience measurements.

Orlowska-Kowalska and Kaminski (Orlowska-Kowalska and Kaminski, 2009) also study use of entropy and saliency-based pruning methods. They are more interested in removal of weights as opposed to nodes. Effectively, pruning is the act of selectively damaging a neural network structure. The reason this work is important is that it highlights the tradeoff between the two previous approaches as being either temporally intensive or suffer from accuracy.

2.2.4 Training Set Generation and Effects on Fault Resilience

The last subject of literature review for the research presented here relates to data set generation for training and testing and how they may affect the fault resilience of a neural network. In other words, in the same way that a data set's generation may affect the ability for a network to learn against a specific classifi-

cation or regression task, the measurement of fault resilience is also expected to reflect this side effect and, therefore, fault resilience will suffer. Here we review some literature regarding uncommon methods for data set generation, per say, but specifically where network accuracy is affected.

The largest corollary to improperly created data sets comes in the form of imbalanced classes. As Liu (Liu, 2009) notes, this problem occured as far back as the late nineties and is now making a resurgence amongst data mining and machine learning applications. Liu goes on to describe that oversampling an undersampling of classes in a data set which, particularly in instances where incorrect sampling creates imbalances in the representation of features which are redundany or irrelevant, can dramatically skew classification. The solution presented by Liu is specific to data sets where class feature ratios are dramatically skewed and, in those cases, a mixture of bagging and interpolation is used to overcome the problem.

Santandar et. al. (Manoel Fernando Alonso Gadi and Mehnen, 2010) concur with statements made by Liu with respect to data skewedness being the core of the data set generation imbalance problem. However, in the instance where the number of classes is not "significantly" different across the data set then a simple technique of character extraction can be used to generate training and testing sets. Further, the methods for characteristic sampling can be done on a number of elements including feature selection, variable selection, feature reduction, and attribute selection. The use of a particular method comes down to the application and data set under investigation.

Finally, Bujang et. al. (Bujang et al., 2012) provide a description of systematic sampling of data related to clinical studies and how the sampling technique provided improved performance towards their problem statement. Comparing this

improved sampling method against a traditional one, based on a sampling frame, helps to augment a situation where such a sampling technique is insufficient. Given the various general techniques and the possibility for bespoke methods, depending on the study being performed, the research presented here will revisit the concept of data sampling after initial fault resilience measurements are made to not only validify the measurements but also to improve resilience where applicable.

2.2.5 Summary of Foundational Research as a Basis of Comparison

The preceding sections regarding the foundational research related to this thesis are summarized below. The purpose of this summary is to outline a basis of comparison for existing fault tolerant neural network research and, specifically, how well they appropriate the various features related to this area of study.

Biological Inspiration Per the research presented with respect to brain damage studies and the emulations therein, there exist a number of features important to a simulated environment remaining biologically inspired and, therefore, likely to account for biological system behaviors. Neurogenesis and synaptogenesis are important to this feature in that they describe the modular and loosely-coupled nature of the lowest level units in the brain (Ramachandran and Blakeslee, 1998). Neuronal decay, as well as neuronal generation, exhibit no obvious signs of specialized systems responsible for rehabilitation (Rahman et al., 2002)(Mulligan et al., 2010).

Fault Diagnosis Implementation Whether a synthesized environment is ac-

tive or passive in its implementation of fault diagnosis affects certain aspects therein. Active diagnosis entails a temporal pause in operation for rehabilitation, as well as the need for specialized components to perform said rehabilitation. Passive implies no such restriction but carries with it the need for each redundant unit itself to carry some sort of specialization. Both of these approaches have implications on both the biological inspiration as well as how fault resilience is measured (Shen et al., 2011)(Dalmi et al., 1998). Given what has been reviewed thus far regarding biological inspiration, a passive implementation of fault diagnosis is crucial for any eventual design of a fault tolerant neural network.

Fault Resilience Measurements As the penultimate focus of this thesis highlighting the shortcomings in fault resilience measurements in previous fault tolerant neural network research is paramount. In that regard, and from the perspective of existing research, the direct result of how the previous foundational features are adhered to is to be discussed in section 2.3.

Error: Measurements of network accuracy directly is subjective to the data, training, and NN method. But error is not importance (Cun et al., 1990b)

Entropy: A better test for saliency. Some methods mentioned later say that removing the most important neuron doesn't have the largest effect. This is directly due to importance being measured by error and not entropy

Epochs: Most methods disregard epochs as a measurement of resilience. But from the perspective of rehabilitation, the number of epochs utilized is telling as to how damaged something is/was. (Karystinos and Pados, 2000)

2.3 Fault Tolerant Neural Networks

Work by (Al-Zawi et al., 2009) provides a basic self-healing neural network framework which is very similar to a basic fault diagnosis architecture mentioned earlier. This system, like those used in common fault diagnosis implementations, uses a separate system to collect fault symptoms and effect changes to the "running system". Important in this research are the two approaches in self management of systems; integrated, expert rules are used to manage the system (this presents the contextual, bespoke portion of the framework) and the generic, adaptive learning part of the system. This provides a nice encapsulation and separation of those parts of a self healing system which can be applied to multiple solutions.

Jin (Jin and Cheng, 2011)(Jin, 2010) has already attempted to implement a self healing ANN. The outcome is proposed to be autonomously reconfigurable but Jin concedes that damage detection is not within scope of their studies. In the event of failure, the faulty neuron is detected and replaced. However, the detection and replacement methods are not inspired by a biological system. Whilst this research highlights the needs for fault resilience in ANN it does not provide an extended analogy to the brains natural fault resilience in the same fashion that the perceptron mimicked the neuron.

Chen et. al. (Chen et al., 1992) have generated an even more tailored, engineering-based solution. Using methods like checksums and error correcting bit patterns, this work presents a great solution to the problem posed but is by no means a

generic solution for ANN self healing. Unfortunately, this implementation moves further away from the foundational principles described in section 2.2.

A successful emulation of Alzheimer's disease was undertaken by (Hamilton and Micheli-Tzanakou, 1997) through first training an MLP with groups of classification data sets. Then, damage was emulated by randomizing weights between neurons in the various layers along with "blurring" of input data. In the most extreme cases neurons were removed entirely from the network. Whilst the study results were aimed towards the implications of the presentation and onset of Alzheimer's disease it also implies that damaged ANNs behave similarly to biological networks (Hamilton and Micheli-Tzanakou, 1997) in relation to generalization accuracy post damage. However, no consistent metric was provided to explicitly quantify damage sustained and how that relates to a brain.

Segee and Carter explore the "myth" of inherent fault tolerance of parallel distributed processing networks (Segee and Carter, 1994). In this article, Radial Basis Function (RBF) networks and MLP networks have their inherent fault resilience compared by utilizing a measurement of Root Mean Squared (RMS) error. As use of RMS is a common theme for fault tolerance measurement in follow-on studies, the definition is presented below.

$$RMSError = \sqrt{(\frac{1}{N})\sum_{i=1}^{N}(F(x) - F(y))^2} \tag{2.1}$$

where N represents the number of data vectors in one epoch, $F(x)$ is the network approximation of the expected output $F(y)$. Part of the discussion presented by Segee and Carter highlights the importance of fault injection into a network to improve tolerance, as proposed by (Carlo H. Seǵuin, 1990).

Seǵuin and Clay, in the area of fault tolerant neural network research, present the

most complete and comprehensive analysis amongst the research gathered in this thesis. Published in 1990, this paper focuses on redundancy being paramount to fault tolerance. The study is limited to stuck-at-faults, another case of having the faulty neuron for analysis, as opposed to neurons which are completely lost (no longer connected to the network and not participating in production of network output) when faulty (more akin to a biological system (Rahman et al., 2002)). Fault tolerance, in this research, is analagous to "noise immunity" in that overcoming faults can be acheived by training against "noise" designed to emulate stuck-at-zero and stuck-at-one faults. On top of this level of tolerance, a faulty unit competition takes place in the case of emulating what Seǵuin and Clay refer to as analog data sets (non-binary expected outcomes). The size of complexity of the units under fault, which utilize a monitored replacement mechanism for rehabilitation, are larger than what is considered ideal by the authors. However, increasing this complexity and, therefore, reducing the footprint of each unit was deemed too difficult. The measurement of fault resilience in this study is limited in two ways: first, from a system perspective, faults are limited to those within the hidden layer. This sets the standard by which most subsequent research pertains since faults in either input or output units constitute different problem statements; particularly from a biological standpoint (akin to losing the sense of touch or the ability to effect a motor response). Second, fault tolerance measurement is primarily focused on the time it takes to train a network with noise induction, rather than introducing faults post-training and rehabilitating. Finally, in order to provide a limited set of recovery mechanisms, Seǵuin and Clay introduce replaceable units which rely on monitoring and temporal redundance techniques (namely, sub-unit competition using error measures).

Chu and Wah (Chu and Wah, 1990), the primary predecessor to the work by

Seq́uin and Clay, poses and analyzes a set of methods to introduce both spatial redundancy and temporal redundancy. This research by Chu and Wah, however, also attempts to account for output layer errors. In their approach, every unit output is computed multiple times, each by different neurons, for competition, which leads to a tremendous temporal overhead. This approach is both not biologically inspired, nor does it follow the passive fault diagnosis principles expected of a structurally, or spatially, redundancy network, as noted in section 2.2. The outcome of this work, again, reinforces that the temporal delay introduced when executing an active self healing subsystem tends toward being a bespoke engineering solution. However, this work also highlights that in a structurally redundant ANN training and generalization take that much longer because inputs need to be presented to multiple copies of the same neuron and its connections.

Work by (Bolt, 1992) and (George Bolt, 1992) focuses only on the fault tolerance of MLPs. In these papers, it is determined that neuronal replacement and rehabilitation offer more than just fault injection. Throughout the study a number of key observations are made. First, the construction of data sets has a significant impact on fault tolerance and general trainability. Second, and related to the first, removal of neurons entirely negatively effects the ability to rehabilitate with an active diagnosis system like those proposed by (Bolt, 1992)(George Bolt, 1992) and (Chu and Wah, 1990). The second conclusion reinforces the findings collated in section 2.2; namely; true fault resilience at a neuronal level must emulate neuronal decay and genesis, not replacement using active monitoring techniques. The fault tolerance measurements introduced in this study are also solely error based, despite discussion about data set effectiveness (which leads to epochs used in training). As a result, Bolt notes that even removing the

seemingly most important units, measured using back-propagation of error, did not result in sudden drops of output accuracy of the network. Per the review of works related to entropy in section 2.2 (Silva et al., 2005)(Cun et al., 1990b) this is due to the incongruency between error and saliency.

Bugmann et. al. (Bugmann et al., 1992) presents fault resilience of a neural network as a minimization/maximization of error. According to Bugmann the effectivenes of one neuron can be measured if it is detected as being damaged in some way (stuck-at type fault). However, removing this neuron causes the MLP under test to get stuck in a local minima and unable to train. Bugmann also goes on to try and replace faulty units once detected. The intent for replacement is not to emulate neurogenesis but, rather, because adding too many units as a means to provide reundancy also led the MLP into a state where it would not converge. This study is yet another example of a fault tolerant neural network design which relies on the detection of damaged neurons (in this instance, requiring them to still be present) and only measuring fault resilience using network error.

Damarla and Bhagat (Damarla and Bhagat, 1989) make a case that the number of connections between the layers of an MLP is important to comparable fault tolerance. The results of this study also highlight, similar to Bugmann et. al., that too many units being trained results in decreasing accuracy of the network. Another important finding in this study is related to comparability of fault tolerant neural networks. This statement, in particular, will influence the selection of data sets in this thesis.

Ahmadi et. al. (Ahmadi et al., 2009) provide a clear description of what they believe to be three necessary features of a fault tolerant neural network. These include fault detection, fault localization, and fault correction. The last of which

is another active unit replacement method using like-for-like comparision of neuronal error. These methods, per the foundational research presented in section 2.2 is a typical active fault diagnosis approach which is neither biologically inspired nor computionally acceptable. In fact, (Bettola and Piuri, 1998), a predecessor study in relation to Ahmadi et. al., have already concluded that active diagnosis results in a temporal delay in detection and action, directly contradicting the three tenets provided.

Ito and Yagi (Ito and Yagi, 1994) use error correcting codes to determine output layer fault nerons. This is a unique example in that it focuses on the output layer of neurons only and uses error correcting codes to both detect and localize faults. Once again, fault tolerance is measured using error at the output layer only. Phatak et. al. (Phatak and Koren, 1992) (Phatak and Tchernev, 2002) (Phatak, 1999) also focus on the optimization of both weights and error within an ANN as a means for providing fault tolerance to an MLP. The latter of which attempts to mitigate the risk of error in applying the Vapnik-Chervonenkis theory. All-in-all these studies attempt to introduce more and more complex calculations for minimization of error in neural network architectures with no redundancy and no simulation of neuron loss akin to what is seen in other studies presented in this thesis.

Hsu et. al. (Hsu et al., 1995) present a method for executing fault tolerance through parallel active diagnosis and rehabilitation of neurons. The structure of these neuronal units mimic those within the study by (Chu and Wah, 1990) in creating what is called a "duplication with comparison" model. An active monitoring component is meant to replace faulty units using comparison of error output. Once again, fault tolerance measurement is limited to error calculations and goes so far as to analyse how error is distributed across neurons in the hid-

den layer.

Deodhare et. al. (Deodhare et al., 1998) and Neti et. al. (Neti et al., 1990) also both treat fault tolerance as a minimization problem. The unique contribution by these two studies has to do with how they considered the weight and error distributions within the hidden layer to be directly correlated to the presence of fault resilience. Whilst the ANN architectures provided do not focus on neuronal redundancy so much as training to avoid catastrophic failure from loss of a neuron, as measured using error, the statements made regarding the need to evenly distribute hidden neuron saliency relates to the extended set of measurements captured in section 2.2 (namely, entropy).

2.3.1 Tabular Comparison of Fault Tolerant Studies Against Foundational Research Features

In order to more readily deduce the gaps in current literature regarding fault tolerant neural networks table 2.3.1 is presented. The four principle features discussed in the foundational research section 2.2 are listed against each fault tolerant study presented in this thesis.

Study	Biologically Inspiration	Fault Diagnosis Implementation	Fault Resilience Measurements
Al-Zawi et al. (2009)	FALSE	ACTIVE	NONE
Jin and Cheng (2011) Jin (2010)	FALSE	ACTIVE	ERROR ONLY
Chen et al. (1992)	FALSE	ACTIVE	NONE
Hamilton and Micheli-Tzanakou (1997)	TRUE	NONE	NONE
Segee and Carter (1994)	FALSE	ACTIVE	ERROR ONLY
Carlo H. Seǫuin (1990)	FALSE	ACTIVE	ERROR ONLY
Chu and Wah (1990)	FALSE	ACTIVE	ERROR ONLY
Bolt (1992) George Bolt (1992)	TRUE	ACTIVE	ERROR ONLY
Bugmann et al. (1992)	FALSE	ACTIVE	ERROR ONLY
Damarla & Bhagat 1989	FALSE	NONE	ERROR ONLY
Ahmadi et al. (2009)	FALSE	ACTIVE	ERROR ONLY
Bettola and Piuri (1998)	FALSE	ACTIVE	ERROR ONLY
Ito and Yagi (1994)	FALSE	NONE	ERROR ONLY
Phatak & Koren (1992)	FALSE	NONE	ERROR ONLY
Phatak & Tchernev (2002)	FALSE	NONE	ERROR ONLY
Phatak (1999)	FALSE	NONE	ERROR ONLY
Hsu et al. (1995)	FALSE	ACTIVE	ERROR ONLY
Deodhare et al. (1998)	FALSE	NONE	ERROR ONLY
Neti et al. (1990)	FALSE	NONE	ERROR ONLY

Table 2.1: Comparison of fault tolerance neural network methods and foundational research areas.

2.4 Research Hypotheses

The following hypotheses are presented in order to frame the experiements undertaken as part of this research.

Hypothesis 1 *Networks which utilize the affordability method will*

exhibit a smaller total average Mean-squared Error (MSE) as levels of retraining increases.

The purpose of hypothesis 1 is to capture the most common definition of fault resilience (Jin and Cheng, 2011)(Jin, 2010). Namely, damage of a resilient network will spare some or all of it's previous value or function. Naturally, the expectation is that, if neuronal redundancy is to provide fault resilience (and, therefore, presentation of value) then the measurement of fault resilience should exhibit this. The next hypothesis provides another perspective to this same concept. Hypothesis 2 captures the expectation that fault resilience and preservation of function is akin to minimizing changes to network error as neurons are removed.

Hypothesis 2 *The more retraining that occurs between onsets of damage the lower the average error lost per neuron.*

The use of AfNN as the basis of neural network design within the experiments presented in this thesis leads to another hypothesis which relates the use of affordability to a network that does not. The basis of measuring the value of affordability related to this hypothesis is the subject of the experiment in section 4.3.

Hypothesis 3 *Networks which utilize an affordability method will achieve a smaller average error lost per neuron through affordability and will therefore provide more added value than an MLP which does not provide affordability.*

The last two hypotheses stem from the same concept of the number of retraining epochs effects on rehabilitation. They stem from the research by (Al-Zawi et al., 2009), (Jin and Cheng, 2011), and (Jin, 2010) which all discuss the effect of

retraining epochs on neural network recuperation post-damage.

> **Hypothesis 4** *The higher the retraining epoch ceiling the more often the network will retrain to pre-damage levels in between onsets of damage within the epoch ceiling.*

> **Hypothesis 5** *The expected positive effects related to hypothesis 4 are greater for an AfNN compared to the epochs needed for statically structured MLPs.*

Hypothesis 5 is similar to hypothesis 1 in that it relates the measures the effect of affordability on fault resilience through comparison with MLPs that do not employ affordability.

The hypotheses presented above which do not relate directly to AfNNs, namely hypotheses 2 and 4, capture the expectations of fault resilience MLPs independent of whether affordability is utilized. This is evidenced by the work in section 2.2. The concept of retraining MLP implementations is a previously used method of fault recovery (Al-Zawi et al., 2009) (Jin and Cheng, 2011) (Jin, 2010). Hypotheses 1, 3, and 5 relate the MLP design presented by (Uwate and Nishio, 2005) to the concept of damage recuperation through retraining of redundant neurons.

2.5 Summary

A diverse set of studies make up what is considered in this thesis as the foundational features of fault tolerant neural network research. Biological inspiration, a qualification as to how analagous a network is to a biological network, is crucial

with respect to how an ANN handles concepts such as neurogenesis, synaptogenesis, and neuronal decay. The fault diagnosis design employed by fault tolerant neural networks falls into either active or passive methods, the former of which immediately denounces the possibility of remaining biologically inspiration. Finally, all studies reviewed utilize fault resilience measurements based only on error captured as part of typical supervised learning algorithms. Use of entropy as a measurement of saliency is a natural and necessary addition to improve understanding of the effects of faults on neural networks. Similarly, measuring epochs needed to recover from faults as well as understanding how data sets affect fault resilience measurements are paramount.

Of all of the fault tolerant research reviewed in this chapter, only one (Bolt, 1992)(George Bolt, 1992) succeeded in meeting at least two of the four core features of fault resilient neural network research as described within this thesis. None of which provide a framework upon which varying fault resilience measurements can be meaningfully applied. Interestingly, the AfNN method, employed by (Uwate and Nishio, 2005) provides a unique and feature-ready architecture with which fault resilience measurements can be made and evaluated, despite being designed for a completely different purpose. Before executing such experiments, however, an evaluation of the AfNN technique must be performed to better understand the inherent learning characteristics it provides.

Chapter 3

Affordable Neural Networks

As mentioned in chapter 1, the brain's ability to reorganize and reinforce its functions is paramount to its ability in producing learned responses. In spite of the biological brain losing neuron cells, and the relationships they have made with surrounding cells, it can still produce acceptable output. Studies into brain neurogenesis, syntapogenesis, and sprouting show that new cells are being made and fresh connections established between sensory neurons, motor neurons, and neurons within the hippocampus. This neuroplasticity is the foundation for producing learned responses in the presence of a highly dynamic and structurally redundant network (Clergue and Collard, 1998) (Mulligan et al., 2010) (Michel and Collard, 1996) (Stroemer et al., 1995).

The highly dynamic neural network exhibited by the human brain presents a stark contrast to that of the structurally and functionally static designs of Artificial Neural Network (ANN) applications used in maths and engineering. It is worth considering whether mimicking this structural redundancy in ANN applications can provide any benefit. **Structural redundancy** within this study

is defined as the duplication of highly salient neurons which, in the event of neuron loss, preserve network performance.

Various works by (Uwate and Nishio, 2005) (Uwate et al., 2007) (Uwate and Nishio, 2010) detail numerous methods for the implementation of, what they call, an **Affordable Neural Network**, which we will redefine. The inspiration behind the Affordable Neural Network (AfNN) is based on providing an analogy to this neuron redundancy by detailing a mechanism which can manage a surplus of hidden layer neurons during training and production of learned responses. Using a Backpropagation (BP)-based Multilayer Perceptron (MLP) with more than the optimal number of neurons in a hidden layer, Uwate and Nishio are able to show that convergence is not only possible but, in some cases, improved when compared with a classic, structurally static, approach (although, the evidence of such claims are not clearly presented) as opposed to the AfNN approach which represents a dynamically structured approach. The efficiency and benefits of this method are highly dependent upon the method by which an optimal number of neurons are selected from the hidden layer but Uwate and Nishio detail the use of chaotic oscillation as the best way of achieving affordability selection (Uwate and Nishio, 2005).

The AfNN method provides this research with a candidate for a comparison with respect to measuring fault resilience through its inherent structural redundancy. In other words, comparing a regular MLP against various AfNN configurations is imperative to producing and validating a set of measurements which quantify fault resilience in ANNs.

3.1 Defining Semantics

To aid in understanding the aims of this portion of our research, and to allow a common semantic for referring to methods employed, the following definitions are created and presented here and further described in subsequent sections:

Affordablity Refers to the ability of an AfNN in providing a pool of neurons, a subset of which can be utilized at a given point in time for generation of network output. This is typically within reference to the hidden layer(s) within an AfNN whereby the number of neurons provided is smaller than the total number of neurons in the layer.

Affordability Method The method by which neurons are selected in order to provide affordability.

Affordability Total This term refers to a value representing the total number of neurons used by the affordability method which are readily available for selection (as opposed to neurons which are damaged or otherwise unusable).

Affordability Target This is the number of neurons which, if available, are utilized at any given time within an AfNN. As mentioned earlier, this is typically less than the affordability total. If the affordability total is less than or equal to the affordability target then the network will utilize zero-affordability. Only when the total is great than the target does affordability differ from a regular MLP.

Zero-Affordability This refers to a specific affordability method whereby *all* neurons are utilized at any given time. This can be viewed

as a typical MLP configuration. The term "zero" refers to the difference between affordability target and total (i.e. zero-affordability is when they are equal).

Affordable Neural Network An Artificial Neural Network which utilizes an affordability method. For the purposes of our research, this is typically an MLP.

Affordability Threshold During damage and/or loss of neurons, the affordability threshold represents the point at which the Affordability Target equals the amount of neurons available for affordability selection.

The studies carried out by Uwate and Nishio do not provide sufficient analysis explaining why their preferred method works. Nor do they provide evidence that they have achieved the most optimal solution to affordable neuron selection. Prior to creating a set of measurements to quantify structural redundancy of the AfNN method, this research aims to provide such explanations and a detailed analysis against alternative affordability methods.

Building upon these earlier studies, Uwate and Nishio make claims regarding the AfNNs ability to sustain damage (i.e. loss of structure) due to the duplication of weight values within the input-to-hidden and hidden-to-output connections (Uwate and Nishio, 2010). This misconception regarding weight magnitude equaling saliency is the subject of LeCun's paper on "Optimal Brain Damage" (Cun et al., 1990a) which presents a method for measuring the actual saliency of a neurons connections using the second derivative of the objective function. A modification of this measurement is presented below as the means for not only measuring structural redundancy within an AfNN but also to compare variations of the affordability methods.

This chapter revisits the findings by Uwate and Nishio and provides a more in-depth analysis towards, and quantifiable comparison of, optimizing affordable neuron selection against a defined measurement. This measurement is an extension on the saliency calculation provided by LeCun et. al and is the first contribution of the research presented here. Further, preliminary analysis of the various affordability methods is produced as a foundation for further chapters in so far as providing a set of networks against which structural redundancy measurements can be made and evaluated.

3.2 Detailed Definition of Affordability

This study (re)defines the AfNN as a feed-forward MLP utilizing an affordability method within its hidden layer. Knowing the number of optimal neurons (i.e. selecting an affordability target) for a given set of data and network configuration is circumstantial and this research will not provide an optimization method therein. In lieu of this omission, this research will either provide reference to literature detailing such a configuration or provide reasoning why a number of neurons is chosen.

