INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
A real-time neural-net computing approach to the detection and classification of underwater acoustic transients

Hemminger, Thomas Lee, Ph.D.

Case Western Reserve University, 1992
A REAL-TIME NEURAL-NET COMPUTING APPROACH TO THE
DETECTION AND CLASSIFICATION OF UNDERWATER
ACOUSTIC TRANSIENTS

by

THOMAS LEE HEMMINGER

Submitted in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

Thesis Advisor: Professor Yoh-Han Pao

Department of Electrical Engineering and Applied Physics
CASE WESTERN RESERVE UNIVERSITY

January 1992
CASE WESTERN RESERVE UNIVERSITY
GRADUATE STUDIES

We hereby approve the thesis of

Thomas Lee Hemminger

candidate for the Doctor of Philosophy

degree.*

Signed:

[Signatures]

Date 11/20/91

*We also certify that written approval has been obtained for any proprietary material contained therein.
A REAL-TIME NEURAL-NET COMPUTING APPROACH TO THE
DETECTION AND CLASSIFICATION OF UNDERWATER
ACOUSTIC TRANSIENTS

Abstract

by

THOMAS LEE HEMMINGER

Underwater acoustic transients can develop from a variety of sources ranging from the cry of a whale to the sound of a torpedo launch. Accordingly, detection and classification of such transients by automated means can be an exceedingly difficult task. This thesis describes the design and implementation of a new approach to this problem based on adaptive pattern recognition employing neural networks and additional techniques including the Hausdorff metric. This system uses self-organization to both generalize and provide rapid throughput while, in addition, utilizing supervised learning for decision making. The design is based on a concept which temporally partitions acoustic transient signals, and as a result, studies their trajectories through power spectral density space. This method has exhibited a high rate of success for a large set of underwater transients contained in both quiet and noisy ocean environments, and is capable of real-time operation.
Dedicated to
My Family
Acknowledgements

I wish to thank Professor Yoh-Han Pao for all his support and encouragement throughout my tenure as a graduate student at CWRU. Dr. Pao agreed to be my advisor during the fall semester of 1989 and immediately presented me with a great number of responsibilities, revealing his faith and trust in my abilities from the outset. I also wish to thank him for his confidence in me at a time when the difficulties of this project seemed insurmountable. His interest in neural networks is highly contagious and provided the motivation I needed to pursue this discipline. I will always be grateful for his influence on my career.

I also wish to thank Dr. Dejan Sobajic for many hours of discussion, and the guidance he provided when I was preparing to present papers or teach classes. Dr. Sobajic has been very helpful in developing my understanding of neural networks and for this I convey my appreciation.

This thesis is derived from work performed on a project sponsored by the Defense Advanced Research Projects Administration (DARPA). DARPA contracted with Loral Defense Systems of Akron, Ohio which, in turn, subcontracted part of the project to CWRU. I am grateful to all of those persons responsible for making our collaboration with Loral possible and I especially wish to thank two Loral employees, Mr. Dennis J. Adams and Dr. Stuart Clary. Mr. Adams has been extremely helpful in relating many of the requirements of sonar systems. He has also provided a good deal of constructive criticism concerning my algorithms, which have improved many of the performance characteristics. Dr.
Clary has been instrumental in enhancing my understanding of the Hausdorff metric and various other mathematical topics. We have had numerous technical discussions for which I am appreciative.

I also wish to thank those persons on my committee for taking the time and having the patience to read this thesis and provide criticism. They are, in addition to Dr. Pao, Dr. Sheldon Gruber, Dr. Frank Merat, and Dr. Cecil Thomas.