The affordability total is determined during network design but the ratio of total neurons to selectable neurons (i.e. the affordability total vs. the affordability target) is dependent upon the training data and is determined through trial and error. During BP training a subset of the total neurons are selected for participation whilst the rest remain non-contributory. This non-participation is preserved when computing errors, updating weight values and computing neuron saliency. The affordability target, by extension, is also a parameter to the system defined

during network design and is equal to the optimal number of neurons in the hidden layer for a classic MLP approach.

Figure 3.1 illustrates a notional construction of an AfNN. In this diagram, the hidden layer is designated l, where m_l is the affordability target in hidden layer l and is a *constant value*, t_l represents the affordability total in layer l. Now we define the following

$$t_l = m_l + r_l \tag{3.1}$$

where r_l is the difference between the affordability total and the affordability target.

At network design, the value of r_l must be greater than zero in order for affordablity to exist within the network, per the definitions in section 3.1. However, as the network sustains damage, the value of r_l decreases. If the value of r_l is less than or equal to zero then affordability is lost (i.e. the network layer is now classified as having zero-affordability).

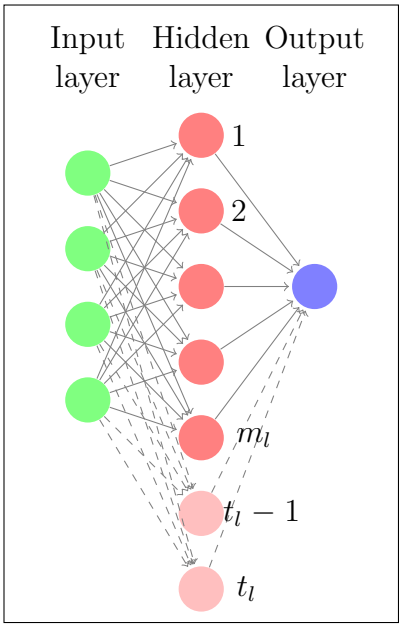


Figure 3.1: Neural network with affordable neurons, modified from (Uwate and Nishio, 2005).

All neurons share the same local induction and activation functions and utilize a back-propagation learning method. Also, early stopping is utilized for all experiments, both during initial network training and subsequent retraining. The target level of error for early stopping is dependent upon the error rate achieved upon initial training which is, in turn, dependent upon the data sets themselves. Considering a training set $\mathbb{N}$, where the training vector $n \in \mathbb{N}$, the weighted sum for neuron j within layer l, is defined as $v_j(n)$ and its corresponding activation is denoted φ (Haykin, 1994). Using this activation, the output of neuron j (denoted $y_j(n)$) is as shown in equation 3.2.

$$y_j(n) = \varphi(v_j(n))\psi_j(n) \tag{3.2}$$

where $0 \leq \psi_j(n) \leq 1$ represents the **affordability scalar**, or a value denoting the participation switch for neuron j at presentation of training vector n. For

each affordability method presented in this chapter (i.e. for each variation of ψ provided) each will return either 1 if the neuron is to participate or 0 if not.
Similarly, when performing weight updates during the feedback process, the change in weight value between neuron i within layer $l-1$ and neuron j within layer l (denoted $w_{ij}(n)$) is defined as follows (Haykin, 1994)

$$\Delta w_{ij}(n) = \eta \delta_j(n) y_j(n) \tag{3.3}$$

where η is the learning rate and δ represents the error signal for neuron j.
It should be noted that the inclusion of the affordability scalar, ψ_j, into the weight update is implied through the reuse of $y_j(n)$. Therefore, equation 3.2 is the primary focus for implementing the AfNN algorithm and will be shown with variance. Specifically, each of the affordability methods compared in this study will each provide a unique definition of ψ_j.
The following sections define the various neuron affordability methods to be studied. In particular, this study will revisit the random and chaotic selection methods mentioned by (Uwate and Nishio, 2005) but also provide a second chaotic map, and a new cyclic selection rule, for comparison. Following that, the learning results of each variant will be compared using a measurement of neuron saliency to highlight the relative benefits of each method against the goals set forth earlier. It is worth noting that equations 3.4 through 3.12, as written here, are unique contributions of this study.

3.2.1 Random Affordability

Following on from work by Uwate and Nishio, the comparison against a randomly selected affordability model is meant to provide comparison to, and credence towards, the use of a chaotic selector. In contract from Uwate and Nishio, this research provides an explicit description of how random affordability is performed. We define $\psi_j^R(n)$ as the random affordability variant of $\psi_j(n)$ as follows

$$\psi_j^R(n) = \begin{cases} 1 \text{ if } r_j(n) \geq u_l(n) \\ 0 \text{ otherwise} \end{cases} \tag{3.4}$$

where $r_j(n)$ represents a random value associated with neuron j during the generalization of input n, $u_l(n)$ represents the m^{th} largest affordability value in layer l (calculated for each neuron using affordable variants for $\psi_j(n)$), and m_l denotes the optimal number of neurons in layer l as defined in section 3.2.

The random number generator used for this study is provided by Boost libraries (Rivera and Dawes, 2014). The seeding of the generator is done at system startup and this one seed is shared amongst all neurons.

An example of random affordability is as follows: Given an AfNN with one hidden layer containing four neurons with an affordability target t_l equal to two neurons, then each time data is passed to this layer two neurons will be selected to participate while the other two will be non-contributory. This works by assigning a random value to each of the four neurons and choosing the two neurons with the largest values of the four. This is repeated each time data passes forward through this layer.

3.2.2 Chaotic Affordability

The chaotic AfNN implementation specified in (Uwate and Nishio, 2005) is described as using a skew tent map associated with each hidden layer neuron where all are given "different initial values" and are updated at every learning. This study utilizes more descriptive methods for initializing each chaotic value and incrementing of the chaotic values. We define $\psi_j^T(n)$, the tent map affordability variant of $\psi_j(n)$, as follows

$$\psi_j^T(n) = \begin{cases} 1 \text{ if } \tau_j(n) \geq u_l(n) \\ \\ 0 \text{ otherwise} \end{cases} \tag{3.5}$$

where $\tau_j(n)$ is the chaotic value associated with neuron j during generalization of training vector n, defined below. $u_l(n)$ is carried forward from the previous section 3.2.1 using the affordability variant $\psi_j^T(n)$.

$$\tau_j(n) = \begin{cases} \frac{2\tau_j(n-1)+1-\alpha}{1+\alpha}, & \text{if } \tau_j(n-1) \geq -1 \text{ and } \tau_j(n-1) \leq \alpha \\ \frac{-2\tau_j(n-1)+1+\alpha}{1-\alpha}, & \text{if } \tau_j(n-1) > \alpha \text{ and } \tau_j(n-1) \leq 1 \end{cases} \tag{3.6}$$

The initial value of the selection criteria for each neuron at presentation of the first training vector (n_0) is randomly initialized between 0 and 1 as follows

$$\tau_j(n_0) = rand(-1...1) \text{ where } \tau_j(n_0) \neq \alpha \tag{3.7}$$

where $\tau_j(n-1)$ is the chaotic value for neuron j during the training vector before vector n, denoted $n-1$. Similarly, the first vector within $\mathbb{N}$ is denoted n_0.

The first chaotic value, $\tau_j(n_0)$, is set randomly using a pseudo-random number generator within the Boost libraries (Rivera and Dawes, 2014). Finally, α is the tent map input parameter designating skewness. Equation 3.6 and the setting of α to 0.05 within this study are duplicated from Uwate and Nishio. A visual representation of the tent map can be found in figure 3.2 (Uwate and Nishio, 2005).

A logistic map is also presented in place of the tent map in order to provide even more data for comparison. For the logistic tent map we define $\psi_j^L(n)$ as follows

$$\psi_j^L(n) = \begin{cases} 1 \text{ if } \iota_j(n) \geq u_l(n) \\ 0 \text{ otherwise} \end{cases} \tag{3.8}$$

where $u_l(n)$ is calculated as described in section 3.2.1 using the affordability variant $\psi_j^L(n)$. The map for neuron j at presentation of vector n, $\iota_j(n)$, is defined as follows

$$\iota_j(n) = (\iota_j(n-1)\beta)(1 - \iota_j(n-1)) \tag{3.9}$$

where β represents the input parameter to the logistic map. This value is set to 3.83 chosen through trial and error but any value for β after the onset of chaos (between roughly 3.5 and 4.0) will suffice. The initial value of the selection criteria for each neuron at presentation of the first training vector (n_0) is randomly initialized between 0 and 1 as follows (again, *rand* is produced using the Boost libraries (Rivera and Dawes, 2014)).

An example of chaotic affordability is as follows: Given an AfNN with one hidden layer containing four neurons with an affordability target t_l equal to two

neurons, then each time data is passed to this layer two neurons will be selected to participate while the other two will be non-contributory. This works by assigning a chaotic value to each of the four neurons and choosing the two neurons with the largest values of the four. This is repeated each time data passes forward through this layer.

$$\iota_j(n_0) = rand(0...1) \tag{3.10}$$

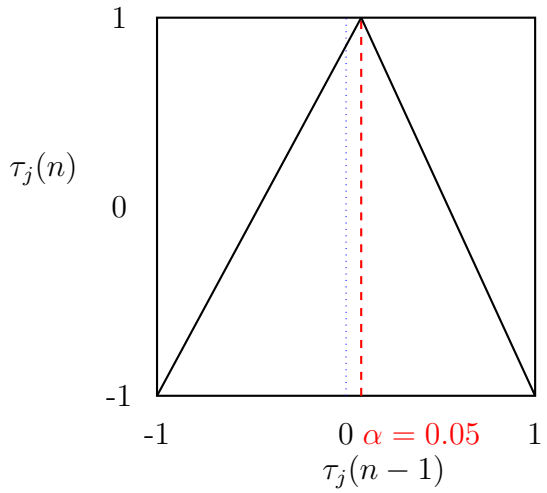


Figure 3.2: Visual representation of a skewed tent map with α value of 0.05

3.2.3 Cyclic Affordability

Research performed by the author has resulted in a simplified affordability method which has the benefit of being deterministic (much like the chaotic method) but also very consistent and understandable. Particularly, it will be shown that providing a consistent selection of groups of neurons will improve upon the reinforcement of a neuron's saliency within the network. We define

$\psi_j^C(n)$ as follows

$$\psi_j^C(n) = \begin{cases} 1 \text{ if } \gamma_j(n) \leq m_l \\ 0 \text{ otherwise} \end{cases} \tag{3.11}$$

where γ_j represents neuron the selection criteria for neuron j as follows

$$\gamma_j(n) = \begin{cases} \gamma_j(n-1) + 1, \text{ if } \gamma_j(n-1) < t_l \\ 0, \text{ otherwise} \end{cases} \tag{3.12}$$

and the initial value of the selection criteria for each neuron at presentation of the first training vector (n_0) is as follows

$$\gamma_j(n_0) = j \tag{3.13}$$

where m_l is the affordability target in the hidden layer l, as defined in section 3.2. We use j as the initial value for γ in order to provide unique values to each neuron within the range 0 to $t-1$, inclusively. Equations 3.11 through 3.13 provide the method for cyclic affordability which is unique to the research carried out in this thesis.

3.3 Comparing Affordability Methods and their Ability to Learn

In order for an objective evaluation of structural redundancy to occur a quantifiable measurement must be made against all variants of the AfNN design. For the purpose of this study, this measurement is in the form of neuron saliency using the second derivative of the objective function. After each AfNN variant has been trained each neuron will be measured against how salient it is towards the networks total error. The form of this saliency measurement is based on Optimal Brain Damage (OBD) by (Cun et al., 1990a). Whilst LeCun considered the use of entropy in order to purposefully remove the least salient items from a neural network the same calculation can be used in determining the most salient units.

However, a problem arises when considering the dynamic nature of the network architecture. At any given time, a particular neuron may not be contributing towards total error and, therefore, will have no saliency against the subsequent output. To overcome this, the calculation of h_k below is modified to include the participation scalar ψ. This means that the saliency measure is only calculated for neuron j for the trials in which it contributed towards the total error, making the calculation of saliency relevant to that neuron without compromising the method in which h_k is generated within the AfNN method. This modification is a contribution of this dissertation and used as a basis for much of the research hereafter. The examples used in this research have only one hidden layer with one output neuron, resulting in simpler calculation for saliency. The following equations are used to calculate the saliency, s_j, of hidden neuron j, as derived

from (Cun et al., 1990a)

$$s_j = \frac{h_k w_{jk}^2}{2}, \text{where } 0 \leq s_j \leq 1 \tag{3.14}$$

where w_{jk} represents the weight value associating hidden neuron j with output neuron k, and h_k is a sum across the entire training set N defined by

$$h_k = \sum_{n \in N} W(n) \tag{3.15}$$

and $W(n)$ is defined as

$$W(n) = \frac{\partial^2 E}{\partial v_k^2} \varphi(v_k(n)) \psi_j(n), \text{where } 0 \leq W(n) \tag{3.16}$$

where $\varphi(v_k(n))$ is the activation of the output neuron (as there is only one in all instances) and $\psi_j(n)$ represents the affordability of neuron j. The second derivative of the error E against the hidden layer weighted sum is as follows

$$\frac{\partial^2 E}{\partial v_k^2} = 2\varphi'(v_k(n))^2 - 2[d(n) - \varphi(v_k(n))]\varphi''(v_k(n)) \tag{3.17}$$

where w_{jk} denotes the weight connection neuron j in the hidden layer with the (only) output neuron k and $d(n)$ represents the desired output of the network for trail n within set N.

In addition to this, it is important to keep track of the saliency against unique neuron groupings. The following was derived for this study to calculate the sum of h_k relative to unique neuron groupings:

$$U(n) = \sum_j 2^j \psi_j(n) \tag{3.18}$$

where $U(n)$ is the hidden layer neuron group identifier at presentation of vector n.

$$U_p \in U(t_l) \tag{3.19}$$

where is element p within set $U(t_l)$ defined as follows:

$$U(t_l) = \{1, ..., 2^{t_l+1} - 1\} \tag{3.20}$$

where t_l represents the affordability total in hidden layer l and $U(t_l)$ represents the set of all unique hidden layer neuron group identifiers. From these we present the calculation of O_p, which represents the saliency of grouping p at presentation of vector n, as follows:

$$O_p(n, U_p) = \begin{cases} \frac{\partial^2 E}{\partial v_k^2} \varphi(v_k(n)), \text{where } U(n) = U_p, \\ 0 \text{ otherwise} \end{cases} \tag{3.21}$$

$$h_p = \sum_{n \in N} O_p(n, U_p) \tag{3.22}$$

where h_p represents the summed saliency against unique group p for all training vectors in N.

The intention of tracking the saliency of each neuron, s_j, is to confirm that a structural redundancy is being produced using the AfNN method whereby s_j distributions are being replicated within the hidden layer pool. The purpose of tracking each unique groupings saliency, h_p, is to show that group selections are more directly correlated to neuron saliency than individual selection. Together

these two saliency measurements, s_j and h_k, are used to show that AfNN methods which more consistently select unique groups of neurons will exhibit more meaningful structural redundancy of highly salient neurons.

3.3.1 Simulated Results

Two data sets are used in testing the affordability methods. The first is a reproduction of the x^2 dataset used by (Uwate and Nishio, 2005). The other is the well known Iris data set (IDS) (Bache and Lichman, 2013). The goal is to train five network variants (the five affordability methods mentioned earlier) with the two training sets for ten network configurations total.

Data Set	Afford. Method	Avg. Err	Min Err	Max Err
IDS	Classic	2.96%	2.32%	3.44%
IDS	Random	7.65%	5.94%	9.85%
IDS	Chaos (log)	7.13%	6.31%	8.67%
IDS	Chaos (tent)	7.09%	6.08%	9.49%
IDS	Cyclic	3.48%	2.48%	4.57%
x^2	Classic	3.0%	2.96%	3.05%
x^2	Random	8.58%	7.12%	9.48%
x^2	Chaos (log)	8.4%	6.43%	10.43%
x^2	Chaos (tent)	8.12%	6.97%	10.09%
x^2	Cyclic	6.37%	2.94%	8.47%

Table 3.1: Average error per epoch after training - results per network configuration.

Table 3.1 shows, per network configuration and data set permutation, statistics

on the average error, calculated over all trials in one epoch against a trained network. This data shows that, with regard to error rates, each AfNN variant produced different levels of accuracy on average. Further, whilst only somewhat portrayed in this table through the variance of total error, the learning rates of the affordability methods, particularly those of the random and chaotic types, were highly erratic throughout training.

After a network is trained the saliency, s_j is calculated using equation 3.14 against each neuron in the hidden layer for each network variant. This is repeated across twenty sessions for each affordability and for all data sets. The distributions of the results are presented in figures 3.3 and 3.4. The number of times a particular neuron is utilized within the network, c_j, based upon its relevant value for ψ is also retained. From this, the correlation $\rho(s_j, c_j)$, is calculated between the number of times a neuron was utilized, c_j, and that neurons s_j value.

Similarly, the value h_p is calculated for each unique grouping of hidden layers neurons, p, utilized during testing. A correlation, $\rho(h_p, c_p)$, is also calculated between the number of times a unique group is utilized, c_p, and its associated value h_p. The histogram of the "group" correlation coefficients is presented alongside the "individual" correlation values, per network configuration and data set in figures 3.5 through 3.12. They are organized by data set and affordability method in order to compare, relative to one affordability method and one data set, the importance of individual neuron reinforcement versus group reinforcement and selection.

In comparing both datasets within figures 3.3 and 3.4, the classic, or "zero-affordability" model is used as a control in that its saliency measurements provide the baseline against which the random, chaotic, and cyclic affordability

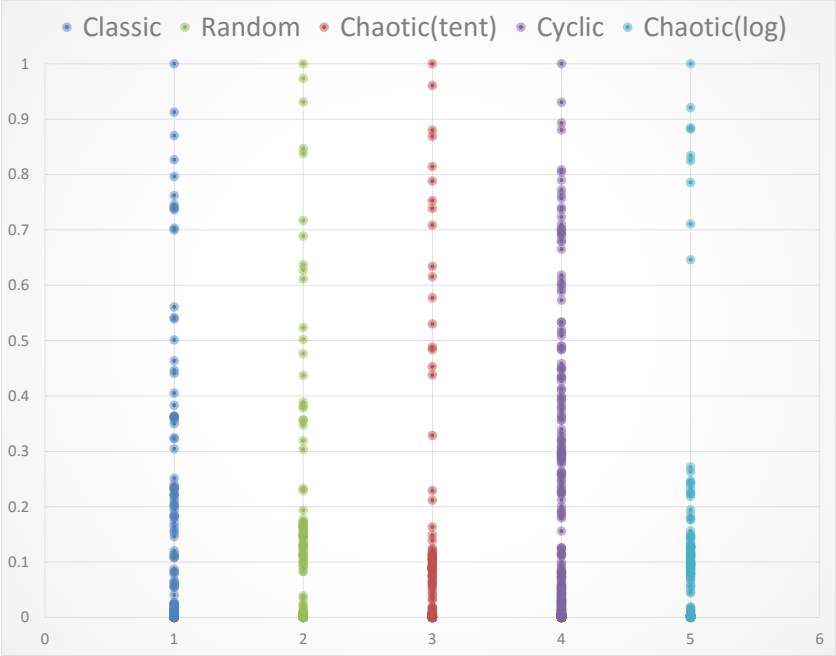


Figure 3.3: Distribution of s_j (y-axis) for each affordability method, normalized between zero and one, for the Iris Data Set

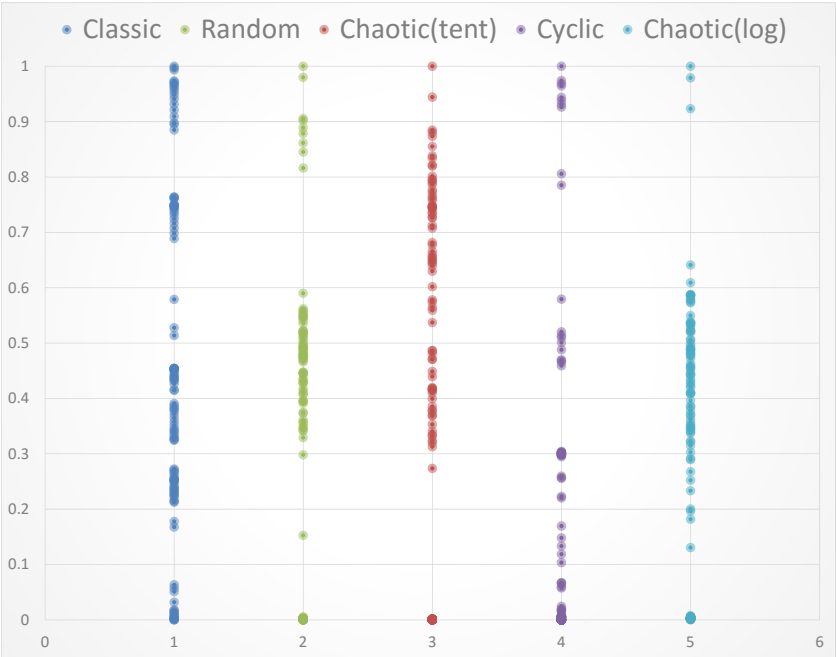


Figure 3.4: Distribution of s_j (y-axis) for each affordability method, normalized between zero and one, for the x^2 Data Set

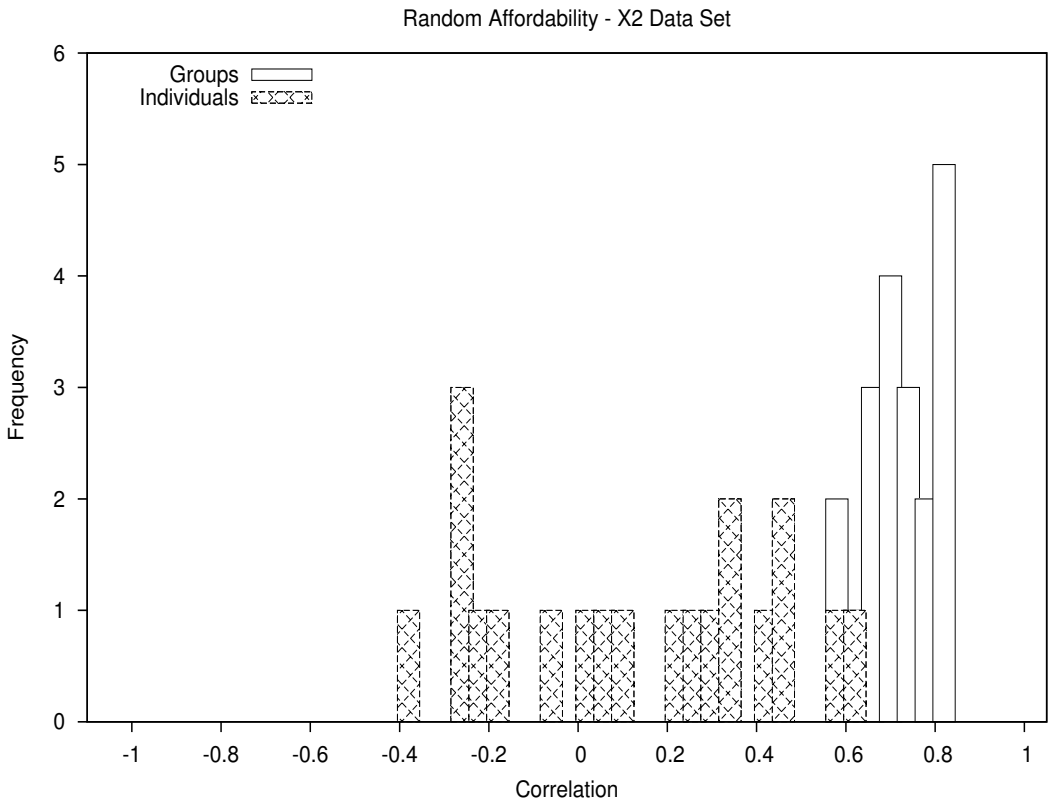


Figure 3.5: Histogram based on the x^2 data set between frequency of random selection, c_j and c_p, for the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

methods are measured.

3.3.2 Analysis and Discussion

Once a network is trained, the relationship between s_j and the total error, in relation to affordability selection, is that not selecting highly salient neurons will lead to higher error rates. In other words, an AfNN can be viewed as a short-term selective pruning of neurons. It follows that the neurons being selected through an affordability method are, preferably, those with the highest saliency.