In addition, I want to thank my family. Even though I started college in my mid-20's, my parents William and Eleanor, were very supportive and allowed me to live in their house at no expense for as long as necessary to earn my Bachelor degree. My brother Robert has always been a great source of technical knowledge and we have had many theoretical discussions which have further advanced my education. Finally, I want to give my heartfelt thanks to my wife Sharon who has experienced little else of married life than that which can be viewed from graduate school. She has experienced the highs and lows of this lifestyle to the same extent as myself and has been an integral part to any accomplishment I have achieved as a graduate student. She has stood beside me 100% and has supplied much of the energy that I have needed to attain my goal.
# TABLE OF CONTENTS

Abstract ........................................................................................................ ii

Dedication ...................................................................................................... iii

Acknowledgements ..................................................................................... iv

Table of Contents ........................................................................................... vi

List of Figures ............................................................................................... viii

List of Tables ................................................................................................. xii

Chapter 1  Introduction ............................................................................... 1

Chapter 2  The Ocean Environment ............................................................ 6

2.1 Physical Properties .............................................................................. 6
2.2 Effects of Noise .................................................................................... 7
2.3 Statistical Characteristics of Ocean Noise ......................................... 8
2.4 Characteristics of the Current Data Set .............................................. 10

Chapter 3  Neural Networks .................................................................. 15

3.1 Neural-Net Computing and Pattern Recognition .............................. 15
3.2 Neural-Nets and Pattern Classification ............................................. 17
3.3 Functional-Link Net ............................................................................ 19
3.4 Self-organization .................................................................................. 22
3.5 Supervised Learning ........................................................................... 23
3.6 Unsupervised Learning ....................................................................... 24
3.7 Similarity Measures ............................................................................ 25
3.8 Associative Memory ............................................................................ 26
3.9 Optimization ......................................................................................... 28
Chapter 4  Current Neural-Net Approaches ........................................ 29

4.1  Target Recognition Problem ............................................ 29
4.2  Neural Networks and Target Recognition ....................... 30
4.3  Passive Sonar .......................................................... 32
4.4  An Early System ....................................................... 34
4.5  Current Systems ....................................................... 35

Chapter 5  Neural-Net Approach to Transient Classification ............. 40

5.1  System Description .................................................... 40
5.2  Feature Selection ..................................................... 45
5.3  Preprocessing .......................................................... 46
5.4  Trajectories .......................................................... 48
5.5  Clustering and Prototypes .......................................... 49
5.6  Trajectories and the Functional-Link Net ....................... 54
5.7  Theoretical Analysis ................................................ 58

Chapter 6  Experimental Results ........................................... 68

6.1  Prototypes ............................................................ 68
6.2  Network Training .................................................... 71
6.3  Run-time Evaluation ................................................ 72
6.4  Run-time Considerations .......................................... 86

Chapter 7  Discussion and Conclusion .................................... 88

7.1  Discussion ............................................................ 88
7.2  Conclusion ........................................................... 90

References ................................................................. 92

Appendix ................................................................. 98
LIST OF FIGURES

Chapter 2

Figure 2.1 Quiet ocean noise .............................................. 12
Figure 2.2 Noisy ocean ...................................................... 13
Figure 2.3 Histogram of quiet ocean ................................. 14

Chapter 3

Figure 3.1 General feedforward network .......................... 18
Figure 3.2 Functionalities of a neuron ................................. 19
Figure 3.3 The Functional-Link Net ................................. 20
Figure 3.4 Hopfield network .............................................. 28

Chapter 4

Figure 4.1 Block diagram of an ATR system ...................... 30
Figure 4.2 Basic passive sonar configuration ...................... 32
Figure 4.3 Blackman-Tukey indirect method of spectral analysis .... 33
Figure 4.4 Direct method of spectral estimation .................... 34
Figure 4.5 Block diagram of discrete wavelet transform .......... 36
Figure 4.6 Transient classification using wavelet preprocessing ... 37
Figure 4.7 Eight channel time-frequency decomposition .......... 38
Figure 4.8 Transient classification using windowed FFT ........... 39

viii
Chapter 5

Figure 5.1 Temporal trajectories through feature space .......................... 41
Figure 5.2 Simple similarity function ..................................................... 41
Figure 5.3 Neural networks that recognize trajectories ....................... 43
Figure 5.4 System architecture .............................................................. 44
Figure 5.5 Short-time Fourier transform .................................................. 47
Figure 5.6 Three spectra as vectors using 2-norm ......................... 50
Figure 5.7 Power spectra and the Hausdorff metric ...................... 52
Figure 5.8 Effects of noise on patterns .................................................... 53
Figure 5.9 Data prewhitener .............................................................. 54
Figure 5.10 Cluster trajectories .............................................................. 55
Figure 5.11 FLN and adjusted Hausdorff distances ..................... 57
Figure 5.12 Spherical interpretation of signal detector .................... 57
Figure 5.13 Irregular surface interpretation of signal detector ............ 57