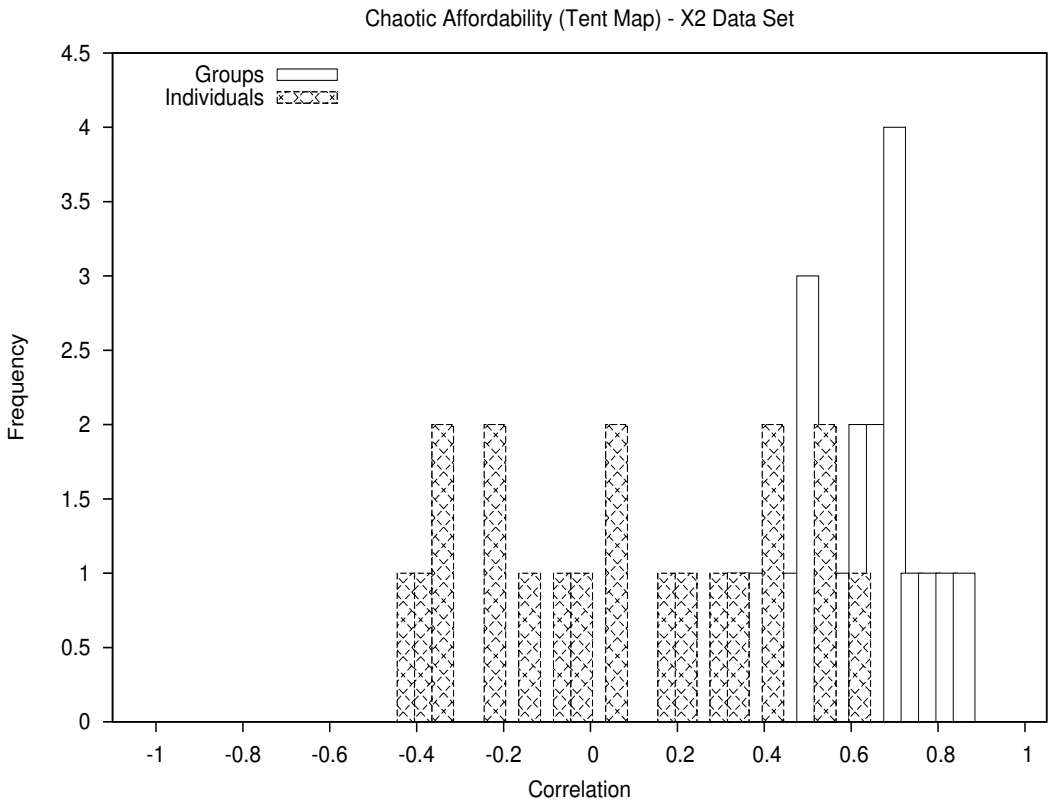


Figure 3.6: Histogram based on the x^2 data set between frequency of chaotic selection, c_j and c_p, for the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

Further, a meaningful structural redundancy of neurons will ensure that neurons with high values for s_j are selected because there are simply more of them (dependent upon affordability method and the selection process therein).

In terms of total error, the cyclic affordability method achieved the lowest error of any trial when compared to the baseline static structure (classic MLP). This method was also the only variant which satisfactorily classified all data points in the IDS. The highest error of any trial was incurred using random affordability which portrays erratic training and lowest average error in both data sets. As the cyclic affordability method was designed specifically to ensure consistent

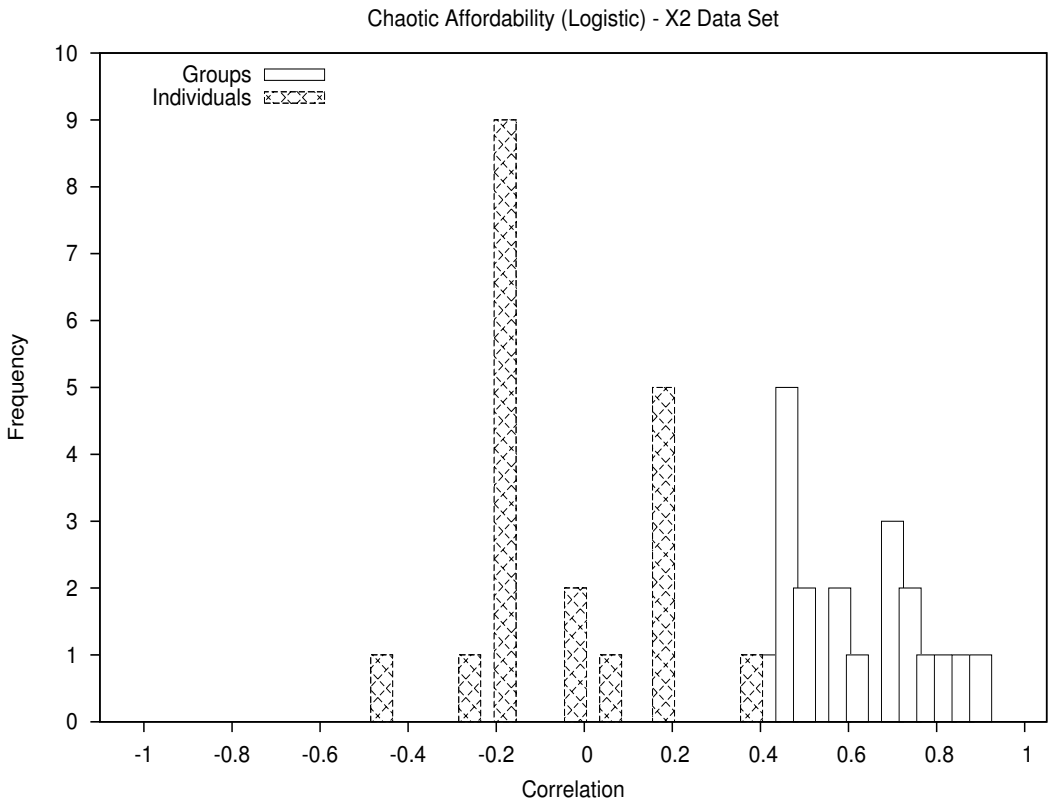


Figure 3.7: Histogram based on the x^2 data set between frequency of chaotic selection, c_j and c_p, against the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

group selection it follows that a lower error rate herein supports the claim that consistent group selection leads to lower average error after training within the data sets and configurations of the experiments presented.

If this claim is to hold true then, in the first instance, the correlation between group selection and group saliency, $\rho(h_p, c_p)$, would need to highlight a dependent relationship. Figures 3.5 through 3.12 provide this evidence. The group correlation values, $\rho(h_p, c_p)$, across all variants average above 0.5 whilst similar calculations against individual neuron reinforcement, $\rho(s_j, c_j)$, stipulate no relationship between how often a particular neuron is selected c_j and how salient it

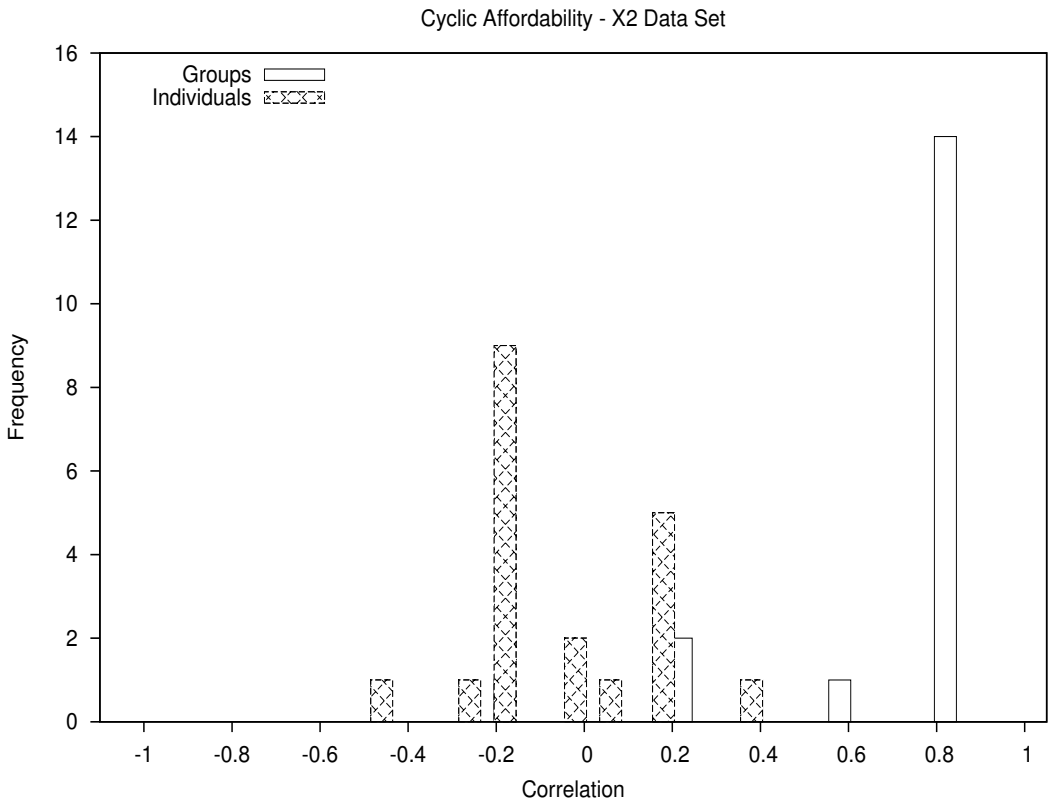


Figure 3.8: Histogram based on the x^2 data set between frequency of cyclic selection, c_j and c_p, against the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

is towards the overall network error as measured by s_j.

Given that unique group selection reinforcement leads to groups with high saliency, and given the inherent relationship between group saliency h_p and the saliency of neurons within that group, it follows that the AfNN method which consistently reinforces neuron groups across all neurons in the hidden layer will produce meaningful structural redundancy and, therefore, lower error rates.

Along these lines, the distribution of individual s_j values in figures 3.3 and 3.4 provide evidence as to the existence of structural redundancy. Looking at the saliency distributions of the classic MLP for the IDS in figure 3.3 it can be

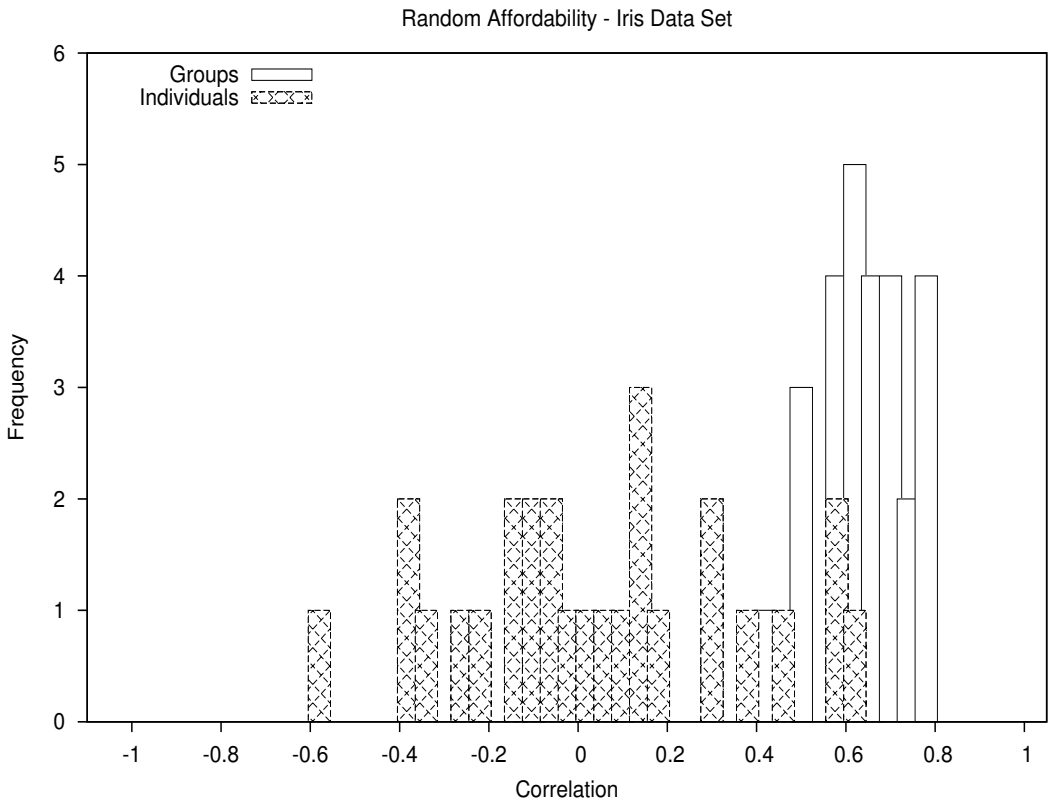


Figure 3.9: Histogram based on the Iris data set between frequency of random selection, c_j and c_p, against the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

discerned that three main groupings of saliency emerge. They can be viewed as the "near zero", above 0.2, and above 0.5 saliency clusters within the distribution. In this respect, the cyclic selection method most effectively exhibits a healthy saliency distribution in that it is not as heavily weighted near zero as the other variants are. Also, in terms of convergence, as mentioned earlier, the cyclic selection method performed best of out the AfNN variants (aside from a classic MLP approach which loses any benefit from structural redundancy). The random and chaotic variants do not contain as many highly salient neurons and, therefore, provide the least resistance to damage against the most important

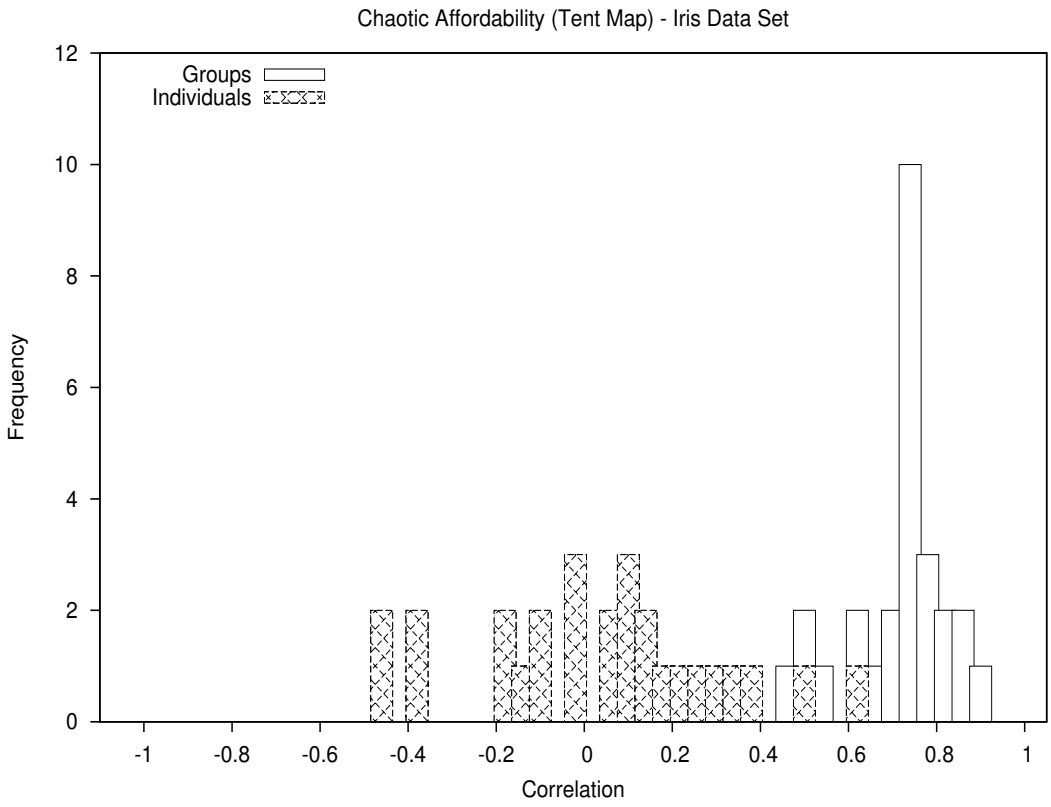


Figure 3.10: Histogram based on the Iris data set between frequency of chaotic selection, c_j and c_p, against the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

units of the network. For the x^2 data set the classic MLP distribution shows a larger amount of highly salient neurons than any of the affordability methods, with the random and cyclic methods being the closest.

Lastly, the differences between the results of the logistic chaotic and tent map chaotic configurations, whilst presented in this chapter in order to provide a meaningful comparison against the works of Uwate and Nishio, act similarly in all data presented herein. As such, in moving forward with further experiments specific to the research presented here, only one of these variations will be maintained in order to simplify future analyses.

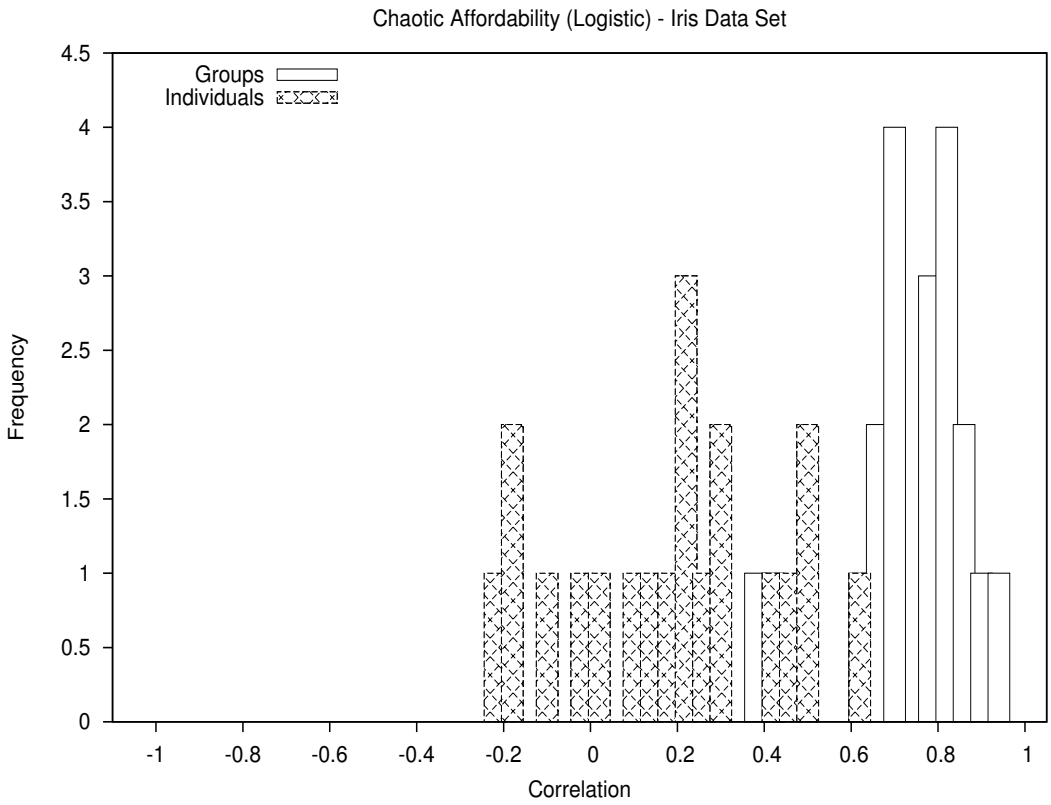


Figure 3.11: Histogram based on the Iris data set between frequency of chaotic selection, c_j and c_p, against the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

3.4 Summary

The biological brain inspires many aspects of engineering and computer science. Not just with its capacity to learn but, also, its ability to do so in such a volatile environment utilizing a mutable structure. By this, we mean, that its very cellular foundation is constantly changing and adapting whilst, normally, having no observable effect on the ability to function.

This innate structural redundancy is of core interest to the research presented here. Namely, in order to present and validate a set of measurements for determining and quantifying fault resilience two components are needed. First, a

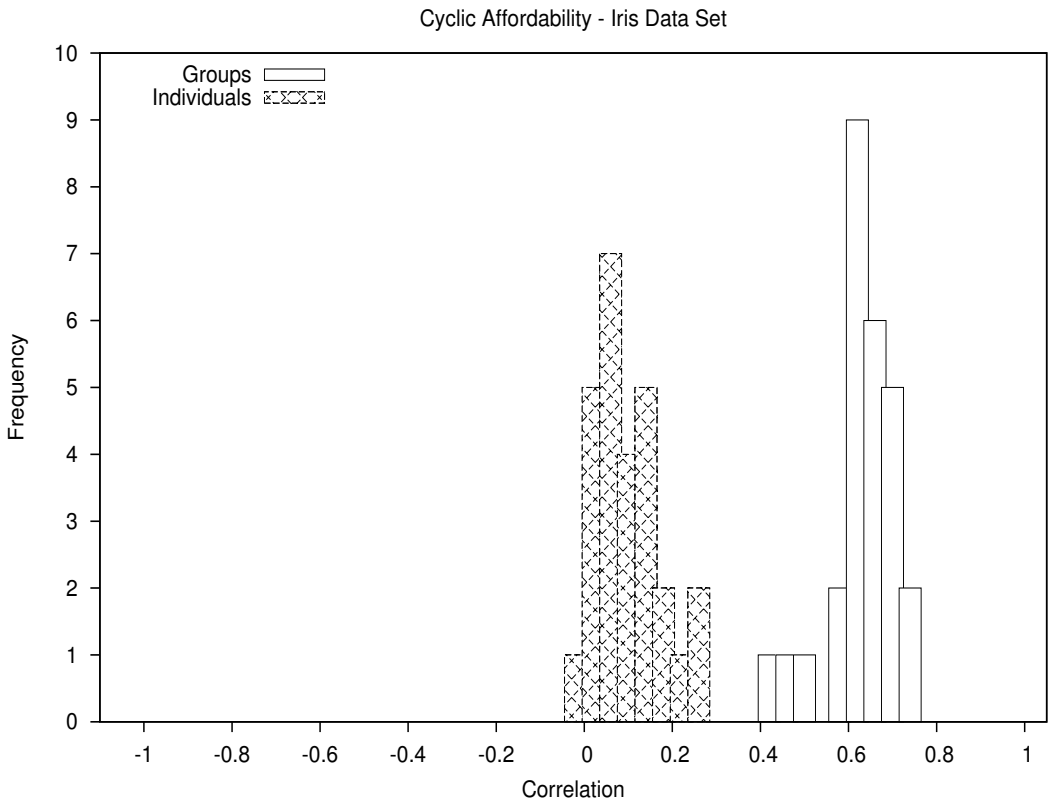


Figure 3.12: Histogram based on the Iris data set between frequency of cyclic selection, c_j and c_p, against the correlations, $\rho(s_j, c_j)$ and $\rho(h_p, c_p)$, for individuals and unique selection groups, respectively.

modified ANN needs to be defined and understood in order to provide a meaningful relative comparison of fault resilience. Second, that modified ANN is tested for existence of basic structural redundancy, without which a relative comparison would yield no meaningful conclusion. The latter is predicated on the fact that structural redundancy is the foundation of a passive fault resilience within the biological brain (Clergue and Collard, 1998) (Mulligan et al., 2010) (Michel and Collard, 1996) (Stroemer et al., 1995).

The purpose of this chapter is to establish a variation of ANN, namely the AfNN, by which comparison of fault resilience can be systematically applied

later on. However, as mentioned above, in order for this establishment to occur, the AfNN method needs to be understood and characterized.

Thus far, it is clear that the AfNN method is able to train, similarly to a classic MLP, and produce trained responses. The accuracy, or level of convergence regarding said training, varies between AfNN variant which is ideal for the overall aims of the research presented here. It is also evident, as depicted in figures 3.3 and 3.4, that the AfNN method provides variations in the levels of saliency of neurons in the hidden layer. These differences are the catalyst for varying levels of structural redundancy and supports the next step in the research presented here.

Chapter 4 provides the first set of measurements, derived from the Mean-squared Error (MSE) calculated whilst using the AfNNs presented, towards understanding and evaluating fault resilience. These measurements, as well as those presented later in this research, comprise the foundation of novel contributions herein.

Chapter 4

Measuring Fault Resilience: Mean Squared Error

In chapter 3 we introduced the concept of Affordable Neural Networks (AfNNs) and their inherent structural redundancy, the prerequisite for meaningful fault resilience calculations. Having established that the AfNN method is able to learn a data set to a satisfactory level (3.3), using multiple variants of the method, and having also confirmed that some level of structural redundancy exists therein, a new set of experiments are undertaken in order to further investigate fault resilience.

In this chapter, the aim is to explore the measurements related to the ability of the AfNN method to sustain damage. These measurements range from quantifying error changed due to loss of neuron, using Mean-squared Error (MSE)-based functions and can help to further evaluating the benefit of the affordability method. In other words, where chapter 3 aimed to discover fault resilience through the corollary of saliency of neurons, this chapter presents a more thor-

ough quantification into various aspects of fault resilience towards providing a set of universal measurements.

Damage, from the perspective of the AfNN method, is similar to a stuck-at-0 fault in the sense that the neuron will not contribute to the output of the network nor will the damaged neurons weights be updated during back propagation. Further, as the selection of neurons within the AfNN acts as a binary switch on each neuron, affecting a number of neurons up to the affordability target, neurons marked as damaged are ignored during this neuron selection.