Chapter 6

Figure 6.1 Parameters used in cluster algorithm ......................... 69
Figure 6.2 Separation of data records into network inputs ............... 72
Figure 6.3a Class A transient in quiet ocean ................................. 73
Figure 6.3b Class A transient in noisy ocean ................................. 73
Figure 6.4 ROC curves for class A classifiers at SNR = 2.0 dB ....... 74
Figure 6.5  ROC curves for class A classifiers at SNR = 1.0 dB ...... 75
Figure 6.6a Class B transient in quiet ocean................................. 77
Figure 6.6b Class B transient in noisy ocean .............................. 77
Figure 6.7  ROC curves class B N-N classifier at SNR = 1.0 dB ...... 78
Figure 6.8a Class C transient in quiet ocean................................. 79
Figure 6.8b Class C transient in noisy ocean .............................. 79
Figure 6.9  ROC curves for class C classifiers at SNR = 2.0 dB ...... 80
Figure 6.10 ROC curves for class C classifiers at SNR = 1.0 dB ...... 81
Figure 6.11a FM chirp ............................................................ 83
Figure 6.11b FM chirp in low level noise .................................... 83
Figure 6.11c FM chirp in noisy ocean ......................................... 83
Figure 6.12 ROC curves for FM chirp classifiers at SNR = 2.0 dB ... 84
Figure 6.13 ROC curves for FM chirp classifiers at SNR = 1.0 dB ... 85
Figure A1 Class A transients in quiet ocean ............................... 99
Figure A2 Class A transients in noisy ocean .............................. 100
Figure A3 Class B transients in quiet ocean .................. 101
Figure A4 Class C transients in noisy ocean .............................. 102
Figure A5 Biological transient N1 ........................................... 103
Figure A6 Biological transient N2 ........................................... 104
Figure A7 FM chirp in noisy ocean .......................................... 105
Figure A8  Classes A and C, trained on A .........................   106
Figure A9  Classes A and C, trained on C .........................   107
Figure A10  Classes A and C, trained on A, utilizing 2-norm ........ 108
LIST OF TABLES

Chapter 5
Table 5.1  Comparison of preprocessing techniques .................. 45

Chapter 6
Table 6.1  Summary of results using DARPA data set ............... 82
Chapter 1. Introduction

Since the beginning of the computer era researchers have sought to develop computational systems modeled after contemporary theories of biological neural architecture. Recent advances in computer technology and advanced algorithms have helped bring this goal closer to reality by enabling the construction of high speed parallel distributed systems, sometimes referred to as connectionist models, which have been structured to perform simple cognitive tasks. These models of intelligent systems are composed of arrays of elemental processors called neurons which perform low level processing and possess a high degree of interconnectivity [Rumelhart, McClelland 86]. Systems of this type are generally termed artificial neural networks (ANN) due to their overall similarity to biological neural structures. The field of neural networks is receiving growing attention because it is believed they hold the key to an alternate form of data processing which reduces or eliminates the need for serial programming methods. In addition, they continue to show promise in the development of systems which learn by example, are adaptive, and remember by association. The ANN has demonstrated its ability to mimic some cognitive tasks by interpreting the importance and interrelationships of a series of complex input parameters, thus synthesizing discriminants, while satisfying massive constraints. These characteristics can be applied to several fields including computer vision, speech processing, control systems, character recognition, or any other realm dealing with large amounts of distributed information, i.e., patterns. Until recently results in some of these areas have been
largely unsatisfactory due to the restrictions imposed by conventional computing
techniques. But with the development of neural network technology there has been
a resurgence in the optimism that these and similar applications can be realistically
addressed with a high degree of success.

This work is concerned with the application of adaptive pattern recognition
and neural networks to the field of underwater sonar, particularly with regard to
transient signals. Identification of underwater transients can be nearly intractable,
due in part to the variability and potentially large number of signals. These
properties coupled with intra-class variability render most conventional methods of
signal classification unsuitable.

Historically, all acoustic data from sonar systems has been evaluated by
humans through aural means, but with the advent of high speed computers, visual
displays have become available to augment the auditory information. This in turn
has permitted sonar operators to process multimodal data yielding a faster reaction
time by possibly combining it in some way to reduce the detection threshold
[Kobus, et al. 86]. These displays are extremely complex and require that potential
sonar operators go through extensive training as well as memory tests for visual
display and auditory pitch. There is now significant interest in determining the
viability of interpreting these signals by means of parallel distributed systems
without the need for human intervention. An automated system could be used to
augment a human listener during periods of high alert status, or perhaps more
importantly, perform its function during extended uneventful periods when human
performance tends to deteriorate into a condition known as vigilance decrement
[Williams 86].
Sonar can be divided into two disciplines — active and passive. Active
sonar systems transmit acoustic energy into the water and listen for their reflected
echoes, similar in concept to radar. This is frequently done to determine the ocean
dept or look for geographical features as is required in geological surveys. It is
also used by the military to search for enemy vessels and mines, by salvage crews,
and by sport fishermen. The transmitted signal is recognized by most people as the
familiar "ping" operating in the region from 2 to 40 kHz.

Passive sonar stations monitor the acoustic environment without providing
an impinging stimulus for the production of echoes. This affords the advantage of
allowing the observer to remain silent while listening for the sounds of others.
Sonar of this variety is used by the military for underwater surveillance and, due to
its nature, remains primarily security classified.

This thesis is structured in the following manner. Chapter two presents a
brief description of the ocean environment. It is concerned with the methods in
which acoustic signals are propagated and the effects of noise on these signals. In
addition, test results are presented which illustrate some characteristics of ocean
noise and how they compare with those of a set supplied by the Defense Advanced
Research Projects Administration (DARPA)*.

Neural networks are described in chapter three. The discussion is focused
on two of the main paradigms in current use today, supervised and unsupervised
learning and includes some additional views on similarity measures, associative
memory and optimization. Unsupervised learning performs a self-organizing

* This is a data set supplied by DARPA which is used to test the effectiveness of various neural
network architectures as part of the Artificial Neural Network initiative.
function, providing the ability to create generalized descriptions of the input data. This is a crucial element which contributes substantially to the current system's real-time capability. Supervised learning is also an integral component, in that it determines the relative importance of various signal attributes and provides decisions based on experience acquired during a training session as well as those obtained during actual system run time. All supervised learning is performed with an alternative learning paradigm referred to as the Functional-Link Net (FLN).

Chapter four presents a brief look at conventional sonar systems and illustrates some current work being done to interpret, by computer, the data received from passive sonar hydrophones. Much of the research performed at the governmental level is classified, especially those issues concerned with signal processing or conditioning, but there are enough publications available to develop a sufficiently representative background.

Chapter five offers a thorough description of the current system architecture including details of the network learning algorithms along with reasons for choices in preprocessing, primarily as it pertains to consistency and noise. Some of the topics discussed are related to normalization, choices in feature scaling, and the determination of attributes submitted to those networks utilized for decision making. One section provides a description of the Hausdorff metric and how it pertains to this task, while another presents a theoretical interpretation of network performance.

Chapter six presents the experimental results of this work after processing the unclassified set of data provided by DARPA. This also includes samples from an additional set, provided by an alternate source, containing real data similar to
that for which the system has been designed. The results demonstrate the robustness of the method and illustrate its resilience when subjected to adverse environmental conditions.

Chapter seven conveys some conclusions with regard to the present system and outlines thoughts on future designs. The benefits of this methodology as well as a few of its shortcomings are also addressed and should be considered when future designs are investigated.
2.1 Physical Properties

The ocean is a complex signal medium presenting a host of problems to those attempting to communicate from within its domain. As is well known, it is not possible to achieve realistic underwater communication with electromagnetic radiation due to severe attenuation. Consequently, the current approach is to use sound waves which can travel relatively unimpeded for great distances. Certain considerations should be taken into account before designing acoustic communication (listening) equipment, therefore, at this juncture it is illustrative to provide a brief discussion which relates some of the difficulties involved.