The first step is to further test fault resilience across more data sets. Section 4.1 details the data sets chosen, and the reasons behind the selections. The reasoning behind this lies with the goal of establishing a diverse and consistent foundation for measuring fault resilience throughout the remainder of the research presented in this dissertation (the works presented by Uwate and Nishio fail to provide such comprehensive evaluation (Uwate and Nishio, 2005) (Uwate et al., 2007) (Uwate and Nishio, 2010)). Section 4.2 analyses measurements related to basic fault resilience which, in this experiment, are defining and comparing the accuracy retained and sustained through damage to the AfNN. Section 4.3 defines a new experiment to test how the affordability method lends itself to the results of experimental results in section 4.2 by further quantifying how the affordability threshold affects fault resilience. Similarly, section 4.4 revisits the experimental results presented in section 4.3 but, instead, focuses on how experimental results compare to statically structured Multilayer Perceptrons (MLPs), holding data sets and hidden layer size constant.

Overall, this chapter aims to take the affordability method presented in chapter 3 and further analyse how and why the method is able to provide fault resilience and quantify the value added, therein.

4.1 Data Sets

At this point in the study, a number of changes are made to the design of further experiments. Specifically, whilst two chaotic variants have been presented thus far with slightly varied results it is decided that it is not statistically significant enough to proceed with both, as indicated in section 3.3.2. Therefore, the logistic map (equation 3.9) is used for all further experiments as the chaotic variant (refer to table 4.2 and figures 3.3 and 3.4). Although, the random affordability is expected to behave just like another chaotic map, given the implementation of boost (Rivera and Dawes, 2014). However, it is left in for completeness of comparison against the source material because the results related to this AfNN variant do provide statistically interesting deltas to those of the chaotic variant. Alongside the changes made to the AfNN methods used, the data sets used for further experiments are altered as well. The x^2 data set is removed due to its obvious linear separability and the lack of insight provided by results reported in chapter 3. Three more data sets are added in its place. At first consideration, these three new sets were chosen simply because they have similar characteristics (discussed below) to that of the Iris data set, making calculations and comparisons relevant across further results. However, what we will see in the proceeding chapters is the real effect these data sets will have on our results, regardless of these similarities.

The three new data sets are the Combined Cycle Power Plant (CCPP), Servo, and Balance Scale sets from the University of California, Irvine Machine Learning Repository (UCI) repository (Bache and Lichman, 2013). They all share a common number of values per pattern, four input parameters and one output, with the Iris data set. Also, the data in each set are taken as are and systemat-

ically sampled to produce training and test sets with a ratio of two to one. The pseudo-code for the training and testing set generation is found in algorithm 1. This algorithm is presented for a number of reasons. First, for completeness in describing the method for creating data sets for experiments herein and, secondly, as a baseline for an experiment undertaken in chapter 6 where algorithm 1 is modifed. Table 4.1 details various quantifiable comparisons between the data sets chosen. Input and Output values are given as ranges ([...]) or sets of values ({a, b, c}). The purpose of presenting table 4.1 is to provide a basis of comparison for the analysis and discussion occuring later in this chapter.

With respect to maintaining the same number of neurons across all data sets, consider the findings discussed in previous research regarding fault tolerant neural networks in section 2.3. Damarla and Bhagat (Damarla and Bhagat, 1989) note that like-sized MLPs, with respect to the number of neurons in each layer, are comparable. This implies that both affordability targets and totals should be comparable across all simulations.

Algorithm 1 Creation of Training and Testing Sets

```
function NORMALIZE DATA SET(dataSet)
    for each inputValueList ∈ dataSet do
        maxValue ← max(inputValueList)
        minValue ← min(inputValueList)
        for each value ∈ inputValueList do
            value ← (value − minValue)/(maxValue − minValue)
        end for
    end for
    for each outputValueList ∈ dataSet do
        maxValue ← max(inputValueList)
        minValue ← min(inputValueList)
        for each value ∈ inputValueList do
            value ← (value − minValue)/(maxValue − minValue)
        end for
    end for
end function

function CREATE SETS(aggregatedDataSet)
    NORMALIZE DATA SET(aggregatedDataSet)
    n ← sizeof(aggregatedDataSet)
    m ← n mod 3
    n ← n − m
    for i = 0 to n do
        trainingSet ← learningPattern[i]
        trainingSet ← learningPattern[i + 1]
        testingSet ← learningPattern[i + 2]
        i ← i + 3
    end for
    for i = 0 to m do
        trainingSet ← learningPattern[i]
        i ← i + 1
    end forReturn trainingSet, testingSet
end function
```

	Data Sets			
	Iris	Servo	Balance	CCPP
Learning Task	Classification	Regression	Classification	Regression
Instances in Training	100	112	345	6403
Instances in Testing	50	55	280	3465
Input Value 1	[7.9 to 4.3]	{A, B, C, D, E}	{B,R,L}	[37.11 to 1.81]
Input Value 2	[4.4 to 2.0]	{A, B, C, D, E}	{1, 2, 3, 4, 5}	[81.56 to 25.36]
Input Value 3	[6.9 to 1.0]	{A, B, C, D, E}	{1, 2, 3, 4, 5}	[1033.3 to 992.89]
Input Value 4	[2.5 to 0.1]	{A, B, C, D, E}	{1, 2, 3, 4, 5}	[100.16 to 25.56]
Output Value 1	{Iris Setosa Iris Versicolour Iris Virginica}	[7.10 to 0.13]	{1, 2, 3, 4, 5}	[495.76 to 420.26]

Table 4.1: List of attributes for each data set for comparison.

The CCPP data set is unique in that the amount of data available from the UCI repository is five times that which is used by this experiment and considerably more than the other data sets used in this research. It is found that attempting to create both training and testing sets from all of the data available results in a very lengthy learning phase, temporally, as well as metrics which are exaggerated to the point of being devoid of any nuance between experimental configurations. As such, the fifth data set present within the CCPP source data is the only set

used (Bache and Lichman, 2013).

In all experiments that follow, the same number of neurons in the hidden layer are used across all data sets. This is in spite of what would be considered optimal for any particular set and is intentional. The motivation for this is to provide a consistent experiment with respect to the metrics gathered whilst constructing, training, damaging, and recuperating each network configuration using each data set in each experiment, dependent upon the context of each experiment therein. The reasoning here is that a like-for-like comparison regarding AfNN fault resilience measures is desired above the most efficient network configuration needed for convergence. This comparison is the basis of all future experiments. Further, it is clear, from the results gathered thus far, that each data set's optimal number of neurons in the hidden layer is unique amongst the group.

Similarly, each experiment only considers one hidden layer within each MLP configuration for a given experiment. No analysis is performed, nor research undertaken, to determine whether any of the data sets mentioned (Iris, Servo, Balance Scale and CCPP) benefit from having more hidden layers. This is partly because none of the experiments described suffer from an inability to converge to acceptable accuracy and also because, even if more hidden layers were to provide more efficient training, the results contained herein aim to keep this element of the experiment constant in order to focus on a relative comparison of fault resilience, as opposed to generalization optimization.

Lastly, and to more explicitly quantify the aforementioned restrictions on hidden layer design, the use of twelve total neurons for the affordability total (in use cases utilizing AfNN methods) whereby the affordability target is set to eight (dependent upon how many neurons remain within the hidden layer) will be the

standard configuration for all experiments. Likewise, for the classic MLP construction, all neurons are used in the hidden layer starting with a total of eight and subject to structural damage (loss of neurons).

4.2 Comparing Damage Resilience Across Multiple Data Sets

Having established that the AfNN method is able to train one data set to a satisfactory level (using multiple variants of said method) and having also confirmed that some level of structural redundancy exists therein, the following experiment is taken in order to further investigate. The first step in this new experiment is to test fault resilience across more data sets. This section details the data sets chosen, the reasons behind the selections, and also the definition of a new experiment to measure fault resilience of the AfNN method after incurring damage to it's structure (loss of neurons and their connections).

4.2.1 Effect of Retraining on Post-Damage Error Rates

The first experiment related to evaluating damage recovery further is as follows. Of the four affordability methods (classic, random, chaotic and cyclic) each is trained to a target MSE (detailed in table 4.2). During training, the saliency of each neuron is calculated per equation 3.14. Post-training, the most salient neuron is removed from the hidden layer and retraining is performed for a set number of epochs. During retraining, saliency and MSE are recalculated and the process is repeated until only one neuron remains in the hidden layer, at which

point MSE is calculated and no further pruning occurs. For reference, please note the following equation for calculation of MSE.

$$MSE_k = \frac{1}{N} \sum_{i=0}^{N} \hat{\epsilon}(\nu(n_i))^2 \tag{4.1}$$

where $\nu(n_i)$ represents the nth input vector within the total number of inputs N and $n \in \mathbb{N}$. $\hat{\epsilon}$ is the difference between the networks output (eq. 3.2) in response to input ν and the desired output. The number of epochs used during retraining is one of either ten, one-hundred, or one-thousand. If the network variant is able to retrain to the original target MSE then retraining stops early, in all cases. Due to there being four affordability methods and, for each method, three levels of retraining run against four separate data sets, forty-eight total configurations are being compared in this experiment. Further, each of the configurations are run ten times and the results averaged. This is because the networks' weights are always randomized and there are some runs that result in skewed measurements where the retraining occurs faster or slower than on average.

$$y'_k = \frac{1}{10} \sum_{10} MSE_k \tag{4.2}$$

Here, y'_k represents the average MSE for output neuron k over ten runs, as described above. Similarly, the average MSE for a neuron in the hidden layer, denoted y'_j, is defined as follows.

$$y'_j = \frac{1}{10} \sum_{10} MSE_j \tag{4.3}$$

The relationship between MSE_j and MSE_k is in relation to a standard back

propagation of error. The values for y'_k and y'_j are calculated each time a neuron is removed after some level of retraining. In this respect, the designation used to describe the value of y'_k against a network with g neurons remaining in the hidden layer is $y'_{k,g}$. The following equation represents the average value for MSE lost per neuron removed, or the mean-summed difference in MSE, denoted $dMSE$.

$$dMSE = \frac{1}{t_l - 1} \sum_{g=2}^{t_l} y'_{g,1} - y'_{g-1,1} z'_g = y'_{g,k} \quad \text{where} \quad k = 1 \tag{4.4}$$

where t_l (eq. 3.1) represents the total number of neurons (prior to damage) of the network (this value is twelve for all but the classic MLP which only has eight). Finally, the total averaged, $tMSE$, across all neurons lost is represented as follows.

$$tMSE = \frac{1}{t_l - 1} \sum_{g=1}^{t_l - 1} y'_{g,k} \tag{4.5}$$

Given these equations, specifically 4.4 and 4.5, we are ready to make a hypothesis regarding the behaviors of the affordable network configurations under experimentation. The following redefines hypothesis 1 (section 2.4, page 34) using the equations above.

> **Hypothesis 6** *Networks which utilize the affordability method will exhibit a smaller total average MSE as levels of retraining increases, measured using equation 4.5.*

As stated earlier in section 4.2, the most salient neuron is removed as the mech-

anism for introducing damage. This is intended to cause the most disruption possible. The varying levels of retraining (i.e. the number of epochs used between onsets of damage) allow the saliency landscape of the remaining neurons to cope with the loss of the previously most valuable neuron therein, measured using eq. 3.14. Part of this experiment, alongside studying the data with respect to hypothesis 6 is to also gather understanding into how the average MSE lost per neuron removed relates to our expectations. Towards this end, another hypothesis is provided based on hypothesis 2 (section 2.4, page 34).

> **Hypothesis 7** *The more retraining that occurs between onsets of damage the lower the average error lost per neuron, measured using equation 4.4.*

Hypothesis 7 captures the expectation that fault resilience and preservation of function is akin to minimizing changes in y'_k as neurons are removed.

4.2.2 Simulated Results

The first table presented (table 4.2) captures the setup and training results against all four data sets with each of the four AfNN variants prior to damage. Of note are the data set and AfNN variants pairings which are able to train "early". That is, to say, when the pairing is able to meet the target MSE before the maximum epochs are reached. The reasoning behind providing this table is both for completion and as a baseline for data presented from here on. Without knowing how well each network configuration and data set combination was able to initially perform prior to damage, the comparisons therein would be incomplete.

Table 4.3 is presented to demonstrate hypotheses 6 and 7. The data therein presents is directly related to both equations 4.4 and 4.5 against each of the data set and affordablility method combinations.

		Target y'_k	Max Epochs	Learn Rate	y'_k Acheived	Epochs Used
Iris	Classic	0.01	1500	0.23	0.01	250
	Random	0.01	1500	0.23	0.01	1500
	Chaotic	0.01	1500	0.23	0.01	1500
	Sequential	0.01	1500	0.23	0.01	350
Balance	Classic	0.01	1500	0.23	0.02	1500
	Random	0.01	1500	0.23	0.03	1500
	Chaotic	0.01	1500	0.23	0.03	1500
	Sequential	0.01	1500	0.23	0.01	1500
Servo	Classic	0.02	1500	0.23	0.02	209
	Random	0.02	1500	0.23	0.02	461
	Chaotic	0.02	1500	0.23	0.02	516
	Sequential	0.02	1500	0.23	0.02	254
CCPP	Classic	0.01	1500	0.23	0.01	1
	Random	0.01	1500	0.23	0.01	5
	Chaotic	0.01	1500	0.23	0.01	6
	Sequential	0.01	1500	0.23	0.01	1

Table 4.2: Results of training all four data sets against all four AfNN variants prior to damage.

		$tMSE$			$dMSE$		
		10	100	1000	10	100	1000
Iris	Classic	0.0198	0.0143	0.0133	0.0047	0.0035	0.0033
	Random	0.0182	0.0156	0.0136	0.0028	0.0025	0.0024
	Chaotic	0.0216	0.0154	0.0127	0.0032	0.0023	0.0020
	Sequential	0.0208	0.0147	0.0137	0.0027	0.0021	0.0024
Servo	Classic	0.0178	0.0136	0.0138	0.0028	0.0016	0.0024
	Random	0.0187	0.0168	0.0148	-0.0004	0.0021	0.0032
	Chaotic	0.0187	0.0178	0.0147	-0.0004	0.0027	0.0032
	Sequential	0.0158	0.0136	0.0133	0.0001	0.0010	0.0021
Balance	Classic	0.0344	0.0288	0.0311	0.0083	0.0081	0.0074
	Random	0.0269	0.0263	0.0239	0.0014	0.0027	0.0013
	Chaotic	0.0294	0.0272	0.0264	0.0041	0.0029	0.0032
	Sequential	0.0344	0.0278	0.0266	0.0079	0.0067	0.0057
CCPP	Classic	0.0033	0.0035	0.0032	5E-05	5E-05	-6E-05
	Random	0.0045	0.0043	0.0043	3E-05	-2E-05	7E-06
	Chaotic	0.0044	0.0042	0.0043	2E-05	-6E-05	3E-05
	Sequential	0.0034	0.0033	0.0033	4E-05	6E-06	2E-05

Table 4.3: $dMSE$ and $tMSE$ results against affordability method and data set showing values for three levels of post-damage retraining.

4.2.3 Analysis and Discussion

We begin with an analysis of table 4.2. Starting with the Iris data set it is shown that all network configurations are able to achieve the target average accuracy as measured by equation 4.2. This is also true for the CCPP and Servo data sets. The only exception, in this regard, is with the balance data set which, in

the current experiment configuration, the four network configurations achieved varying levels of initial average accuracy, as depicted in table 4.2; with values ranging between 0.01 and 0.03.

Also of interest is a look into the number of epochs needed for each of these configurations to achieve the listed error rates. For the Iris data set the classic and sequential configurations are able to terminate training early, at 250 and 350 epochs, respectively. However, the chaotic and random cases must utilize all of the available epochs to reach the desired goal. The implication, based on the data presented within the system of this experiment, is that one epoch of training provides more value to the sequential and classic variants as compared to the chaotic and random. This observation is of interest with respect to hypotheses 7 and 6 which are concerned with measuring the added value of retraining post-damage. The CCPP and Servo data sets also depict early terminations with the classic and sequential cases, once again, depicting the higher average value added per epoch of training.

Lastly, the Balance data set portrays a data set which is seemingly difficult to train against, as depicted by both the utilization of all available epochs and the inability, in nearly all cases (barring the sequential variant), to achieve the target y'_k.

The remainder of analysis and discussion for this section will be split into two parts, one for each of the hypotheses presented. First, $tMSE$ values across all configurations are examined with respect to hypothesis 6 to ascertain how varying levels of retraining affect MSE, on average. Next, examination of $dMSE$ in relation to hypothesis 7 towards understanding whether or not basic fault resilience exists through measurement of average MSE introduced, per neuron lost, in all configurations.

4.2.3.1 Total Average MSE

The Iris data set holds true in all cases in that, per the values presented in table 4.3, measurements of $tMSE$ meet the expectation captured in hypothesis 6. As does the Servo data set. The Balance data set did not hold true in the Classic MLP variant in that error increased between one-hundred and one-thousand epochs. The CCPP data set did not hold true in both the classic and sequential variants. The former increase from 0.0033 to 0.0035 between ten and one-hundred, going back down to 0.0032 in the one-thousand level of retraining. The latter increased from 0.0042 to 0.0043 between one-hundred and one-thousand epochs.

All in all, hypothesis 7 holds true in most cases. This is to say that, in regards to these data sets and affordability configurations, higher levels of retraining between removal of neurons does indeed have a positive impact on how well the networks are able retain value as measured using $tMSE$ (eq. 4.5). These experimental results do not, however, detail whether or not the affordability method provides more or less fault resilience than the classic MLP, as will be the subject of proceeding experiments.

4.2.3.2 MSE Per Neuron Lost

The Iris data set holds true with respect to hypothesis 7 in all cases but the sequential data set. In this instance, one-thousand epochs of retraining loses value as shown by the increased average error per neuron lost. However, the

$dMSE$ value for this case is lower than the average for this data set at this level of retraining. In other words, it is more accurate to describe the one-hundred epoch case for the sequential configuration as *outperforming* the other configurations at this level of retraining as opposed to claiming that the one-thousand epochs of retraining is *underperforming*.

The servo data set, in all cases, did not meet the expected outcome. The classic network configuration, at least, saw improvement between ten and one-hundred levels of retraining; worsening thereafter. The other three configurations, random, chaotic, and sequential, all strictly lost value (gained error per neuron lost as the epoch ceiling increased) throughout the experiment as measured using equation 4.4.

The balance data set failed to satisfy hypothesis 7 for the random and chaotic cases, whilst meeting expectation for both the classic and sequential. With respect to the random affordability configuration the outcome seems akin to that of the sequential configuration of the Iris data set results; the results associated with ten epochs of retraining has drastically outperformed the remaining configurations leading to the result described. As for the chaotic variant, the increase of $dMSE$ between one-hundred and one-thousand epochs of retraining, whilst in line with the smallest change between levels of retraining against this data set, is enough to warrant a failing condition.

Finally, the CCPP data set, like the balance data set, fails to meet expectation in both the random and chaotic variants. However, the values of $dMSE$ are so small in this test that the results vary between positive and negative average MSE per neuron lost. Overall, due to the drastically small margin of change exhibited against this data set, the hypothesis is neither satisfied nor contradicted within the data presented.

4.3 Measuring Value Added Through Affordability

The affordability method exhibits, thus far and with respect to the data sets studied, a level of retained value in the presence of damage as measured using equations 4.5 and 4.4 in relation to the experiment in section 4.2. This experiment aims to further elucidate whether the results documented are due to the redundancy provided by the affordability method or, perhaps, whether the data sets and neuron selection method are the cause. To do so, the affordability threshold (reference definitions in section 3.1, page 39) is used to discriminate the measurement of fault resilience. In doing so, it is expected that the value retained in the presence of damage above the affordability threshold will equal or exceed the retained value below it and the following experiement is designed accordingly.

4.3.1 Determining the Value of Affordability

Building upon the equations presented in section 4.2 the following functions are provided to measure fault resilience value with respect to the affordability threshold discriminator. The value for y'_k (4.2) is averaged with respect to neurons remaining both above and below the affordability threshold as follows:

$$AMSE = \frac{1}{r}\sum_{j=m}^{t} y'_j, \text{ where } m > t \tag{4.6}$$

where r comes from equation 3.1.

$$BMSE = \frac{1}{m}\sum_{j=1}^{m} y'_j, \text{ where } m > t \tag{4.7}$$

where $AMSE$, the mean-squared error above the affordability threshold, is the rolled-up MSE for each of the neurons in the hidden layer above t. Similarly, $BMSE$ is the rolled-up MSE average below the threshold, inclusive of the affordability threshold value. Finally:

$$\Delta MSE = AMSE - BMSE \tag{4.8}$$

where ΔMSE represents the difference between quantities $AMSE$ and $BMSE$, as defined in equations 4.6 and 4.7. The range of possible values are $\Delta MSE \in 0, 1$. As such, since the classic MLP does not provide any affordability the assumed cumulative average error (i.e. $AMSE$) is set to one. This measure is unique in that the sign of the value indicates whether or not the network benefits, on average, from the affordability method (i.e. extra neurons above the threshold are out-performing neurons within a damaged, non-affordable, network). A negative value indicates *less* error, whilst a positive one indicates more with respect to the values above and below the affordability threshold.

The existence of structural redundancy through the affordability method provides quantifiable value added, as measured by Delta Mean-squared Error (ΔMSE) (eq. 4.8), when compared to network not utilizing affordability. The expectations of the experiment can be described with the following (a redefinition of hypothesis 3; section 2.4, page 34)

Hypothesis 8 *Networks which utilize an affordability method will achieve* $\Delta MSE < 0$ *(equation 4.8) at all levels of retraining, and, therefore will provide more added value than an MLP which does not provide affordability.*

4.3.2 Simulated Results

Table 4.4 depicts the results of the current experiment. This table is organized to show the value of ΔMSE (eq. 4.8) against each of the affordability configuration and data set combinations. Each column in the table corresponds to various levels of retraining. The organization of this information is designed such that an analysis against hypothesis 8 is readily made.

As for how to interpret the value of ΔMSE presented in table 4.4 the key is in understanding the difference between AMSE (eq. 4.6) and BMSE (eq. 4.7) and what this may reveal about how the existence of an affordability threshold affects fault resilience. It is expected, in this experiment, as captured in hypothesis 8, that a negative value for ΔMSE equates to a particular data set and configuration for which affordability provides added value to the network in the form of fault resilience. It does not, however, indicate that the average MSE above or below the affordability threshold are devoid of value. It indicates that, as the network loses neurons and becomes more akin to a regular MLP, value is lost by *losing* affordability. The opposite is also true in that a positive value for ΔMSE indicates that the existence of affordability provided a net loss of value (or net gain in error) as compared to the same network after reading zero affordability.

		ΔMSE		
		10	100	1000
Iris	Classic	0.9763	0.9854	0.9877
	Random	-0.0040	0.0012	-0.0003
	Chaotic	-0.0056	-0.0016	-0.0012
	Sequential	0.0073	0.0031	-0.0013
Servo	Classic	0.9816	0.9861	0.9867
	Random	0.0006	0.0025	-0.0030
	Chaotic	0.0007	0.0022	-0.0016
	Sequential	0.0041	0.0011	-0.0017
Balance	Classic	0.9576	0.9724	0.9709
	Random	0.0031	0.0049	0.0017
	Chaotic	-0.0008	0.0027	-0.0006
	Sequential	0.0023	0.0027	-0.0039
CCPP	Classic	0.9969	0.9968	0.9971
	Random	0.0046	0.0033	0.0031
	Chaotic	0.0047	0.0045	0.0044
	Sequential	0.0012	0.0012	0.0012

Table 4.4: ΔMSE results against affordability method and data set showing values for three levels of post-damage retraining.

4.3.3 Analysis and Discussion

It is worth mentioning again, as noted in section 4.3.2, that within the current experiments, once an AfNN loses enough neurons (in this case, four) and the network size goes below the affordability threshold, it essentially becomes a classic MLP. This means that, when reviewing fault resilience measurements, there is merit in both the results produced above and below the affordability threshold

(revisit definition in section 3.1).

Given that each affordability method is able to converge to near classic MLP levels as mentioned earlier and evidenced by table 4.2, the results support that a pre-damaged and pre-trained AfNN holds similar potential value as that of a classic MLP in all cases except those noted against the Balance data set. As damage is sustained, the existence of affordable selection and, thereby, structural redundancy, which provides (in all but the cases mentioned) similar value to that of a classic MLP, is the basis for fault resilience. In other words, fault resilience is, in this experiment, the measurement of how much value the affordability method retains as measured by equation 4.8 and described in hypothesis 8. Where this assumption is tested is in the comparison of varying levels of retraining between onsets of damage, as per previous experiments, which is expected to affect the value of ΔMSE.