The underwater environment is a non-linear transfer medium tending to be multipath in nature [Burdic 91], meaning that sounds emanating from a point source can arrive at the listener highly distorted. Due to these factors, along with others, signals originating from sources of the same class (same source type) can appear nearly unrelated from the point of view of the listener. Additional complicating factors are absorption and ducting. For an active system, a higher emitted frequency translates into greater resolution, but unfortunately the level of absorption of acoustic waves within the ocean also increases with frequency. For this reason the acoustic frequencies are generally limited to less than 40 kHz. From a larger perspective the sea can be modeled as a wave guide in which case sound waves will travel horizontally for very great distances. This is called ducting and can be
attributed to the substantial horizontal dimension involved and the effects of stratification from temperature changes and currents [Oppenheim 78].

2.2 Effects of Noise

In addition to the above, noise from within the ocean plays a very important role in the distortion of underwater acoustic transients. Knowing the type of noise in a communication channel is advantageous since this will usually govern some of the parameters of the system design. This discussion on noise is included for two reasons. The first is to illustrate that the noise characteristics of the ocean, while not usually Gaussian, can occasionally be approximated by a Gaussian or similar distribution in quiet ocean. This provides common ground for comparison with other methods which are frequently tested with Gaussian noise. The second reason for this discussion is possibly more significant, namely to lend some credibility to the test set provided by DARPA. This unclassified set is, for the most part, not a true representation of ocean data since it has been determined (section 2.4) that the background is colored Gaussian noise, and that all signals from a class are basically similar except for their SNRs. But, some sections of the ocean do have characteristics similar to the test set, signifying that the data should be a reasonable representation of a subset of ocean conditions.

With regard to passive sonar systems there are two major sources of noise worthy of concern — self and ambient. Self noise is that which is created by the listener and may include sounds from his own propellers and power plant. Effects from this type of noise can frequently be reduced by shutting down all the noise sources or by using an adaptive noise cancelling filter [Widrow, Stearns 85].
The ambient noise source is of greater concern because it is created by elements beyond the control of the listener. This latter form can result from a multitude of sources of which a few are cited here. Noise in the 1 to 10 Hz range is generally created by turbulence in the ocean caused by tides and seismic activity. The 10 to 300 Hz range of noise is created mostly by shipping, with a large portion of it being composed of tonals due to rotating machinery. Higher frequencies are generally caused by surfaces winds, but tend to diminish at greater depths. Additional noise is created by cracking ice in arctic regions and biologics such as whale cries, snapping shrimp and porpoise whistles [Veitch, Wilks 85]. This introduces an interesting issue on whether some of these sounds should be considered as noise or as targets. The general consensus is — anything which is not currently a target is considered noise whether of biological or mechanical origin, and this is where problems can arise because, for example, the non-Gaussian "noise" from snapping shrimp can be distorting the target signal.

### 2.3 Statistical Characteristics of Ocean Noise

Several researches have studied ocean characteristics and have provided some interesting observations. A brief discussion of some of these findings are presented here and will be of value when the current system is described. In deriving noise models several statistical tools are often useful, such as measures of the mean, variance, skew, and kurtosis, as well as histograms.

As one would expect, ocean noise can be both non-stationary and non-Gaussian particularly when in close proximity of such transient noise sources as cracking ice, biologics, or machinery. In researching the literature several
distributions have proven useful of which two are included here: (1) the Gaussian, and (2) the Gaussian - Gaussian mixture (GGM). The Gaussian distribution is probably the most familiar probability density function and is shown here for completeness:

\[ p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \]  \hspace{1cm} (2.1)

where \( \sigma^2 \) is the variance and \( \mu \) is the mean of the distribution (taken as zero for the current discussion). The GGM is composed of two Gaussian pdfs,

\[ p(x) = (1-\varepsilon)g(x;\sigma_1) + \varepsilon g(x;\sigma_2), \]  \hspace{1cm} (2.2)

and is often an appropriate distribution to fit to noise that is nearly Gaussian, where \( g(x,\sigma) \) represents a zero mean Gaussian of standard deviation \( \sigma \). Bouvet and Schwartz [88] have categorized two types of background noise according to their pdfs. The first discussed here is produced by merchant ships. This type of noise remains relatively stationary throughout the observation period with a skew near zero as for a Gaussian pdf but with a kurtosis of around two (kurtosis of a Gaussian is three). The low kurtosis can be explained by the presence of tonals from the machinery which have a kurtosis of 1.5. The distribution has been found to fit the GGM quite well with one strong Gaussian component and another being a weak perturbation.
Background noise without the presence of biologics or machinery can also be characterized by a stationary GGM, but here it is much closer to a true Gaussian.