If hypothesis 8 is true then, at all levels of retraining, the value depicted by ΔMSE should be less than zero. The error accumulated above the threshold should be less than that of below the threshold. It should also be noted that the Classic MLP is *not* expected to provide a ΔMSE value less than zero since it does not contain affordability, by definition in section 3.1. If ΔMSE were to be positive then the affordability method, within the context of network damage, provides less accuracy above the affordability threshold compared to that of a regular MLP and, therefore, the cost of redundancy negates any potential fault resilience within the context of this experiment. From looking at table 4.4 it is clear that this is not the case.

Before discussing each data set and affordability variant combinations individually it is important to state that, per hypothesis 8, the only experimental configuration that illustrates the behaviors expected is chaotic affordability for the

Iris data set. In no other instance did all three levels of retraining meet the expectation of hypothesis 8. This means that, in the first instance, optimization of ΔMSE is dependent epon the level of retraining that occurs.

The classic MLP, in all cases, depicts a positve value for ΔMSE, as expected since no affordability exists. Of note is that the value does worsen in that against the Iris and Servo data set cases, the values for ΔMSE gets closer to one as levels of retraining increases. Given that $AMSE$ is one for these cases the conclusion is that $BMSE$ is worsening and levels of retraining increases. This is not true for the Balance and CCPP cases where values fluctuate. The reason this is of interest, and as an observation regarding the measurement of ΔMSE in general, is that the value for $BMSE$ for both the Iris and Servo data sets *decreased* as training epochs increased, which reflects the findings in table 4.3. However, because the aforementioned measurement is in relation to affordability and the value added therein, a value of one for $AMSE$ in all cases results in an overall increasing trend in the value measured.

Considering the AfNN methods against the Iris data set, the random and chaotic variants both failed to meet the hypothesis. The random affordability method exhibited negative values for the ten and one-thousand level of retraining and the chaotic method provided negative values for all three. The sequential data set, similarly, provided positive values for ten and one-hundred levels but met the expectation of hypothesis 8 for the one-thousand epoch case.

The Servo data set tells a different story. The random, chaotic, and sequential all fail to meet the hypothesis for ten and one-hundred levels of retraining but, conversely, succeed by providing negative values for ΔMSE against one-thousand level of training. The Balance data set is similar in that the chaotic and sequential cases exhibit some negative values. However, the random variant

is the first instance whereby all levels of retraining exhibit positive values for ΔMSE. The CCPP data set also provides a positive result in all three of the affordability methods tested.

In light of these results it seems as though the expectation, as set out by hypothesis 8 is not always correct. It may be safer to assume that some level of retraining can be applied to the affordability method to optimize the value added in using such a method, as measured using ΔMSE but that simply applying the algorithm will not guarantee such an outcome. Attempts to maximize affordability, whilst of interest, is beyond the scope of this dissertation; this research is only concerned with how to measure this value, not optimize it. What needs to be investigated further is whether or not the data itself has led to this result or if the algorithm utilized is simply variable with respect to numbers of retrinaing epochs. Also, is a hetero-relative comparison of an affordability variant against itself the same of comparing the value of affordability against statically sized MLPs. That is to say, as a network is damaged, reaches zero affordability, and continues to lose neurons thereafter, are the results discussed in this section affected by the levels of retraining above the affordability threshold? Or, alternatively, will the results hold true when comparing against MLPs which are constructed to have a static number of neurons in the hidden layer, randomly initialized, and trained?

4.4 Comparison of Fault Resilience using MSE Against Structurally Static Controls

As mentioned in this chapter's introductory section, our next experiment at this point in the research presented here is to further analyze the data gathered surrounding the MSE metrics thus far. The findings related to the experiment discussed in section 4.3.3 find positive early results with respect to the existence of fault resilience within the AfNN method. However, what is not clear is whether the network performance which led to this discovery is unique to the AfNN method or not. Specifically, it is worth analyzing whether or not the same levels of accuracy can be obtained in statically structured networks compared to AfNNs which have been damaged and re-trained. Details of this experiment are presented below.

4.4.1 Experiment Design

For this experiment, the data used in section 4.3 is reused but compared against a new set of data. The MSE values represented in table 4.3 are compared against MSE values obtained from statically structured MLPs. These MLPs are given hidden layer sizes between one and eleven and compared against the equivalent AfNN variants during retraining (e.g. when an AfNN variant is damaged to the point of having only six neurons in its hidden layer then it will be compared to a statically structured MLP with six neurons in its hidden layer). The MSE values obtained in this experiment are also averaged over multiple runs, as in the previous experiment in section 4.3.

$$dMSE_{static} = \frac{1}{t_l - 1} \sum_{g=2}^{t_l} y'_{g,static} - y'_{g-1,static} \tag{4.9}$$

where t_l (eq. 3.1) represents the total number of neurons (prior to damage) of the network (this value is twelve for all but the classic MLP which only has eight). $y'_{g,static}$ represents the output of a statically structured MLP with g neurons in its hidden layer (the output neuron count is always one, as with the AfNN models so we, therefore, omit k for simplicity). Finally, the total averaged MSE across all neurons lost is represented as follows.

$$tMSE_{static} = \frac{1}{t_l - 1} \sum_{g=1}^{t_l - 1} y'_{g,static} \tag{4.10}$$

The purpose behind the introducion of equations 4.9 and 4.10 within the research presented here is to provide another metric for determining the fault resilience of the AfNN method. Consider that, in the previous experiment presented in section 4.3, the focus is strictly on the MSE value during retraining. This is, essentially, a comparison of absolute MSE achievable at various levels of damage. However, damage can also be simply viewed as a restriction to the internal structural capacity of an MLP as opposed to hollistically measuring loss of accuracy. Considering this, there is no basis of comparison to determine whether a regular MLP of a particularly sized hidden layer would be able to achieve a higher or lower MSE and, in turn, whether or not the AfNN method is actually under or over performing with respect to the MSE levels reached. In light of this, we redefine hypotheses 1 and 2 (section 2.4, page 34) again as follows.

Hypothesis 9 *Networks which utilize the affordability method will exhibit a smaller total average MSE as levels of retraining increases, relative to statically structured MLPs, measured using equations 4.10 and 4.5.*

Hypothesis 10 *The more retraining that occurs after onset of damage the lower the average delta error lost per neuron, as compared to a statically structured MLP, measured using equations 4.9 and 4.4.*

This experiment is designed to show that the AfNN method exhibits fault resilience as evidenced by a lower MSE when compared to a statically structured network of the same size. In other words, as the AfNN networks are damaged (after initial training) they perform as well or better than a network of the damaged AfNN size, thereby providing equal or greater value post-damage.

4.4.2 Simulated Results

Table 4.5 depicts the results of the current experiment. The results from experiment 4.2 are repeated here for ease of comparison. The reason for providing both sets of results ($tMSE$, $tMSE_{static}$, $dMSE$, and $dMSE_{static}$) is to better characterize the fault resilience of the AfNN method with respect to hypotheses 9 and 10. To describe it another way, the hypotheses are concerned with how the AfNN method compares against statically structured MLPs of equivalent sizes and, relative to that, whether the AfNN method, at some point, loses enough value to warrant the use of statically structured MLPs or not.

		$tMSE$			$dMSE$		
		10	100	1000	10	100	1000
Iris	Classic	0.0198	0.0143	0.0133	0.0047	0.0035	0.0033
	Random	0.0182	0.0156	0.0136	0.0028	0.0025	0.0024
	Chaotic	0.0216	0.0154	0.0127	0.0032	0.0023	0.0020
	Sequential	0.0208	0.0147	0.0137	0.0027	0.0021	0.0024
Servo	Classic	0.0178	0.0136	0.0138	0.0028	0.0016	0.0024
	Random	0.0187	0.0168	0.0148	-0.0004	0.0021	0.0032
	Chaotic	0.0187	0.0178	0.0147	-0.0004	0.0027	0.0032
	Sequential	0.0158	0.0136	0.0133	0.0001	0.0010	0.0021
Balance	Classic	0.0344	0.0288	0.0311	0.0083	0.0081	0.0074
	Random	0.0269	0.0263	0.0239	0.0014	0.0027	0.0013
	Chaotic	0.0294	0.0272	0.0264	0.0041	0.0029	0.0032
	Sequential	0.0344	0.0278	0.0266	0.0079	0.0067	0.0057
CCPP	Classic	0.0033	0.0035	0.0032	5E-05	5E-05	-6E-05
	Random	0.0045	0.0043	0.0043	3E-05	-2E-05	7E-06
	Chaotic	0.0044	0.0042	0.0043	2E-05	-6E-05	3E-05
	Sequential	0.0034	0.0033	0.0033	4E-05	6E-06	2E-05
		$tMSE_{static}$			$dMSE_{static}$		
		10	100	1000	10	100	1000
Iris		0.0518	0.0212	0.0120	0.0001	0.0031	0.0035
Servo		0.0223	0.0187	0.0112	-9E-05	0.0006	0.0009
Balance		0.0990	0.0279	0.0266	0.0008	0.0059	0.0052
CCPP		0.0024	0.0020	0.0020	0.0008	0.0007	0.0007

Table 4.5: $dMSE_{static}$ and $tMSE_{static}$ results against affordability method and data set showing values for three levels of post-damage retraining.

4.4.3 Analysis and Discussion

The analysis and discussion is split into the following sub-sections. First, a discussion regarding hypothesis 9 and the comparison between $tMSE$ and $tMSE_{static}$. Next, a similar description of results regarding the differences between $dMSE$ and $dMSE_{static}$. The goal is not necessarily to determine a "winner" concerning overall error rates or average error differences per configuration, but to better understand how closely the AfNN method mimics a non-damage MLP configuration to further explore fault resilient characteristics of the AfNN method.

4.4.3.1 Statically Structured MLP Averate Total MSE

With regards to all four data sets, the values for $tMSE_{static}$ are decreasing as epochs increase. The only exception is in respect to the CCPP data set where there is no difference between the one-hundred and one-thousand epoch cases. Comparing these results against those presented in section 4.2 (and again here in table 4.5) there exist two attributes of note. First, analyzing how well the affordable networks fair against their statically structured relatives ($tMSE$ vs $tMSE_{static}$). Secondly, understanding whether or not the particular data set vs. epochs of training configurations are consistent between the two measurements with respect to increasing, or decreasing, error rates. For instance, analysis of the Iris data set shows that, for the ten epochs of retraining/training, the AfNN methods outperform the statically structured MLP. This is also true for the one-hundred epoch case. For the one-thousand epoch case, the statically structured MLP out performs, on average, the AfNN method with respect to $tMSE$. This same scenario plays out again with the servo data set. The balance data set exhibits major improvements on the side of the AfNN method for the

ten epochs of retraining case. The one-hundred and one-thousand cases see values on the side of the AfNNs are within 0.001 of the static MLP but not for the classic AfNN. Lastly, the CCPP data set failed to meet the levels of accuracy (shown by a higher total average MSE) for all levels of epochs vs. the static MLP configuration.

4.4.3.2 Statically Structured MLP Average Error Per Neuron Lost

In both the Iris and Servo data sets, the value for $dMSE_{static}$ worsens as epochs increase. However, the value cannot be taken in isolation as this may be more of an indication that, as evidenced by the $tMSE_{static}$, the overall network performance is increasing and, therefore, the range of error across the entire configuration set, and subsequently, the differences between them, is more dramatic. In other words, we can see from $tMSE_{static}$ that the average error is decreasing as epochs increase. The increase in average error per each neuron in the hidden layer indicates that the range of MSE values across all t_l possibilities is larger. This in itself warrants discussion. What is of more importance is a comparison to the AfNN variations from section 4.2 that is represented again in table 4.5. For the Iris data set, the values for $dMSE$ decreased as epochs increased in all but the sequential variant case. On the contrary, the Servo data set exhibited increasing values aside from the classic variation. The question now becomes, are these results indicative of an algorithmic side effect or, perhaps, the data sets tendencies to be trained more effectively at certain hidden layer sizes. As for the Balance data set, and in line with results from the experiment in section 4.3.2, the results are inconsistent and will require further investigation. Lastly, the CCPP data set once again exhibits a capacity to hardly be affected by changes

in number of epochs. This is still believed to be a result of the large number of training patterns in the data set itself but will also require investigation later in this dissertation (see experiment in section 5.1).

4.5 Summary

Chapter 3 introduces the concept of AfNNs and their inherent structural redundancy. The results of experiments therein help to support the affordability method, to a degree. It is clear that AfNNs can train as well as a regular MLP with little added overhead (see section 3.2). It is also evident that the way groups are selected using this method is critical to its training performance and potential retraining post-damage (3.3.2). With a foundation upon which structural redundancy exists the next step in this research is to measure fault resilience.

This is where chapter 4 begins. Our first experiment in section 4.2 investigates the basic premise of fault resilience. Namely, as a network loses neurons how does that affect it's performance with respect to generalizing previously trained responses to various data sets. The results are positive in that the AfNN variants all exhibit some level of a priori knowledge retention. What is not clear is why the Servo data set exhibited inconsistent results and also whether the affordability method provides more, or less, resilience than a regular MLP which is also subjected to removal of neurons.

In section 4.3 the experiment is designed to highlight the benefit and added value of affordability. A measurement is introduced (equation 4.8) which directly addresses the calculation of value added by the affordability method as compared to the classic MLP. By focusing on the difference of potential value

of an AfNN both above and below the afforadbility threshold, it becomes clear that the surplus of neurons in the hidden layer do indeed help to retain accuracy. Once again, however, the questions arises as to whether or not the slight inconsistencies in some of the results presented are due to nuances of the data sets or a fault of the algorithm employed. The direction to be taken, as a result of these inquiries, will be explored in the next chapter.

Finally, the last experiment presented in section 4.4 aims to more accurately describe the level of inherent fault resilience of the affordability method using a comparison of MSE against statically structured MLPs. The intention here is two fold. Firstly, to discover a stronger corollary which will help in determining whether or not the data sets or the algorithms employed are responsible for the fault resilience observable in section 4.3. Second, knowing not only how AfNNs relate to classic MLPs under damage but also how all variants, including the classic MLP compare to statically structured networks without a pre-disposed saliency map, with respect to training convergence and accuracy. The outcome of this first experiment is mostly positive. Indeed, most cases portray a clear benefit of the AfNN method in these regards. Similarly, comparing the results of the experiments in sections 4.3 and 4.4 once again provides evidence of retained saliency through damage and retraining, as was expected from the saliency measurements made in chapter 3.

However, two problems arise. Firstly, the classic variant is also performing well experiment 4.4. There is no benefit to using AfNNs if a regular MLP already exhibits comparable fault resilience without extra affordability calculations. This is answered, primarily, through the existence of fault resilience above the affordability threshold which does not exist in the classic variant; however, stronger evidence of benefit is preferable. Secondly, the servo data set, which depicts

positive results in section 4.3.3, does not perform to expectation as discussed in section 4.2. Specifically, only the classic and sequential variants portray an above 1.0 ratio, on average. Also, across all data sets, the amount of retraining seems to not positively correlate with higher ratios. If the Mean-squared Accuracy (MSA) ratios are comparable to statically structured MLPs then what is the benefit of the affordability method in this regard?

It is also worth discussing the effect that removing the most salient neuron has on the rate of false positives and negatives in relation to network output. False negative cases may occur as a result of not retraining enough or having the removal of neurons lead the network to get stuck in minima, for instance. This is based on the fact that by constantly removing the most salient neurons the networks are not allowed to create a saliency landscape dramatic enough to provie all positive classes and, instead, produce false negatives. False positives are also at risk of being effected by neuron removal in that the removal of neurons could instantly create a false positive case that is not "trained away" through re-training. This subject will not be investigated further in this thesis but merits investigation otherwise.

Chapter 5 takes a different look at the potential value retained within an AfNN under damage. The number of epochs used to reach the levels of accuracy discussed at length in this chapter also tells a story about how a network is performing. Similarly, the use of MSE in itself makes assumptions regarding the disribution of the data sets under test and, therefore, experiments in the next chapter are performed using calculation of entropy. These two experiments will help to further elucidate the gap between a data sets contribution to measurements of fault resilience and the actual capacity of the affordability method to retain value in the presence of damage.

Chapter 5

Measuring Fault Resilience: Epochs and Entropy

At this point in the research the fault resilience of the Affordable Neural Network (AfNN) method has been measured using variations of error calculations with respect to hypotheses 6 through 10. Specifically, chapter 4, and the hypotheses contained therein, exhibit positive yet inconsistent results with resepect to the ability of the AfNN method to sustain damage using four data sets chosen. The current chapter focuses on these shortcomings in two ways.

First, by understanding the number of epochs utilized in the previous experiments it is expected that some of the behaviors observed will be better understood and help to further distinguish the effect of the data set on the analyses presented. Second, the use of entropy as a calculation of network accuracy provides an alternative to Mean-squared Error (MSE), further corroborating the value provided by the AfNN method.

These calculations, whilst focusing on the experimental data presented against

the various configurations and data sets previously in use for chapters 4 and 3, introduce further measurements which are key to measuring and analyzing fault resilience in neural networks.

5.1 Comparing Epochs Required for Effective Retraining

Having established that fault resilience exists within the affordability method, this experiment is designed to further quantify the usefulness of the AfNN method with respect to inherent fault resilience. To do this, we will be using the same experiment configuration as that of section 4.4 (page 89). However, instead of analyzing MSE, the focus will be examining the number of epochs actually used to retrain each affordability variant.

5.1.1 Experiment Design

The primary goal of this experiment can be described using the following example. If it can be shown in the servo data set from the experiment in section 4.4 that both the static and affordable cases are able to train to the target MSE without utilizing all of the allowable epochs, then it cannot necessarily be assumed that the method exhibits the fault resilience expected but, rather, the data lends itself to being trained too easily, thereby skewing the results; providing an inconclusive analysis. This would make sense, given that the experiments from chapter 4 showed positive results in relation to hypothesis 8 (section 4.3, page 84) and therefore, merits investigation.

What this example illustrates is a scenario whereby comparing MSE alone, whether against static variants or across levels of retraining, may not reveal improvement as measured using the various methods presented in chapter 4 because of how the data sets are constructed and, subsequently, how quick it is for a network to meet the target MSE as measured using utilized retraining epochs. The secondary purpose of this experiment is to further quantify the benefit of retraining between onsets of damage. It is expected that the epoch training ceiling positively affects the resilience of the network during further damage. This is captured in hypothesis 4. The reasoning here is purely within the confines of the data gathered thus far and within the constraints of the experiments. Referring to the analysis in section 4.2 it can be seen that, when comparing the one-hundred and one-thousand epochs of retraining values for $tMSE$, the network performance, in some cases, improves before encountering degradation due to a lack of total neurons in the network. This trend, along with the overall results of the experiment in section 4.2 shows the benefit of higher epoch ceilings. Understanding whether or not the number of epochs actually needed is less than the ceiling is paramount to these observations as the results may further exemplify the benefits of the AfNN method.

Further, the retraining rates related to epochs needed for statically structured Multilayer Perceptrons (MLPs) are also of interest to this study. Namely, the experimental results within section 4.4 are predicated on a comparison with statically structured MLPs; it is, therefore, important to understand how those results may be revisited with the results presented here. This expectation is captured by hypothesis 5 (section 2.4, page 35).

5.1.2 Simulated Results

This experiment will diverge from representing MSEs and, instead, depict epochs actually used during retraining in relation to the epoch ceilings used in the experiment. The values presented in table 5.1 represent the number of configurations (one configuration per each neuron lost) in which the network was able to retrain to pre-damage levels within the maximum epochs allowable. The total number of configurations, therefore, will always be one less than the affordability total, as defined in section 3.1 (page 39). For the classic network variants, the number of configurations is seven. For the remaining configurations (random, chaotic, sequential, and static) the maximum is eleven. The epoch maximums are in relation to the various levels of retraining used in previous experiments (i.e. ten, one-hundred, or one-thousand maximum epochs). All configurations are being trained to the specified maximum number of epochs or the target y'_k as specified in table 4.2, whichever comes first. Also, as in previous experiments, the results for each variants are averaged across ten runs. Table 5.1 exhibits the results of the current experiment.

Alongside the rolled-up values for average epochs against all levels of retraining, figures 5.1 through 5.4 are presented to further clarify the behavior regarding how levels of retraining affect the ability for the AfNN method to retrain within the various epoch maximums.

		10	100	1000
Iris	Classic	1	6	6
	Random	0	8	10
	Chaotic	0	8	10
	Sequential	6	9	10
	Static	0	0	0
Balance	Classic	0	1	3
	Random	0	0	1
	Chaotic	0	0	1
	Sequential	0	1	4
	Static	0	0	0
Servo	Classic	0	6	10
	Random	0	5	10
	Chaotic	0	5	10
	Sequential	7	10	10
	Static	0	0	10
CCPP	Classic	7	7	7
	Random	11	11	11
	Chaotic	11	11	11
	Sequential	11	11	11
	Static	0	0	0

Table 5.1: Number of configurations for which the specified network type and data set configuration was able to retrain within the epoch maximums.

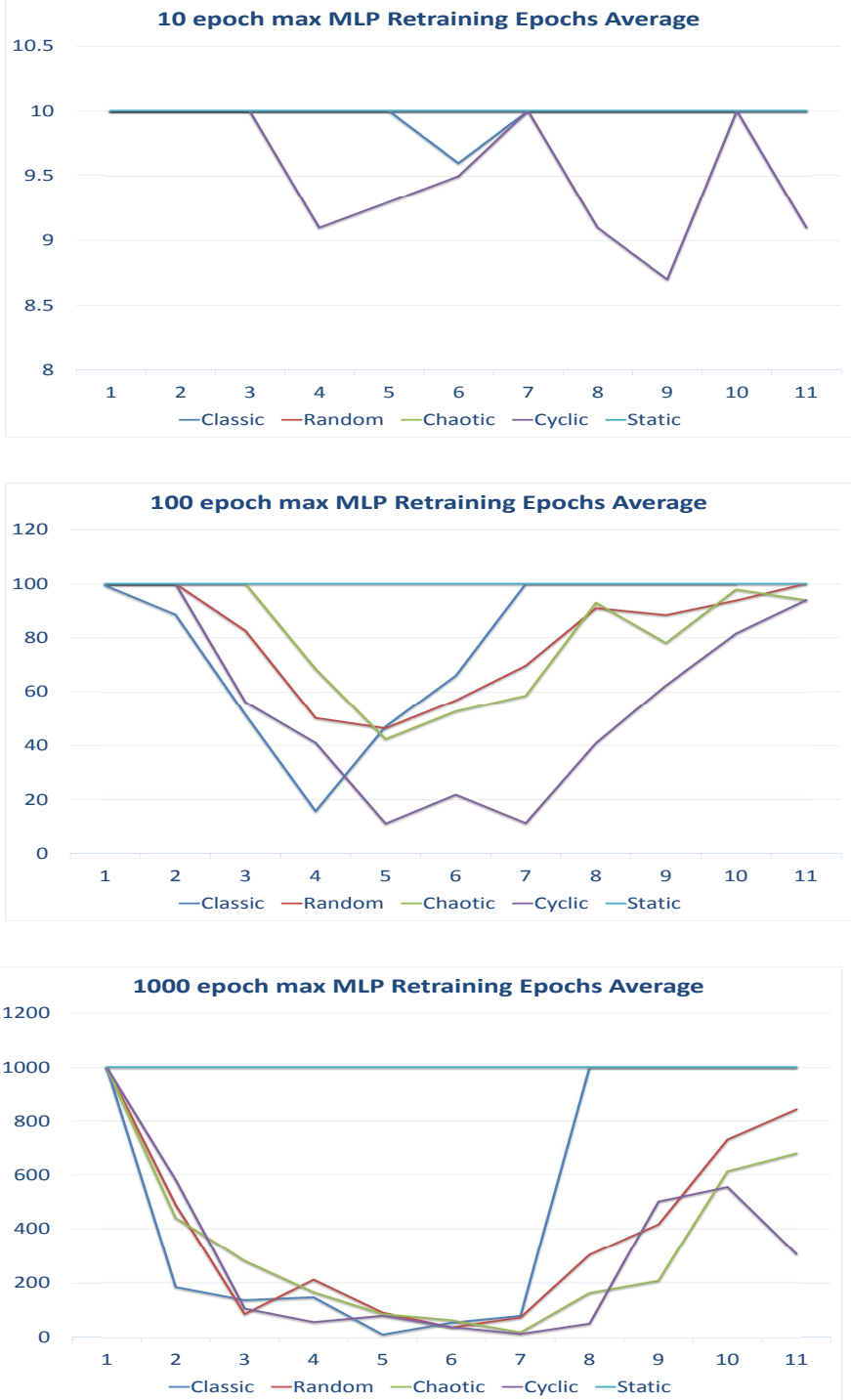
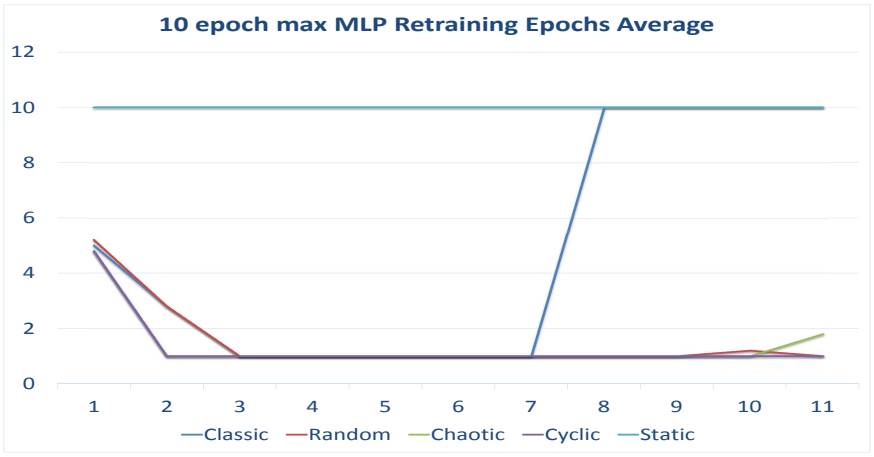
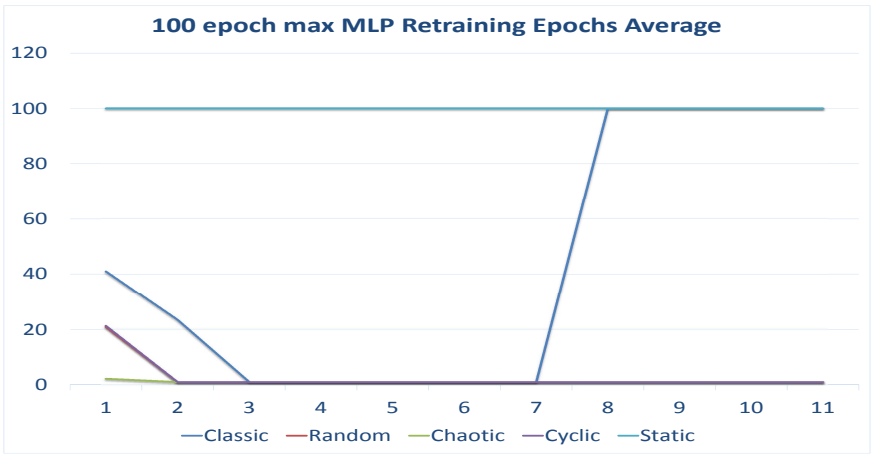


Figure 5.1: Comparison of epochs used across four AfNN variants using the Iris data set during damage retraining (10, 100, and 1000 epochs).

Figure 5.2: Comparison of epochs used across four AfNN variants using the Balance data set during damage retraining (10, 100, and 1000 epochs).

Figure 5.3: Comparison of epochs used across four AfNN variants using the Servo data set during damage retraining (10, 100, and 1000 epochs).





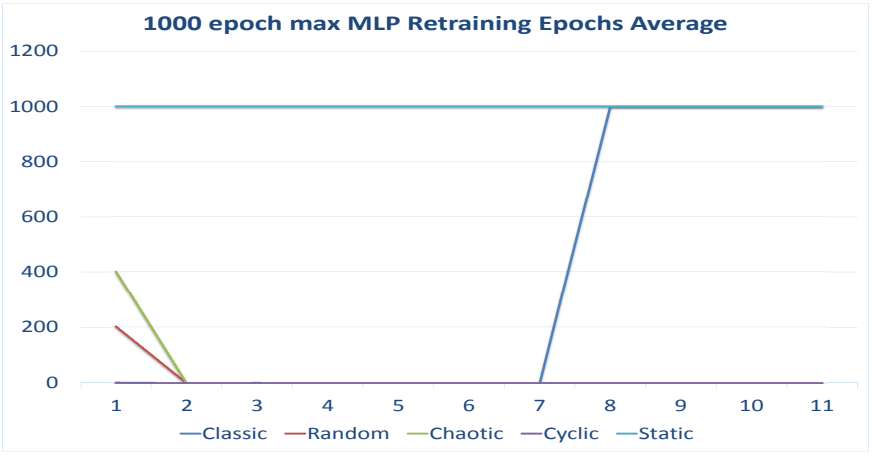


Figure 5.4: Comparison of epochs used across four AfNN variants using the CCPP data set during damage retraining (10, 100, and 1000 epochs).

5.1.3 Analysis and Discussion

As described in section 5.1.1 and summarized by hypotheses 4 and 5 the expectations are two-fold. First, the higher level of retraining that occurs, the better the network will perform as more damage is encountered. This is measured using MSE, as in previous experiments and, likewise, the data presented in section 5.1.2 depict this in showing how many epochs were actually utilized, in all scenarios, to reach the target MSE post-damage. Second, in order to further explore the failures related to previous experiments, it is worth determining whether those failures are on the part of the affordability algorithm or the data set itself. In this way, recording the epochs used at varying levels of damage help to make this distinction.

The most interesting and relevant finding, considering the results presented for the current experiment, relates to that of the statically structured MLPs. In all cases but one, the static structures require all of the epochs alloted to meet (or come close to) the target MSE. This, coupled with the observation that the AfNN variants were able to retrain in less epochs, is a continuation of the method described with respect to the existence of measureable fault resilience. What this means is that, against a statically structured MLP of the same size as a damaged AfNN, the AfNN contains a level of fault resilience which, in some cases, exceeds training a network from scratch (e.g. using a statically structured network) measured by epochs needed to meet target y'_k. Also, because the statically structured MLPs and the AfNN variants use the same data at all sizes (i.e. neurons in the hidden layer) the impact of the data set on the results is effectively ignored. In other words, using fewer epochs to retrain, on average, as compared to statically structured MLPs are truly indicative of retained value as

a result of the affordability method and not how well a data set can be learned. The discussion now moves onto a detailed comparisons against each data set to understand the differences between fault resilience inherent to the MLP against that of the AfNN. It should be noted, however, that in all cases any level of retraining above the affordability threshold indicates fault resilience that would not exist within the classic AfNN variant. This is because the classic configuration, by definition, utilizes zero affordability, per section 3.1.

5.1.3.1 Iris Data Set

In the first instance, an immediate conclusion can be drawn from the static versus maximum epoch percentages strictly within the parameters of this experiment. Namely that, per the values presented in table 5.1, the static variants use all available epochs in all cases, as mentioned earlier in this section. This is to say that the AfNN outperforms the static MLP constructs in all configurations. The value added by the AfNN method is that starting from scratch (e.g. with a statically structured MLP), at any of the hidden layer size variations tested, the AfNN method is able to train back to pre-damage levels within the epoch ceiling. This, in and of itself, is one of the core tenants of fault resilience as captured back in chapter 4 with hypothesis 7. The relationship here is one of understanding that target MSE is achievable in fewer epochs. Moving onto a more homogeneous comparison of AfNN variants, table 5.1 depicts an interesting trend in relation to the expectations related to hypothesis 5. Specifically, as the epoch ceiling is raised, the number of AfNN variants able to retrain to target MSEs without utilizing the entire epoch window also raises from seven at ten epochs to thiry-six at one-thousand. This is an indication that the more

retraining that occurs between onsets of damage positively affects the ability for the AfNN method to retrain following further damage. The classic AfNN configuration, in this regard, performed the worst in two of the three epoch window sizes, acheiving the pre-ceiling retraining target only six times at both one-hundred and one-thousand epoch ceilings. The random, chaotic, and sequential variants performed similarly between one-hundred and one-thousand epochs of retraining, coming in at around eight and ten, respecitvely. For the ten epochs of retraining case, the sequential variant depicts a much higher value added compared to the other variants at this level. Looking at the graphs in figure 5.1 reveals where the value added is placed within the previously discussed results.

5.1.3.2 Balance Data Set

The results relating to the Balance data set are not positive with respect to hypotheses 4 and 5. Whilst the retraining epoch ceiling increase does correlate to increase of cases where networks are able to retrain within the epoch ceiling, the increases are negligible (less than or equal to one). Also, specifically with respect to hypotheses 5, whilst the statically structured MLP was unable to retrain within the specified windows even once within all cases, the classic and sequential variants were only able to do so once and the random and chaotic only once for the one-hundred and one-thousand cases, respectively. Overall, the indiciations here by the data presented for this experiment depict a data set for which retaining value betwen damage is difficult. Whether this is because of the data set itself or whether it is a fault of the algorithm is still to be investigated and, as such, is the topic of chapter 6 whereby the Balance data set is analyzed

and relevant experiments re-run.

For the ten epoch case there was no value distinguishable between any of the networks being compared. In fact, all variants, including the statically structured MLP, were unable to retrain within the epoch ceiling during damage retraining. For the one-hundred and one-thousand cases the classic and sequential variants provided some extra value between four and eight neurons remaining, as depicted by figure 5.1.2. The random and chaotic variants also provided some fault resilience in the one-thousand epoch case around seven neurons remaining per table 5.1. This is not entirely surprising given the results related to hypothesis 8. It is expected that this behavior is related to the same cause which leads to those results.

5.1.3.3 Servo Data Set

Per the results presented in table 5.1, and with respect to the statically structured MLP the Servo data set acts much like the Iris in all cases but one. The one-thousand epochs of retraining exhibited the only case whereby the statically structured network was able to train to the target MSE in less than the maximum number of epochs. This instance provides the only case where retraining of an AfNN variant did not provide better value than creating a statically structured network for all cases above seven neurons remaining as depicted by figure 5.1.2. The classic, random, and chaotic AfNN variants, per table 5.1, all followed nearly the exact same value increases alongside epoch ceiling increases; values of roughly zero, to five, to ten across the board for all three. The sequential variant performed the best overall with respect to the values in the afrorementioned table of results, particularly in the one-thousand and one-hundred cases where

the configuration was able to retrain within the epoch ceiling ten out of eleven times in both cases.

All configurations, including the statically structured MLP, as mentioned earlier in this section, were able to retrain over ninety percent of the time within the epoch ceiling.

5.1.3.4 CCPP Data Set

This data set is unique in that, because the number of training patterns is so large, every AfNN variant (including the classic) is able to retrain within one epoch in all cases as depicted in figure 5.4. The statically structured MLP, however, still require all epochs in training towards the target MSE at all levels, per the results presented in table 5.1. Also, as seen in the aforementioned table, out of the AfNN variants the classic performed the worst with a seven across all epoch ceiling configurations. The random, chaotic, and sequential AfNN types were able to retrain within the epoch ceilings, at all levels, one-hundred percent of the time. This is a resounding positive for the ability of the AfNN method to retain value throughout damage and retraining, despite the lack of positive trending of values as expected in our hypotheses.

5.2 Comparing Entropy of Neurons

Fault resilience measurements, thus far, include comparisons of MSE, epochs required for retraining within three maximum levels, and comparison of accuracy against statically structured MLPs which have sustained no damage. All three experiments support, in some way, the expectation that the AfNN method

provides a level of fault resilience above and beyond what is normally present within a typical MLP under damage. Whilst there exist some negative results with respect to expected behaviors, the evolution of further experiments help to investigate these failings in more detail. In particular, if the failings thus far need to be understood under the scope of whether the affordability algorithm or the data sets themselves are at fault.

The next step in the experimental investigation is to understand how well each AfNN is able to learn the data sets under test. Whilst MSE is a common way to measure error in training, works by (Cun et al., 1990a) remind that magnitude does not equal saliency. If the goal is to accurately determine how well a network is encoding a data set then entropy is the more accurate measurement. In other words, use of entropy as opposed to MSE, which makes assumptions about the probability distribution of the data sets themselves, will more accurately describe saliency of each neuron.

As mentioned in section 4.1, all data sets utilize the same values for affordability total and target despite what would be optimal for the data set. Results for previous experiments portray this in the way that each data set is trained to various levels of efficiency and benefit from a varying number of neurons in the hidden layer. Use of entropy, in this instance, should help to further understand how much the results thus far are tied to data sets vs. the algorithms presented.

5.2.1 Experiment Design

As a network trains post-damage the MSE improves inconsistently across all data sets. If the AfNN method and associated algorithms actually provide fault resilience then a measurement of entropy should also show improvement during

retraining. Much like previous experiments, the data is organized into a table depiciting the values for both the total average entopy, $t\hat{H}(E)$ (eq. 5.5), and the average entropy difference, $d\hat{H}(E)$ (eq. 5.4), both defined below, alongside each data set and AfNN variant in the study.

The calculation of entropy comes directly from works by (Silva et al., 2005) as follows:

$$\hat{H}(E) = -\frac{1}{N}\sum_{n=1}^{N} log\hat{f}(e(n)) \tag{5.1}$$

where E is the error (difference) random variable. Also, our estimation of error is not based on analysis of the data sets distribution. As we don't know the distribution of the error variable, we must rely on nonparametric estimates. For the estimation of $f(x)$ we use the nonparametric kernel estimator

$$\hat{f}(e(n)) = \frac{1}{Nh}\sum_{l=1}^{N} K(\frac{e(n) - e(l)}{h}) \tag{5.2}$$

where h is the smoothing parameter of the standard Gaussian kernel K given by

$$K(x) = \frac{1}{\sqrt{2\pi}} exp(-\frac{1}{2}x^2) \tag{5.3}$$

Given these functions we will now revisit and modify equations 4.4 and 4.5 from section 4.2 as follows.

$$d\hat{H}(E) = \frac{1}{t_l - 1}\sum_{g=2}^{t_l} \hat{H}_g(E) - \hat{H}_{g-1}(E) \tag{5.4}$$

Where $\hat{H}_g(E)$ represents the value of $\hat{H}(E)$ for the single output neuron of the network when g neurons remain in the hidden layer. Finally, the total averaged entropy across all neurons lost is represented as follows.

$$t\hat{H}(E) = \frac{1}{t_l - 1} \sum_{g=1}^{t_l - 1} \hat{H}_g(E) \tag{5.5}$$

As this is a measurement of entropy, a value of zero for $\hat{H}(E)$ implies that the probabilities of the data set outcomes are fully characterized by the network and, therefore, the network can predict results with one-hundred percent accuracy. Similarly, a high value for entropy implies that the values being given to the network for training, and the expected results compared against, are not encoded with any level of predictability or, rather, have an equal probability of occuring from the perspective of the network.

The expectations of this experiment are very similar to those within section 5.1.1. The more a network retrains the better it performs but, this time, measured through entropy. We expect the following hypotheses to hold true. First, a redefinition of hypothesis 1 (section 2.4, page 34).

> **Hypothesis 11** *Networks which utilize the affordability method will exhibit a smaller total average entropy as levels of retraining increases, measured using equation 5.5.*

Similarly, redefinition of hypothesis 2 stated in 2.4, page 34 follows,

> **Hypothesis 12** *The more retraining that occurs between onsets of damage the lower the average entropy lost per neuron, measured using*

equation 5.4.

5.2.2 Simulated Results

The results presented herein are formatted similarly to those within section 4.2 due to the similarities in value derivation and applicability. Per hypotheses 11 and 12 table 5.2 is presented. The data therein presents is directly related to both equations 5.4 and 5.5 against each of the data set and affordablility method combinations.

5.2.3 Analysis and Discussion

As the experiment presented here in section 5.2 mimics that of section 4.2, our results will be concerned not just with how $t\hat{H}(E)$ and $d\hat{H}(E)$ increase or decrease in relation to our expectations captured in hypotheses 11 and 12, but also with how the behavior and trends of the results herein mimic those of the latter experiment. In other words, do the calculations of entropy mimic those of MSE such that the total and average values over all configurations and data sets within each experiment portray a similar set of results? Likewise, the analysis and discussion presented here will be broken down in two sections; the first being in regards to the total averaged entropy (eq. 5.5) and the second in relation to the calcuation of average entropy lost (or gained) per neuron lost (eq. 5.4).

		$t\hat{H}(E)$			$d\hat{H}(E)$		
		10	100	1000	10	100	1000
Iris	Classic	0.4960	0.4620	0.4573	-0.0420	-0.0318	-0.0304
	Random	0.5162	0.4831	0.4595	-0.0190	-0.0107	-0.0142
	Chaotic	0.5337	0.4746	0.4568	-0.0145	-0.0144	-0.0128
	Sequential	0.5118	0.4756	0.4593	-0.0019	-0.0123	-0.0168
Servo	Classic	0.4739	0.4559	0.4366	-0.0151	-0.0149	-0.017
	Random	0.5077	0.4751	0.4425	0.0006	-0.0004	-0.0025
	Chaotic	0.5072	0.4782	0.4399	0.0002	-0.0019	-0.0023
	Sequential	0.4712	0.4408	0.4311	-0.0044	-0.0067	-0.0091
Balance	Classic	0.5383	0.5162	0.5189	-0.0212	-0.0155	-0.0300
	Random	0.551	0.5373	0.5294	-0.0032	0.0002	-0.0029
	Chaotic	0.5592	0.5411	0.5127	-0.0007	0.0007	-0.0024
	Sequential	0.5691	0.5443	0.5008	-0.0011	-0.0040	-0.0029
CCPP	Classic	0.3543	0.3545	0.3561	-0.0100	-0.0112	-0.0108
	Random	0.3703	0.3679	0.3718	0.0002	-0.0006	0.0014
	Chaotic	0.3698	0.3693	0.3698	-0.0001	0.0005	-0.0003
	Sequential	0.3603	0.3581	0.3591	-0.0033	-0.0037	-0.0032

Table 5.2: $d\hat{H}(E)$ and $t\hat{H}(E)$ results against affordability method and data set showing values for three levels of post-damage retraining.

5.2.3.1 Total Average Entropy

With respect to hypothesis 11 and the calculation of $t\hat{H}(E)$ for each data set and AfNN combination holds true. To clarify, all cases presented in table 5.2 with respect to the calcuation of $t\hat{H}(E)$ exhibit decreasing values associated with higher epoch retraining ceilings. These results do *not* match those within section 4.2 in that, within the previous experiment, the balance and servo data sets

were inconsistent and not strictly decreasing in error in all cases. This outcome, however, is not entirely unexpected. Recall that in section 5.2 it is noted that the calculation of entropy makes no assumptions regarding the distribution of information within the testing and training sets. Rather, it aims to efficiently capture saliency of the information, rather than optimize against error magnitude. However, this does not mean that the values of entropy contradict those of MSE nor are they without fault. The kernel estimation is an assumption made during this calculation which provides its own assumptions (eq. 5.3). On the contrary, the values presented in this experiment help to supplement the analysis of results from section 4.2 by providing another dimension of fault resilience measurement. In particular, the abilities for the various data sets to be efficiently trained against and, subsequently, retain value during damage, are revisited here.

It should be noted that these networks within the current experiment are still being trained and evaluated, with respect to early termination and target accuracy, using MSE. The calculation of $\hat{H}(E)$ is performed on top of those for MSE, not in place of. In other words, despite not meeting the expectations of hypothesis 6 in section 4.2, the same experimental configuration for which we also calculate $\hat{H}(E)$ depicts a network which *is* benefitting from epoch ceiling increases as measured using equation 5.5.

As for any caveats to the above conclusions, it should be noted that whilst the CCPP data set did not present strictly increasing value retained as the epoch ceiling increased, the values themselves are so neglibly different (going from an average of 0.364 at ten epochs, to 0.363 at one-hundred, and back to 0.364 at one-thousand) that they are viewed by the authors of the research presented here as having no difference in the grand scheme. In fact, it can be ascertained that the

CCPP data set, as exhibited in earlier experiments, is simply easier to trained against due to the large number of training patterns and characterizations of the data itself.

5.2.3.2 Entropy Per Neuron Lost

The reason we also view entropy lost (or gained) per neuron removed during damage and retraining is not to corroborate the findings related to calculation of total average entropy (reference eq. 5.5, page 114) but, instead, to measure how the total average entropy changes throughout the experiment. Interestingly, the Iris data set, which has provided positive results in nearly every experiment, fails to meet the expectations of hypothesis 12. This is to say that the saliency lost per neuron removed against the Iris data set increases (i.e. saliency is lost, per neuron removed, as the epoch ceiling increases) throughout the experiment in all AfNN variants. This does not mean value is not retained and, therefore, fault resilience is lacking. Again, this data set is proven reliable in previous experiments. The conclusion to be drawn here is that the data set itself is leading to this condition of failure. This is a conclusion that can only be drawn after having looked at all of the experimental results, and therefore the measurements of fault resilience made, throughout this dissertation. If anything, this result strengthens the use of these measurements as a means to hollistically calculate fault resilience of an MLP.

The Servo data set provides positive results with respect to hypothesis 12. The Balance data set, however, does not follow the hypothesis in all cases. For the classic configuration, the value for $d\hat{H}(E)$ increases from -0.0212 to -0.0155, before decreasing again at the one-thousand epoch case to -0.0300. Similarly,

the random and chaotic variants increase then decrease across the three epoch levels. The chaotic variant was different in that it decreased from -0.0011 to -0.0040 before increasing to -0.0029. The Balance data set also portrays the most dramatic changes against this measurement throughout the three epoch levels amongst all of the data sets presented.

Lastly, the CCPP data set, much as it has with respect to $dMSE$, does not portray strictly decreasing values for $d\hat{H}(E)$ in table 5.2. The difference here is, the results are no longer negligible. For instance, the values for $d\hat{H}(E)$ across all levels of retraining for the random AfNN variant changed, on average, 0.0006. This is not far off from that of the chaotic AfNN in relation to the Iris data set which depicts an average change of 0.0009. Certainly, the average change, per data set and AfNN combination, across the entire experiment, is just 0.0019 with values ranging from 0 to 0.0075. Therefore, the CCPP data set also fails to meet the expectations of hypothesis 12.

5.3 Summary

Chapters 3 and 4 lay a foundation of measuring fault resilience, using the AfNN method, and its variations described therein, as a medium. Section 4.5 summarizes the results that lead to this chapter. In particular, section 4.2 quantifies fault resilience within the AfNN method by measuring how MSE changes as both the epoch ceiling increases and neurons are removed. Section 4.3 builds upon that by measuring, specific to the AfNN method, how much value the affordability threshold provides. Lastly, section 4.4 compares previously captured and presented MSE results against those of statically structured MLPs, the con-

trol within an experiment of constantly shifting hidden layer sizes and selection methods.

Coming out of those experiments we have a set of measurements, solely based on MSE in one form or another, which aim to quantify and understand just how much value is gained and retained by the AfNN in the presence of faults. Along this journey it is clear that results are not consistent across all data sets and AfNN variants. Specifically, each data set and variant are expected to provide unique measurable results per the hypotheses presented so far in the research presented here. On the contrary, a set of hypotheses are constructed throughout these experiments to capture expected generic behaviors against specific measurements under the assumption that fault resilience exists. Also, these hypotheses are, primarily, met with positive outcomes.

Coming into chapter 5 the goal is to two-fold. First, to further corroborate the results within the previous chapter using measurements other than those based on MSE. Second, to help to understand the outcomes within the previous chapter which did *not* strictly meet the hypotheses defined therein. Further, these measurements are based on the same trials used in the previous chapter, i.e. the trial construction and executions are still based on MSE as to keep the results relatable between experiments.

Section 5.1 looks at the epochs actually used within the previously run experiments in chapter 4. The aim is to understand whether results based on MSE were effected by early termination of retraining due to meeting the target MSE early. Furthermore, in any instance where this has occured, then the goal is understanding exactly how often and whether or not this behavior can help to further understand fault resilience. Indeed, hypotheses 4 and 5 are constructed to capture the expectations related to this behavior. Namely, as the epoch ceil-

ing increases, and networks are given more epochs with which to retrain, further damage is more readily sustained as depicted by subsequently lower epoch utilization as more neurons are removed. In practice, it is clear that this measurement does provide a valuable datum with which to understand fault resilience and, specifically, the Balance data set, which provids inconsistent results in the previous chapter, also fails to meet hypotheses 4, 5, 11, and 12.

Finally, section 5.2 mimics the experiment in section 4.2, but using a calculation based on saliency as opposed to error, in order to provide more dimensionality to what is described in this earlier section as the most common definition of fault resilience, as per section 4.2. Indeed, it is expected, and captured in hypotheses 11 and 12 that, as damage is sustained and the networks are retrained, a measureable change in entropy should exist and that measurement should depict systems which are able to retain and reinforce saliency as retraining occurs between onsets of damage.

Moving on from these two experiments the conclusion is that the existence of fault resilience is further quantified and understood using the measurements of epochs used and entropy. However, in line with the same negative results from chapter 4, the Balance (and to some degree the Servo) data set consistently exhibits the lack of ability to retain value and, therefore, exhibit fault resilience.