Other types of noise are produced by cracking ice in the Arctic regions, ship propellors, biologics, and many other sources — all contributing to the overall background. Veitch and Wilks [85] have shown that some noise can be closely modeled by a continuous spectrum (frequently Gaussian) with the addition of tonal bursts which temporally change in spectral position. They have also shown that their distribution remains nearly Gaussian even in areas of high variance. This observation agrees with the discussion above where Bouvet and Schwarz make similar inferences about the ambient background noise.

2.4 Characteristics of the Current Data Set

At this point it is appropriate to relate the type of data analyzed with the current system. For testing purposes, DARPA supplied 33 minutes of unclassified acoustic sequences having well behaved noise characteristics with which to evaluate the system described in this thesis. The test data is composed of six classes of transient signals consisting of complex signatures and tonals acquired at 25 kHz and stored as 16 bit integers. Several of the original signals are redundant, therefore, additional classes were added including an FM chirp to better interpret the efficacy of the system. Details of all signals tested are located in chapter six. The background noise is divided into two classes, quiet and noisy ocean. Measurements of both types indicate that noisy ocean is approximately 6 dB greater in average power than quiet ocean.
To further characterize the noise the mean, variance, skew, and kurtosis were generated from several seconds of quiet and noisy ocean from the DARPA set. These parameters were calculated by dividing the data into record lengths of 256 samples, computing the statistics on the sample sets, and plotting the results against the record number. The $j^{th}$ record mean is given by:

$$m_j = \frac{1}{256} \sum_{i=0}^{255} x_{ji}$$

(2.3)

and the central moments by:

$$m_j^{(a)} = \frac{1}{256} \sum_{i=0}^{255} (x_{ji} - m_j)^a$$

(2.4)

where the assumption is made that the data are stationary within the record and that the bias is relatively insignificant considering the record size. The sample skew is a measure of the asymmetry of a distribution given by $S_j = m_j^{(3)} / (m_j^{(2)})^{3/2}$ and the sample kurtosis is an indication of the fatness of the tails: $K_j = m_j^{(4)} / (m_j^{(2)})^2$.

These tests were performed by Veitch and Wilks [85] on the quiet ocean background and demonstrated near Gaussian behavior for certain areas of ocean noise. Figures 2.1 and 2.2 illustrate the behavior of the DARPA data in both quiet and noisy ocean with relation to these measures where it is noted that the skew is very close to the Gaussian value of 0 as is the kurtosis at a value near 3.

These assumptions are further substantiated by approximating the probability density function by means of a histogram which was constructed from quiet ocean,
but is representative of both noise types. A true Gaussian curve is superimposed over the plot illustrating, in figure 2.3, the nature of the data. Thus it is apparent, with a high degree of certainty, that the noise supplied for test purposes is Gaussian and that it can be similar, at times, to actual ocean data.

**Figure 2.1** First four central moments of quiet ocean noise. These parameters were calculated by dividing the data into record lengths of 256 samples, computing the statistics on the sample sets, and plotting the results against the record number. Using 1600 records it is evident that the skew and kurtosis are similar to those of a Gaussian distribution being 0 and 3 respectively.
Figure 2.2 First four central moments of noisy ocean. The processing is identical to that used to create figure 2.1, but here the variance has increased.
Figure 2.3 Histogram of normalized quiet ocean noise from the test set superimposed on exact Gaussian probability density function. This histogram exhibits strong similarity to a plot by Veitch and Wilks [85] for quiet ocean noise.
Chapter 3. Neural Networks

At this point it is illustrative to review some of the aspects of neural computing and to describe a few current algorithms in use today. This field is concerned with studying system designs based on current understanding of the brain and invariably consist of distributed networks of simple processors which learn by example and exhibit a high degree of adaptability. There are currently four neural network paradigms, (1) supervised learning, (2) unsupervised learning, (3) associative memory, (4) optimization. This work is concerned primarily with the first two topics, therefore, most of this chapter will be devoted to them.