A common theme throughout the research presented here is to understand whether or not the algorithm (AfNN) or the data are the cause of our results. The next chapter in this research is designed to look at exactly what, if anything, within the data sets may be contributing to the cases where results do not meet expectations. In doing so, characteristics of the data sets will be evaluated and, where needed, changed and experiments re-run to understand the effect of changing said characteristics. It is expected that this exercise will finally help to

distinguish, with respect to the few questions remaining, where and how the data set is responsible for measureable fault resilience and where the AfNN method takes credit.

Chapter 6

Further Analysis of Data Set Effects on Fault Resilience

At this stage of the research presented here, nearly all observations provide positive corollary to the claims made regarding the AfNN methods ability to exhibit fault resilience within the data sets and constraints of the experiements herein. However, a few observations remain.

The purpose of this chapter is not to necessarily improve the outcomes presented previously by increasing descreasing error, entropy, and epochs needed for recuperation but, rather, to investigate how the make-up of the training and test configurations related to the Servo and Balance data sets may have contributed to the preceding results. In doing so, the goal is to understand how the data itself affects the measurement of fault resilience thus far. To do this, the experiments presented herein will analyse and modify the training and test setups, recording the change in results after re-running experiments presented previously in this dissertation. Furthermore, the ability to affect the results of previously calcu-

lated fault resilience measurements within this chapter will help to validate the use of the novel metrics presented as part of this dissertation.

6.1 The Balance Scale Data Set

The first step taken in this experiment is to review the steps taken, thus far, to generate our training and test sets for previous experiments. The Balance Scale data set, also known throughout this research as the Balance data set, comes from the University of California, Irvine Machine Learning Repository (UCI) repository (Bache and Lichman, 2013). Information for the four input attributes, and the output classification, are as follows (duplicated from table 4.1, page 70):

- Class Name: 3 possibilities (L, B, R)
- Attribute 1: Left-Weight, 5 possibilities (1, 2, 3, 4, 5)
- Attribute 2: Left-Distance, 5 possibilities (1, 2, 3, 4, 5)
- Attribute 3: Right-Weight, 5 possibilities (1, 2, 3, 4, 5)
- Attribute 4: Right-Distance, 5 possibilities (1, 2, 3, 4, 5)

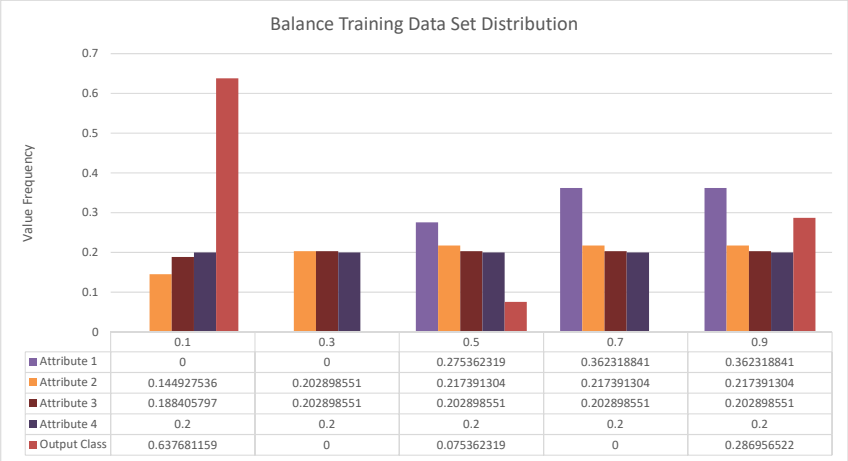
The listed class name and attribute values represent the raw data contained within the balance data set prior to any normalization or alteration. Normalization, per algorithm 1 during the "Normalize Data Set" step, is performed by turning each attribute and the output classification to numeric values, ranging from 0.1 to 0.9. Then, using a simplistic three to one division whereby, given the state of the data set from the UCI repository, the data vectors are chosen sequentially and placed into training and testing sets by first moving three to training and another to test until all data is allotted. This method relies heavily

on the organization of the raw data set itself. The resulting normalized values were as follows:

- Class Name: (0.1, 0.5, 0.9)
- Attribute 1: (0.1, 0.3, 0.5, 0.7, 0.9)
- Attribute 2: (0.1, 0.3, 0.5, 0.7, 0.9)
- Attribute 3: (0.1, 0.3, 0.5, 0.7, 0.9)
- Attribute 4: (0.1, 0.3, 0.5, 0.7, 0.9)

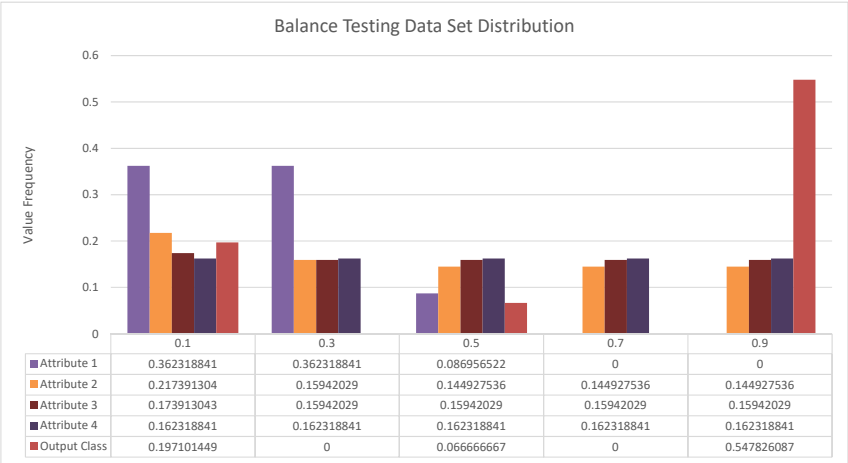
The reason for introducing this level of detail is towards understanding the analysis performed on the test and training sets. Specifically, an analysis of how each attribute and the output classification were represented between the testing and training sets shows a large discrepancy. Please see figures 6.1 and 6.2. These figures provide a visual representation of each attribute distribution against each output class. In other words, for each possible output class, the applicable density represents the skewness of each attribute. Per the definition of data set skewness presented by (Liu, 2009), an imbalanced data set is one in which features, or attributes, are over or under represented in comparison to the output classifications.

Within the aforementioned figures, it is provided, per the value frequency against each input attribute and output classification, that not all value are represented consistently between the training and testing sets. In particular, attribute 1 and the output class require redistribution to be consistently represented. The first two possible values for attribute 1, the values of 0.1 and 0.3, are nonexistent in the training set. Conversely, representation of values 0.5 and 0.9 are omitted from the testing set for attribute 1. Similarly, the training set is over-representing



	0.1	0.3	0.5	0.7	0.9
Attribute 1	0	0	0.275362319	0.362318841	0.362318841
Attribute 2	0.144927536	0.202898551	0.217391304	0.217391304	0.217391304
Attribute 3	0.188405797	0.202898551	0.202898551	0.202898551	0.202898551
Attribute 4	0.2	0.2	0.2	0.2	0.2
Output Class	0.637681159	0	0.075362319	0	0.286956522

Figure 6.1: Distribution of the four input attributes across the Balance data training set.



	0.1	0.3	0.5	0.7	0.9
Attribute 1	0.362318841	0.362318841	0.086956522	0	0
Attribute 2	0.217391304	0.15942029	0.144927536	0.144927536	0.144927536
Attribute 3	0.173913043	0.15942029	0.15942029	0.15942029	0.15942029
Attribute 4	0.162318841	0.162318841	0.162318841	0.162318841	0.162318841
Output Class	0.197101449	0	0.066666667	0	0.547826087

Figure 6.2: Distribution of the four input attributes across the Balance data testing set.

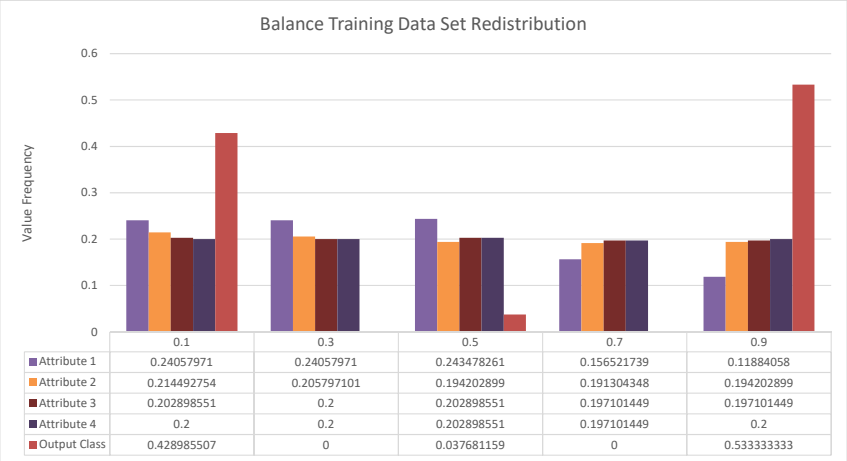
the value of 0.1 for the output class and the testing set is over-representing the value of 0.9.

By redistributing the data so that these two value sets are more consistently represented it is expected that the experiment outcomes for experiments in sections 4.2, 4.3, 5.1 and 5.2, are affected. Figures 6.3 and 6.4 are presented for completeness and portray how the redistribution of Balance data into testing and training sets affected the frequency of possible values for input attributes and output classification. The proceeding sections in this chapter detail the outcomes pertaining to re-running the aforestated experiments.

This experiment is, by no means, an exhaustive optimization of training and testing set generation towards minimizing neural network MSE or epochs required for training. However, any change in the outcome of previously run experiments through alteration of training and testing set generation, particularly positive change, is considered relevant enough for this study against the data set and experimental constructions herein. Consequentially, assuming a change does occur, the findings presented here form a basis for further inquiry related to training AfNNs in future research and how to perform said optimizations.

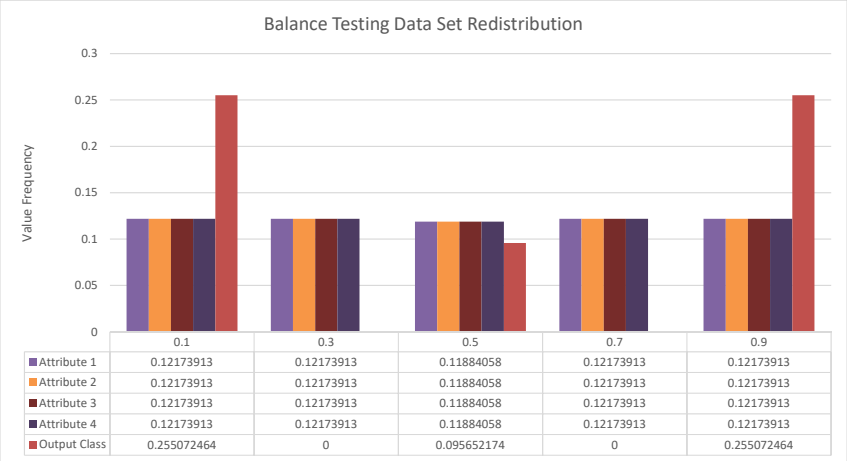
6.1.1 Experiment Design

In this experiment, the hypotheses of previous experiments are revisited in conjunction with a different training and testing data distribution, against the Balance data set. Simulated results tables are also presented, alongisde previous results, for comparison. No changes were made to the AfNN method, the neuron selection processes used during training and testing, nor the measurements



	0.1	0.3	0.5	0.7	0.9
Attribute 1	0.24057971	0.24057971	0.243478261	0.156521739	0.11884058
Attribute 2	0.214492754	0.205797101	0.194202899	0.191304348	0.194202899
Attribute 3	0.202898551	0.2	0.202898551	0.197101449	0.197101449
Attribute 4	0.2	0.2	0.202898551	0.197101449	0.2
Output Class	0.428985507	0	0.037681159	0	0.533333333

Figure 6.3: Distribution of the four input attributes across the Balance data training set.



	0.1	0.3	0.5	0.7	0.9
Attribute 1	0.12173913	0.12173913	0.11884058	0.12173913	0.12173913
Attribute 2	0.12173913	0.12173913	0.11884058	0.12173913	0.12173913
Attribute 3	0.12173913	0.12173913	0.11884058	0.12173913	0.12173913
Attribute 4	0.12173913	0.12173913	0.11884058	0.12173913	0.12173913
Output Class	0.255072464	0	0.095652174	0	0.255072464

Figure 6.4: Distribution of the four input attributes across the Balance data testing set.

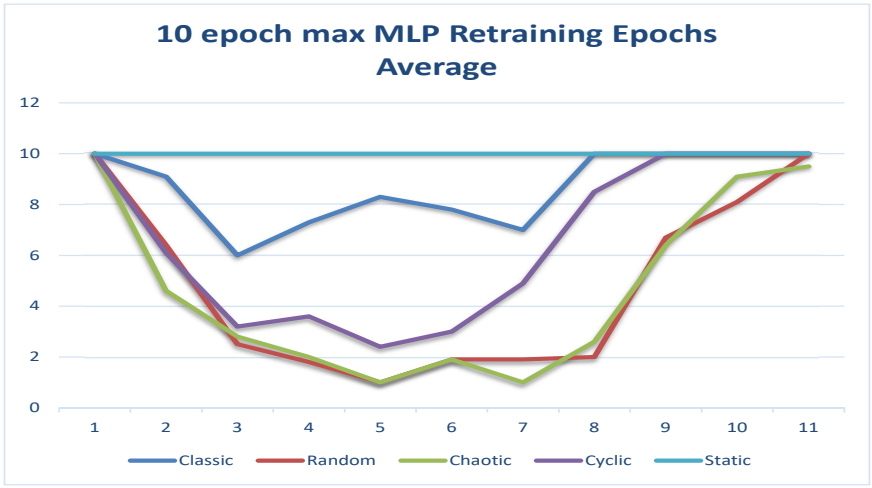
of MSE, entropy, or epochs. The only differences relate to those made to training and testing set construction mentioned above.

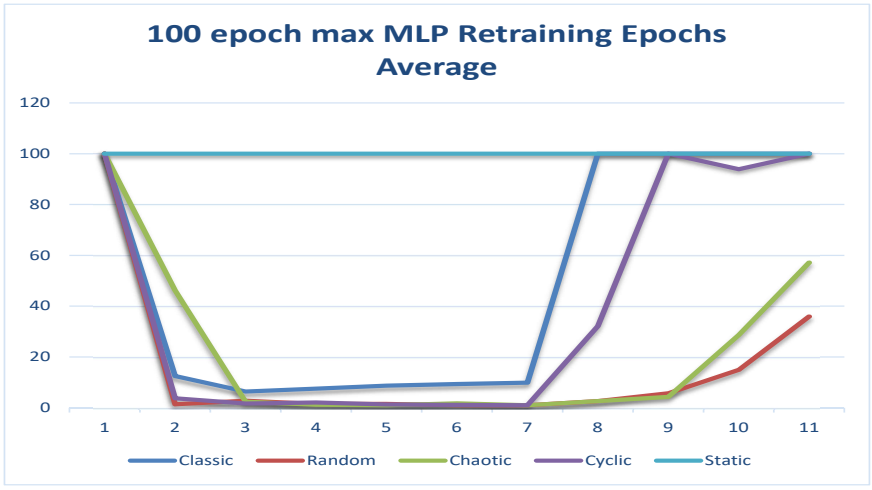
The hypotheses under re-test are as follows: hypothesis 6 (page 74), hypothesis 7 (page 75), hypothesis 8 (page 84), hypothesis 9 (page 91), hypothesis 10 (page 91), hypothesis 4 (page 35), hypothesis 5 (page 35), hypothesis 11 (page 114), and hypothesis 12 (page 115).

In summary, hypotheses 6 and 7 relate MSE and fault resilience. Hypotheses 8 is concerned with ΔMSE. Hypotheses 9 and 10 are in relation to statically structured MLP behavior, measured using MSE against afforadable variants. Hypothesis 4 and 5 are in reference to the number of epochs used during re-training. Finally, hypotheses 11 and 12 relate to the calculation of entropy in measuring fault resilience.

6.1.2 Simulated Results

Results will show the before and after distributions of the Balance data set. Table 6.1 compares results against experiment 4.2 (see page 78) against the new distribution of the Balance data set. Similarly, table 6.2 compares results against experiment 4.3 (see page 85), table 6.3 compares results against experiment 4.4 (see page 92), and table 6.4 compares results against experiment 5.2 (see page 116). Lastly, figure 6.5 depcits the epochs used, per neurons remaining, as compared to the results from section 5.1.2 (see figures 5.1 to 5.4 on page 103).





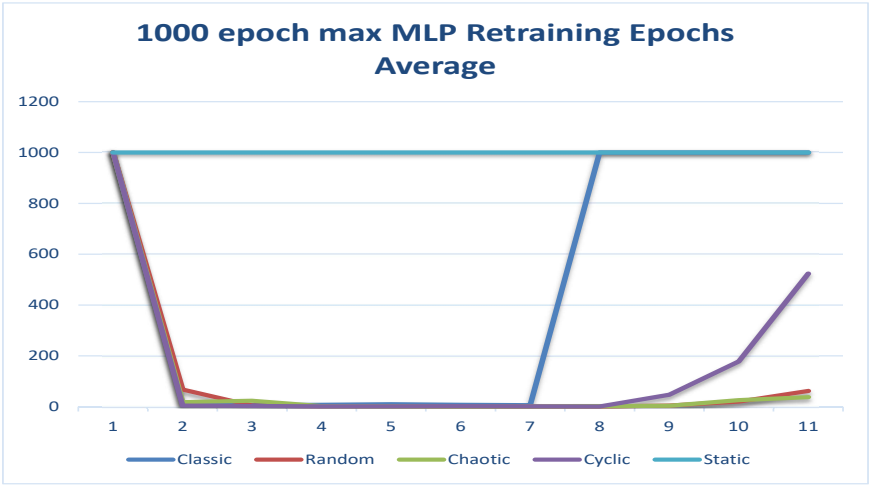


Figure 6.5: Comparison of epochs used across four AfNN variants using the Balance data set during damage retraining (10, 100, and 1000 epochs).

		$tMSE$			$dMSE$		
		10	100	1000	10	100	1000
4.2 Results	Classic	0.0344	0.0288	0.0311	0.0083	0.0081	0.0074
	Random	0.0269	0.0263	0.0239	0.0014	0.0027	0.0013
	Chaotic	0.0294	0.0272	0.0264	0.0041	0.0029	0.0032
	Sequential	0.0344	0.0278	0.0266	0.0079	0.0067	0.0057
New Results	Classic	0.0178	0.0143	0.0140	0.0038	0.0030	0.0027
	Random	0.0143	0.0122	0.0120	0.0012	0.0010	0.0008
	Chaotic	0.0142	0.0134	0.0121	0.0013	0.0009	0.0008
	Sequential	0.0144	0.0131	0.0121	0.0032	0.0028	0.0026

Table 6.1: $dMSE$ and $tMSE$ results against affordability method and experiment revision showing values for three levels of post-damage retraining.

		ΔMSE		
		10	100	1000
4.3 Results	Classic	0.9576	0.9724	0.9709
	Random	0.0031	0.0049	0.0017
	Chaotic	-0.0008	0.0027	-0.0006
	Sequential	0.0023	0.0027	-0.0039
New Results	Classic	0.9784	0.9862	0.9870
	Random	-0.0043	-0.0034	-0.0033
	Chaotic	-0.0042	-0.0047	-0.0034
	Sequential	0.00000	-0.0013	-0.0024

Table 6.2: ΔMSE results against affordability method and experiment revision showing values for three levels of post-damage retraining.

6.1.3 Analysis and Discussion

Table 6.1 depicts results akin to those in section 4.2. The intended outcome of the experiment, as detailed previously, is to have strictly increasing $tMSE$

	$tMSE_{static}$			$dMSE_{static}$		
	10	100	1000	10	100	1000
4.4 Results	0.0990	0.0279	0.0266	0.0008	0.0059	0.0052
New Result	0.0629	0.0146	0.0118	0.0024	0.0032	0.0033

Table 6.3: $dMSE_{static}$ and $tMSE_{static}$ results against affordability method and experiment revision showing values for three levels of post-damage retraining.

		$t\hat{H}(E)$			$d\hat{H}(E)$		
		10	100	1000	10	100	1000
5.2 Results	Classic	0.5383	0.5162	0.5189	-0.0212	-0.0155	-0.0300
	Random	0.551	0.5373	0.5294	-0.0032	0.0002	-0.0029
	Chaotic	0.5592	0.5411	0.5127	-0.0007	0.0007	-0.0024
	Sequential	0.5691	0.5443	0.5008	-0.0011	-0.0040	-0.0029
New Results	Classic	0.4710	0.4513	0.4473	-0.0456	-0.0282	-0.0236
	Random	0.4496	0.4376	0.4333	-0.0162	-0.0161	-0.0146
	Chaotic	0.4471	0.4365	0.4333	-0.0166	-0.016	-0.0145
	Sequential	0.462	0.4441	0.4356	-0.0184	-0.0143	-0.0142

Table 6.4: $d\hat{H}(E)$ and $t\hat{H}(E)$ results against affordability method and experiment revision showing values for three levels of post-damage retraining.

values and decreasing $dMSE$ values as epochs increase. Previously, the results failed to meet this intent for the classic variant, in the case of $tMSE$, and the classic, random, and chaotic variants with respect to $dMSE$. Currently, all values across both metrics now behave as expected. Further, the metrics themselves are lower overall. Due to the relationship of MSE to the calculations themselves, per equations 4.4 and 4.5, this behavior is the result of MSE values having a smaller difference above and below the affordability threshold and the MSE values themselves being lower. As expected, rectifying the imbalance of data in the training and testing sets has not only improved the metrics but, at it's founda-

tion, the ability for the AfNN variants to train against the data.

A similar situation applies to tables 6.2 and 6.3. For calculations against ΔMSE previous failures exist in all AfNN variants such that not all metrics gathered are negative and, therefore, the network is retaining value in relation to the affordability threshold. The current metrics satisfy the expectation of hypothesis 8 in all cases but the sequential ten epoch configuration. However, whilst the value is not negative, it is also not necessarily positive and for the purposes of this analysis does not strictly provide a failing case. $tMSE_{static}$ and $dMSE_{static}$ decreased in overvall absolutely value and now both meet hyptheses 10 and 5. The only outlier with respect to this claim is that of the ten epoch $dMSE_{static}$ case where the value moves from 0.0008 to 0.0024 between the results in section 4.4 and those presented in this chapter. Whilst this is outside the scope of evaluation it is still an interesting point which can be researched further to determine whether this is a direct result of the imbalanced data set or whether this is simply the result of averaging results containined an outlier trial.

Entropy calculations presented in table 6.4 also depict improvement. Mimicking the initial results related to $tMSE$, the classic variant $t\hat{H}(E)$ calculation fails within section 5.2. Against $d\hat{H}(E)$, all variants fail in the aforementioned earlier results. Revisiting hypotheses 11 and 12 are both expected to decrease as time goes one. The recent results utilizing the redistributed Balance data set exhibit a positive outcome for all variants and both measurements, $t\hat{H}(E)$ and $d\hat{H}(E)$. The max absolute range of values for $t\hat{H}(E)$ per epoch level have reduced to 0.0264 compare to 0.0683, both against the sequential variant. This decrease in entropy variability alongside decreased entropy absolute values also supports a conclusion that these affordability variants are able to retain more value during fault utilizing the re-distributed data set.

Finally, figure 6.5 illustrate results more closely relatable to those of the Iris and Servo data sets from the results of the experiment in section 5.1. In other words, within the construction of this experiment, the statically structured congiguration makes use of all available epochs in all cases. Next, the classic variant is utilizes the most epochs otherwise. The chaotic, random, and sequential variants all follow a similar pattern of epochs utilized as the epoch ceiling increases. The chaotic and random variants, in particular, require very few epochs between nine and two neurons remaining, inclusively, with respect to the one-hundred epoch case; the sequential variant only failing to match this at eight and nine neurons remaining. Similarly, the one-thousand epoch case exhibits positive results for these three variants, the sequential being slightly less efficient in this regard, but all much improved compared to the results in section 5.1.

In addition to simply meeting the intention of hypothesis 5 the results from the current experiment are a stark contrast to those presented in section 5.1 where utilization of a number of epochs less than the allowed maximum in all configurations is rare and not once did these experimental constructions require half or less of the allowable limit.

These results, within the scope of the experimental configurations, provides a quantifiable assessment regarding the robustness of the metrics utilzed in conjunction with the hypothetical constructions. Per the changes in experimental results related to a redistribution of the data set and the subsequent conclusions with respect to the relevant experimental hypotheses the metrics are validated insofar as they, firstly, did not change indiscriminantly nor remain the same and, secondly, are altered in a predictable manner. Similarly, the new experimental results against the redistributed data set now conform to the behaviors of experimental results from previous chapters with respect to the other data sets

evaluated.