In addition to the four functionalities related above, neural computing systems have three general features [Mammone, Yehoshua 91]: (1) they consist of many simple processing elements, (2) these units possess a high degree of interconnectivity, (3) they use iterative techniques to update the interconnections between the processing elements. Each of these features is also discussed on the following pages.

3.1 Neural-Net Computing and Pattern Recognition

One goal of neural-net computing is to assimilate vast quantities of distributed information (patterns), and from them, make inferences about new information heretofore unseen. Problems of this nature arise in image processing, speech recognition, control systems, and additional areas where data is distributed spatially as well as temporally. Early researchers in the field such as Minsky,
Rosenblatt, and Widrow developed single layer networks called perceptrons [Rosenblatt 59], but these soon proved insufficient at solving anything other than linearly separable problems, i.e., those that can be separated by hyperplanes in feature space. The difficulties associated with the perceptron have for the most part been eliminated with the development of the semilinear feedforward network, and efficient training algorithms as presented by Rumelhart, Hinton, and Williams [86]. These systems have several layers of interconnected non-linear processors which allow the synthesis of complex decision boundaries. They have been used and studied extensively and have demonstrated a high degree of versatility [Widrow et al. 88], [Waibel et al. 89].

The task of pattern recognition with neural networks can be generalized to that of mapping an n-dimensional pattern vector $X \in \{R^n\}$ to an m-dimensional output vector $Y \in \{R^m\}$ while satisfying constraints derived from a set of input-output patterns; being accomplished through linear and non-linear transformations. There are also intermediate spaces within the hidden layers of the network itself, the dimensions of which, should be considered in the design. For example, if the dimensionality of the hidden layer is small compared to the input, information can be lost and fault tolerance reduced. Alternatively, if the dimension of the hidden region is too large it can be difficult for the network to generalize, and problems with noise can develop [Pao 89a], [Mammone, Yehoshua 91].

This mapping is something that must be learned. Neural networks are non-linear systems, therefore matrix theory cannot, in itself, produce the complete solution. This was recognized when Rosenblatt [62] was experimenting with his perceptron, but no algorithms existed for training a non-linear multi-layer net.
With the formulation of the generalized delta rule (GDR) it became possible
to train networks with hidden layers [Rumelhart et al 86]. The GDR is a well
known method based on gradient descent, being an extension of the method used to
train flat nets called the delta rule (DR).

3.2 Neural-Nets and Pattern Classification

The inherent strength of neural structures is derived from their non-linear
processes. These functions allow for the synthesis of discriminants, enabling the
system to separate input patterns which are intermeshed spatially. Without the non-
linear functions the entire system would collapse to a single layer matrix
transformation.

Figure 3.1 illustrates a common feedforward neural network configuration.
It consists of an input layer, two hidden layers, and an output layer. Within the
first layer the input vector is multiplied by a matrix of weights producing a
hyperplane decision region. The second layer of the network performs another
linear transformation after each input of this stage has been modified by the non-
linearity. Outputs from this layer describe convex boundaries called hyperregions
which are able to separate a wide class of input groups. Although one hidden layer
is generally sufficient [Cybenko 89], [Cotter 90] [Hornik 86], many experimenters
employ an additional layer to allow for arbitrary decision regions. Some feel that
this will increase learning speed or improve performance, but this is not always the
case. In fact, it is difficult if not impossible to determine precisely the number of
hidden layers or the number of neurons in those layers which are necessary for a
given task. Nevertheless, research is being conducted in this area with some
success and discussions can be located in [Fogel 91] and [Gutierrez, Wang, Grondin 89].