6.2 Summary

Coming into chapter 6 the novel fault resilient metrics presented in this research provide numerous results with respect to the inherent fault resilience of the AfNN method. Namely, four data sets combined over four affordable variant and epoch limit configurations are tested against metrics derived from MSE, epochs, and entropy. However, the Balance data set in particular fails to meet the expectations related to the various hypotheses associated with the aforementioned metrics. This provides an opportunity for the research presented here to investigate, as well as validate, the ability of these metrics to meet their intended purpose. In other words, in investigating these failing cases against the Balance data set, and subsequently altering the applicable training and testing sets therein, this research is able to demonstrate the value provided by using the fault resilience measurements presented.

Analysis into the training and testing set make-up shows an imbalance of classes. Rebalancing these sets and re-running all relevant experiments leads to the comparison of old and new results as presented in section 6.1.2. The results, in all cases across the duplicated experimental results, now meet the hypotheses designed to capture the related expectations. Specifically, calculations of $tMSE$ and $dMSE$, both against classic variants and statically structured MLPs, now increase and decrease, respective to the calculations, as expected by hypotheses 7, 6, 9, and 10. ΔMSE calculations now meet the intended result per hypothesis 8. Epoch utilization, per hypothesis 5 improved. Finally, $t\hat{H}(E)$ and $d\hat{H}(E)$

values now satisfy hypotheses 11 and 12.

Discovering the imbalance in the training and testing sets and rectifying it carries an expectation that basic MLP training will improve. The success against duplicated experimental metrics, as compared to those presented earlier in this dissertation, in turn, is also expected. Fulfilling this expectation also validates the metrics themselves, corroborating that they are predictably consistent with our assumptions.

Chapter 7

Conclusion

The research presented in this thesis provides a novel and consistent set of measurements of neural network fault resilience utilizing a modified MLP technique, namely the AfNN presented by (Uwate and Nishio, 2005), as a biologically inspired framework on which these measurements can be performed. In doing so, steps are taken towards understanding the inherent fault resilience of the human brain. The incremental discoveries of each chapter are described below.

7.1 Incremental Experimental Summaries

Chapter 3 reviews the AfNN technique and how it can be used as a basis for a structurally redundant and fault resilient MLP with which fault recovery measurements can be made. Therein, the results support the claim that the AfNN method is able to train, similarly to a classic MLP, and produce trained responses. Amongst the AfNN variants the levels of accuracy vary and, subsequently, their inherent fault resilience measurements are affected. It is also

evident, as depicted in figures 3.3 (page 55) and 3.4 (page 55), that the AfNN method provides varying levels of saliency of neurons in the hidden layer amongst the affordability variants. These differences are the catalyst for varying levels of structural redundancy and, therefore, provide the perfect application of fault resilience measurements presented in the next chapter.

Chapter 4 takes the first steps into novel measurements of fault resilience. After expanding the experimental configurations in section 4.1 the first measurement, based on MSE is presented in section 4.2. The outcome of this first experiment provides positive early results which corroborate with the basic training results from the previous chapter. An altered MSE-based measurment is provided in the following section and followed up with a third experiment comparing MSE against controlled MLPs. The outcomes of these experiments all provide reproducible and quantifiable evidence of fault resilience within the AfNN variants and configurations tested. The inability for certain data sets to acheive expected levels of fault tolerance only helps to embolden the techniques presented.

To further diversify and provide credence to the methods presented, chapter 5 introduces new experiments and calculations where entropy and epochs help to further provide more perspective on measureable fault resilience. The results, once again, are consistent and confirm the results presented in previous chapters. In other words, the configurations which acheived higher levels of fault tolerance, as measured using the algorithms presented in this research, maintain positive results with respect to their associated hypotheses against these experiments. Similarly, those configurations which performed poorly in previous experiements still fail to meet the expectations of the hypotheses presented.

The value of measurements is endorsed not only by satisfying experiemental expectations but also when said expectations are not met so long as the metrics

are consistent and reproducible. The last chapter of novel research presented in this thesis, chapter 6, challenges the metrics presented previously in an effort to validate them. By revisiting the generation of data sets this research not only confirms the relationship between generalization error, as is inherent to the derivation of the calculations, but also provides an explanation as to, at least in part, why previous experiements have failed. The change in metrics calculated in conjunction with a change in data set generation is a positive test of the research presented.

7.2 Summary of Contributions

Overall, within the scope of the experiments presented herein, the results support a number of contributions made previously in this thesis. They are as follows:

1 Error-based resilience measurements have been produced and justified, which are designed around the concept of neuronal redundancy (see 4.5, page 95). This use of error-based measurement is novel because it is specific to how network error is affected by loss of structure.

2 Entropy-based measurements have been derived for further measuring saliency of redundant neuronal units (see 5.3, page 119).

3 Metrics for quantifying fault rehabilitation through the use of error, epochs, and entropy, within a number of control settings, have been provided, through experimental results.

4 Analysis on the effects that data set generation, and attribute distribution therein, have on the weight distributions within an MLP and, subsequently,

the fault resilience measurements presented. (see 6.2, page 135).

Given these contributions, the field of fault resilient neural networks can begin to make meaningful comparisons between methods. In other words, of the studies listed in section 2.3 (page 26) which solely utilize a measurement of Root Mean Squared (RMS) for comparing network outputs during damage detection, a more direct metric designed to accomodate the existence of redundancy is now available. Subsequent analyses using these contributed metrics can lead to more relevant discussions regarding the efficacy of fault resilience frameworks. Similarly, previous and future research which utilize detection and replace fault diagnosis technique (active diagnosis and recovery) can also apply these measurements, particularly those related to entropy of redundant units, to augment otherwise ambiguous results regarding removal of neurons and subsequent network behaviors (Chen et al., 1992) (Chu and Wah, 1990) (Bolt, 1992) (George Bolt, 1992). Through more rigourous quantification of fault tolerance in experimental research the solutions therein can be optimized for saliency distribution, rehabilitation cost, and levels of redundancy. These discriminating metrics can help to consolidate an otherwise broad and incomparable field of study, wrought with bespoke solutions and lacking in quantifiable comparison of approaches. The contributions of this thesis are a much needed addition towards fulfilling this next step of fault resilient neural network research.

Chapter 8

Further Research

As this dissertation presents a set of tools it is expected that the application and extension of these methods are far reaching. Potential extensions to this research, in the opinion of the author, can be categorized as either utilization of the metrics or, alternatively, modification of the experiments presented and configurations.

The following sections address potential avenues of further research based on this thesis.

8.1 Amendments and Alterations to Experimentation

One of the more readily available alterations to the current experiments relates to the various configuration selections which are chosen either through trial and error or arbitrarily, as they do not necessarily affect the outcome of the experi-

ments: namely, the affordability thresholds and affordability totals chosen across the various investigations. The metrics themslves are dynamic in this regard and are easily applied to varying configurations.

The selection of data sets is also a natural step to take in expanding this research. Using data sets with similar numbers of input and output parameters is a matter of convenience for the research presented here. However, for the Affordable Neural Network (AfNN) method in particular, new data sets would provide interesting next steps after this research.

Another line of inquiry which follows from the research presented here is whether or not the metrics provided as novel contribution can be expanded upon. Measurements not based on Mean-squared Error (MSE), entropy, and epochs could, in theory, help to augment and generalize those provided here. Indeed, in applying the concepts provided here it may be more appropriate to generate similar metrics specific to the target architecture in a case where a non-Multilayer Perceptron (MLP) configuration is employed.

The effect of neuron removal on false positives and negatives is also a subject brought up in section 4.5 of this thesis that merits further investigation.

8.2 Application of Fault Resilience Metrics

Use of only MLPs within the research presented here is motivated by the extension of previous work regarding the AfNN and providing a meaningful comparison. However, the method itself can and should be applied to different Artificial Neural Network (ANN) architectures. Radial basis functions and support vector machines, due to their similar construction to an MLP are interesting choices.

Similarly, application of these principles to Deep Neural Network (DNN) architectures may help to understand and better utilize a topology that is inherently dynamic and distributed (Baral et al., 2018)(Koutsoukas et al., 2017).

Fault tolerant neural networks utilizing Radial Basis Functions (RBFs) or Support Vector Machines (SVMs) are valid in the field of fault tolerant and self-healing ANNs (Arisariyawong and Charoenseang, 2002). The fault resilience metrics are readily applicable to these types of systems as well.

Neural network pruning techniques, particularly those based on saliency measurements (Cun et al., 1990b)(Zhao et al., 2010), relate to the resilience measurements presented in this thesis. It is worth investigating whether the affordability method could also be used as a method for neurogenesis. With a sufficiently large neuron pool, increasing the affordability target during or post training is comparable to neurogenesis studies by (Michel and Collard, 1996)(Jin and Cheng, 2011)(Jin, 2010) without needing a method for neuron replacement or generation. Further, increasing the number of participating neurons using the affordability method could be optimized to discover optimal hidden layer sizes for given data sets.

8.3 Analysing and Optimizing Data Set Features

Chapter 6 aimed at revisiting one data set with a view to alter the results of previous experiments. This experiment is predicated on the assertions of previous research regarding the distribution of weights within neural networks being a direct consequence of data set attribute sampling (Liu, 2009)(Manoel

Fernando Alonso Gadi and Mehnen, 2010). In further research it is possible to perform more thorough analysis as to other statistical aspects of training data sets and whether or not the fault resilience metrics contributed by this research can be used in optimizing such characteristics.

8.4 Building a Strictly Passive Fault Tolerant Neural Network

Finally, it is the personal interest of the author that the introduced metrics and AfNN descriptions be extended to include active self-healing elements utilizing Intelligent Agent (IA) principles in the similar fashion to how neuronal selection is described. Such a system could make measurements of saliency and error as the neuronal level and, potentially, autonomously evaluate themselves against peer units to determine self importance. An ANN framework comprised of IA neuron units which can autonomously self-select and self-replicate based on their own isolated fault resilience measurements can lead to scalable and passive fault adaptive systems.

Works that focus on find-and-replace fault recovery paradigms, which subsequently suffer from an inability to detect faults without temporal cost (Jin and Cheng, 2011)(Chen et al., 1992), can be reimagined using the metrics contributed by this dissertation as a basis of IA behaviors.

Bibliography

A. Ahmadi, M. H. Sargolzaie, S. M. Fakhraie, C. Lucas, and S. Vakili. A low-cost fault-tolerant approach for hardware implementation of artificial neural networks. In *2009 International Conference on Computer Engineering and Technology*, volume 2, pages 93–97, Jan 2009. doi: 10.1109/ICCET.2009.204.

M.M. Al-Zawi, A. Hussain, D. Al-Jumeily, and A. Taleb-Bendiab. Using adaptive neural networks in self-healing systems. In *Developments in eSystems Engineering (DESE), 2009 Second International Conference on*, pages 227–232, 2009. doi: 10.1109/DeSE.2009.55.

R.A. Andersen, M.H. Schieber, Nitish Thakor, and G.E. Loeb. Natural and accelerated recovery from brain damage: Experimental and theoretical approaches. *Pulse, IEEE*, 3(2):61–65, 2012. ISSN 2154-2287. doi: 10.1109/MPUL.2011.2181093.

S. Arisariyawong and Siam Charoenseang. Dynamic self-organized learning for optimizing the complexity growth of radial basis function neural networks. In *Industrial Technology, 2002. IEEE ICIT '02. 2002 IEEE International Conference on*, volume 1, pages 655–660 vol.1, 2002. doi: 10.1109/ICIT.2002.1189980.

Joel T. Ausonio, William Holderbaum, and Richard J. Mitchell. *Towards Optimizing the Selection of Neurons in Affordable Neural Networks*, pages 254–260. CSREA Press, 2014. ISBN 1-60132-274-7.

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Chitta Baral, Olac Fuentes, and Vladik Kreinovich. *Why Deep Neural Networks: A Possible Theoretical Explanation*, pages 1–5. Springer International Publishing, Cham, 2018. ISBN 978-3-319-61753-4. doi: 10.1007/978-3-319-61753-4_1. URL `https://doi.org/10.1007/978-3-319-61753-4_1`.

S. Bettola and V. Piuri. High performance fault-tolerant digital neural networks. *IEEE Transactions on Computers*, 47(3):357–363, Mar 1998. ISSN 0018-9340. doi: 10.1109/12.660173.

George Ravuama Bolt. *Fault Tolerance In Aritificial Neural Networks*. dissertation, University of York, 1992.

G Bugmann, P Sojka, M Reiss, M Plumbley, and JG Taylor. Direct approaches to improving the robustness of multilayer neural networks. In *Artificial Neural Networks, 2: Proceedings of the 1992 International Conference on Artificial Neural Networks (ICANN–92), Brighton, United Kingdom, 4–7 September, 1992*, pages 1063 – 1066, 1992. doi: 10.1016/B978-0-444-89488-5.50049-X. URL `http://epubs.surrey.ac.uk/839795/`.

M. A. Bujang, P. A. Ghani, N. A. Zolkepali, M. M. Ali, T. H. Adnan, S. Selvarajah, and J. Haniff. Modification of systematic sampling: A comparison with a conventional approach in systematic sampling. In *2012 International Con-*

ference on Statistics in Science, Business and Engineering (ICSSBE), pages 1–4, Sept 2012. doi: 10.1109/ICSSBE.2012.6396525.

Reed D. Clay Carlo H. Seq́uin. Fault tolerance in feed-forward artificial neural networks. *University of California, Berkeley*, 1990.

Chung-Hsing Chen, L.-C. Chu, and D.G. Saab. Reconfigurable fault tolerant neural network. In *Neural Networks, 1992. IJCNN., International Joint Conference on*, volume 2, pages 547–552 vol.2, 1992. doi: 10.1109/IJCNN.1992.226931.

L.-C. Chu and B. W. Wah. Fault tolerant neural networks with hybrid redundancy. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, volume 2, page 639649, 1990.

M. Clergue and P. Collard. Genetic algorithm for artificial neurogenesis. In *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, pages 410 –415, may 1998. doi: 10.1109/ICEC.1998.699790.

Le Cun, Denker, and Solla. Optimal brain damage. *AT&T Bell Laboratories*, 1990a. URL `http://www.cnbc.cmu.edu/~plaut/IntroPDP/papers/`.

Le Cun, Denker, and Solla. Optimal brain damage. *AT&T Bell Laboratories*, 1990b. URL `http://www.cnbc.cmu.edu/~plaut/IntroPDP/papers/LeCunDenkerSolla90NIPS.pdf`.

I. Dalmi, I. Kovacs, I. Lorant, and G. Terstyanszky. Adaptive learning and neural networks in fault diagnosis. In *Control '98. UKACC International Conference on (Conf. Publ. No. 455)*, volume 1, pages 284 –289 vol.1, sep 1998. doi: 10.1049/cp:19980242.

T.R. Damarla and P.K. Bhagat. Fault tolerance in neural networks. In *Southeastcon '89 Proceedings: Energy and Information Technologies in the S.E. 1*, pages 328–31, 1989.

D. Deodhare, M. Vidyasagar, and S. Sathiya Keethi. Synthesis of fault-tolerant feedforward neural networks using minimax optimization. *IEEE Transactions on Neural Networks*, 9(5):891–900, Sep 1998. ISSN 1045-9227. doi: 10.1109/72.712162.

D. Federici. Fault-tolerance by regeneration: using development to achieve robust self-healing neural networks. In *Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on*, volume 5, pages 2808–2813 vol. 5, 2005. doi: 10.1109/IJCNN.2005.1556370.

Gary Morgan George Bolt, James Austin. Fault tolerant multi-layer perceptron networks. *Technical Report: YCS 180*, 1992.

J.L. Hamilton and E. Micheli-Tzanakou. Neural network modeling of memory gradient in alzheimer's disease. In *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, volume 3, pages 1367–1370 vol.3, 1997. doi: 10.1109/IEMBS.1997.756631.

B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal brain surgeon and general network pruning. In *Neural Networks, 1993., IEEE International Conference on*, pages 293 –299 vol.1, 1993. doi: 10.1109/ICNN.1993.298572.

S. Haykin. *Neural Networks: A Comprehensive Foundation*. Macmillan, New York, 1994.

Yuang-Ming Hsu, V. Piuri, and E. E. Swartzlander. Time-redundant multiple

computation for fault-tolerant digital neural networks. In *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on*, volume 2, pages 977–980 vol.2, Apr 1995. doi: 10.1109/ISCAS.1995.519929.

Jinglu Hu and K. Hirasawa. Overlapped multi-neural-network: a case study. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 1, pages 120–125 vol.1, 2000. doi: 10.1109/IJCNN.2000.857824.

H. Ito and T. Yagi. Fault tolerant design using error correcting code for multilayer neural networks. In *IEEE International Workshop on Defect and Fault Tolerance in VLSI Systems*, pages 177–184, Oct 1994. doi: 10.1109/DFTVS.1994.630028.

Zhanpeng Jin. *Autonomously Reconfigurable Artificial Neural Network On A Chip*. PhD thesis, University of Pittsburgh, 2010.

Zhanpeng Jin and A.C. Cheng. A self-healing autonomous neural network hardware for trustworthy biomedical systems. In *Field-Programmable Technology (FPT), 2011 International Conference on*, pages 1–8, 2011. doi: 10.1109/FPT.2011.6132669.

G.N. Karystinos and D.A. Pados. On overfitting, generalization, and randomly expanded training sets. *Neural Networks, IEEE Transactions on*, 11(5):1050–1057, 2000. ISSN 1045-9227. doi: 10.1109/72.870038.

M.A. Khabou and P.D. Gader. Automatic target detection using entropy optimized shared-weight neural networks. *Neural Networks, IEEE Transactions on*, 11(1):186 –193, jan 2000. ISSN 1045-9227. doi: 10.1109/72.822520.

Alexios Koutsoukas, Keith J. Monaghan, Xiaoli Li, and Jun Huan. Deep-

learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of Cheminformatics*, 9(1):42, Jun 2017. ISSN 1758-2946. doi: 10.1186/s13321-017-0226-y. URL `https://doi.org/10.1186/s13321-017-0226-y`.

Kisong Lee, Howon Lee, and Dong-Ho Cho. Collaborative resource allocation for self-healing in self-organizing networks. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, 2011. doi: 10.1109/icc.2011.5962426.

T. Y. Liu. Easyensemble and feature selection for imbalance data sets. In *2009 International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing*, pages 517–520, Aug 2009. doi: 10.1109/IJCBS.2009.22.

Alair Pereira do Lago Manoel Fernando Alonso Gadi and Jorn Mehnen. Data mining with skewed data. In *New Advances in Machine Learning*, pages 173–188, 2010. doi: 10.5772/9382.

O. Michel and P. Collard. Artificial neurogenesis: an application to autonomous robotics. In *Tools with Artificial Intelligence, 1996., Proceedings Eighth IEEE International Conference on*, pages 207 – 214, nov. 1996. doi: 10.1109/TAI.1996.560453.

M.K. Mulligan, Lu Lu, R.W. Overall, G. Kempermann, G.L. Rogers, M.A. Langston, and R.W. Williams. Genetic analysis of bdnf expression cliques and adult neurogenesis in the hippocampus. In *Biomedical Sciences and Engineering Conference (BSEC), 2010*, pages 1–4, 2010. doi: 10.1109/BSEC.2010.5510847.

C. Neti, M. H. Schneider, and E. D. Young. Maximally fault-tolerant neural networks and nonlinear programming. In *1990 IJCNN International Joint*

Conference on Neural Networks, pages 483–496 vol.2, June 1990. doi: 10.1109/IJCNN.1990.137759.

A. Nickelsen, J. Gronbaek, T. Renier, and H. Schwefel. Probabilistic network fault-diagnosis using cross-layer observations. In *Advanced Information Networking and Applications, 2009. AINA '09. International Conference on*, pages 225–232, 2009. doi: 10.1109/AINA.2009.66.

T. Orlowska-Kowalska and M. Kaminski. Effectiveness of saliency-based methods in optimization of neural state estimators of the drive system with elastic couplings. *Industrial Electronics, IEEE Transactions on*, 56(10):4043–4051, 2009. ISSN 0278-0046. doi: 10.1109/TIE.2009.2027250.

D. S. Phatak. Relationship between fault tolerance, generalization and the vapnik-chervonenkis (vc) dimension of feedforward anns. In *Neural Networks, 1999. IJCNN '99. International Joint Conference on*, volume 1, pages 705–709 vol.1, 1999. doi: 10.1109/IJCNN.1999.831587.

D. S. Phatak and I. Koren. Fault tolerance of feedforward neural nets for classification tasks. In *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, volume 2, pages 386–391 vol.2, Jun 1992. doi: 10.1109/IJCNN.1992.226957.

D. S. Phatak and E. Tchernev. Synthesis of fault tolerant neural networks. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, volume 2, pages 1475–1480, 2002. doi: 10.1109/IJCNN.2002.1007735.

M. Rahman, R. Thottappillil, M. Berg, and H. Hillborg. Comment on 'effect of surface charge on hydrophobicity levels of insulating materials'. *Generation,*

Transmission and Distribution, IEE Proceedings-, 149(3):300–304, 2002. ISSN 1350-2360. doi: 10.1049/ip-gtd:20020072.

A. RajaRajan. Brain disorder detection using artificial neural network. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 4, pages 268–272, 2011. doi: 10.1109/ICECTECH.2011.5941901.

Ramachandran and Blakeslee. *Phantoms In The Brain*. HarperCollins, 1998.

R Rivera and B Dawes. Boost c++ libraries, 2014. URL `http://www.boost.org`.

B. E. Segee and M. J. Carter. Comparative fault tolerance of parallel distributed processing networks. *IEEE Transactions on Computers*, 43(11):1323–1329, Nov 1994. ISSN 0018-9340. doi: 10.1109/12.324565.

Ting Shen, Fangyi Wan, Bifeng Song, and Yun Wu. Damage location and identification of the wing structure with probabilistic neural networks. In *Prognostics and System Health Management Conference (PHM-Shenzhen), 2011*, pages 1–6, 2011. doi: 10.1109/PHM.2011.5939524.

Silva, DeSa, and Alexandre. Neural network classification using shannons entropy. In *European Symposium on Artificial Neural Networks*, 2005.

M. Steinder and A.S. Sethi. Probabilistic fault localization in communication systems using belief networks. *Networking, IEEE/ACM Transactions on*, 12 (5):809–822, 2004. ISSN 1063-6692. doi: 10.1109/TNET.2004.836121.

R. Paul Stroemer, Thomas A. Kent, and Claire E. Hulsebosch. Neocortical neural sprouting, synaptogenesis, and behavioral recovery after neocortical infarction

in rats. *Stroke*, 26(11):2135–2144, 1995. doi: 10.1161/01.STR.26.11.2135. URL `http://stroke.ahajournals.org/content/26/11/2135.abstract`.

Yongning Tang, E.S. Al-Shaer, and R. Boutaba. Active integrated fault localization in communication networks. In *Integrated Network Management, 2005. IM 2005. 2005 9th IFIP/IEEE International Symposium on*, pages 543–556, 2005. doi: 10.1109/INM.2005.1440826.

Yongning Tang, Guang Cheng, Zhiwei Xu, and E. Al-Shaer. Community-base fault diagnosis using incremental belief revision. In *Networking, Architecture, and Storage, 2009. NAS 2009. IEEE International Conference on*, pages 121–128, 2009. doi: 10.1109/NAS.2009.24.

Y. Uwate and Y. Nishio. Back propagation learning of neural networks with chaotically-selected affordable neurons. In *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on*, pages 1481 – 1484 Vol. 2, may 2005. doi: 10.1109/ISCAS.2005.1464879.

Y. Uwate and Y. Nishio. Learning process of affordable neural network for back-propagation algorithm. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–7, 2010. doi: 10.1109/IJCNN.2010.5596355.

Y. Uwate, Y. Nishio, and R. Stoop. Scale-rule selection of affordable neural network for chaotic time series learning. In *Circuit Theory and Design, 2007. ECCTD 2007. 18th European Conference on*, pages 811–814, 2007. doi: 10.1109/ECCTD.2007.4529720.

Lifen Yuan, Yigang He, Jiaoying Huang, and Yichuang Sun. A new neural-network-based fault diagnosis approach for analog circuits by using kurtosis and entropy as a preprocessor. *Instrumentation and Measurement,*

IEEE Transactions on, 59(3):586 –595, march 2010. ISSN 0018-9456. doi: 10.1109/TIM.2009.2025068.

M.H. Zadeh and M.A. Seyyedi. A self-healing architecture for web services based on failure prediction and a multi agent system. In *Applications of Digital Information and Web Technologies (ICADIWT), 2011 Fourth International Conference on the*, pages 48–52, 2011. doi: 10.1109/ICADIWT.2011.6041420.

Shouling Zhao, Quan Liu, and Binbin Zhang. A fast obs pruning algorithm based on pseudo-entropy of weights. In *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, volume 3, pages 451–455, 2010. doi: 10.1109/ICACC.2010.5486816.