![Diagram of a neural network]

**Figure 3.1** General feedforward neural network.

The processing elements within a neural network (figure 3.2) are usually sigmoidal, the most common probably being the Logistic function. The argument of this function is the output of the previous layer multiplied by the set of weights in the current layer, with $\theta$ as a bias or threshold term.

$$y = \frac{L_0}{L_0 + e^{-(w^x + \theta)}}$$  \hspace{1cm} (3.1)
Another common function is $y = \tanh(x)$, which is simply a scaled logistic. Other functions include the threshold logic unit, limited ramps, and exponentials. The latter are used in probabilistic neural networks which are designed to approach the Bayes-optimal decision boundaries as the number of training samples is increased [Specht 90].

![Diagram of a neuron](image)

**Figure 3.2** Functionalities of a neuron.

### 3.3 Functional-Link Net

This work utilizes a network proposed by Pao [89a], which performs enhancements to the input patterns before they are passed to higher layers of the network. This configuration is called the Functional-Link Net (FLN), and it can, depending on the circumstances, be utilized as a flat net, obviating the need for
hidden layers [Pao 89b]. In this case delta rule training can be employed, having
the substantial benefit of decreasing the training time by orders of magnitude over
that needed to train a multilayer net. In the context of this work the FLN is
regarded as a stand alone flat network.

Figure 3.3 illustrates the general structure of the FLN. It consists of a set of
inputs representing the original pattern along with an additional set of enhanced
inputs. These enhancements have the effect of increasing the dimensionality of the
input vector thus providing the hyperplanes of the flat net with a greater ability to
separate the input patterns. Although the original pattern may still exist as an input
(see figure 3.3), the additional enhancements present alternative regions for
discrimination. In this case the augmentations can become the dominant factors,
having significant connection weights, while the weights of non-separable
components are diminished in importance.

![Figure 3.3 The Functional-Link Net.](image-url)
There are two primary models for the design of the FLN. One method is called functional enhancement, meaning that all or some of the inputs are transformed by non-linear functions. Some examples of these functions are \( \sin(n\pi x) \), \( \cos(n\pi x) \), \( \sin(2\pi x) \), \( \cos(2\pi x) \), etc., being subsets of an orthonormal basis set, but many other choices are possible. The second design method is called joint activation. In this procedure the entire input vector is multiplied by its transpose creating an outer vector product, thus taking full advantage of any interrelationships between components of the input pattern. This second method is actually a variation of the functional enhancement model [Sobajic 88]. If the vector is \( n \)-dimensional there will be \( n(n+1)/2 \) individual inputs of which \( n(n-1)/2 \) are enhancements, not including squared terms, but this number can be reduced by pruning those enhancements not considered necessary. Both methods form an augmented representation of the input pattern within a higher dimensional space.

Another distinct advantage of the FLN is that it can be analyzed by employing generalized inverses. For example, consider a set of \( P \) patterns, each represented by a vector \( \mathbf{x} \) having \( K \) attributes. If the dimensionality of the vector is increased with \( N \) enhancements the resultant pattern will have \( K+N \) elements. These are multiplied by \( K+N \) weights \( \mathbf{w} \) forming a dot product. At this point a bias term \( \theta \) is usually added to the summation to permit the activation function to shift on the independent axis if necessary. The result: \( z = \mathbf{w} \cdot \mathbf{x} + \theta \) is input to a non-linear activation function. Without loss of generality this function can be disregarded with attention directed instead at the output \( z \). In this case the system is entirely linear, enabling one to solve for the weights \( \mathbf{w} \) with a matrix inverse.
The bias term can be considered as having a constant input value of 1.0 as shown below:

\[
\begin{bmatrix}
  x_1^{(i)} & \cdots & x_{K+N}^{(i)} & 1.0 \\
  \vdots & \ddots & \vdots & \vdots \\
  x_1^{(p)} & \cdots & x_{K+N}^{(p)} & 1.0
\end{bmatrix}
\begin{bmatrix}
  w_1 \\
  \vdots \\
  w_{K+N} \\
  \theta
\end{bmatrix} =
\begin{bmatrix}
  z^{(i)} \\
  \vdots \\
  z^{(p)}
\end{bmatrix}
\] (3.2)

In general, \( P > K+N \) in which case the coefficient matrix is not square, but the system can be solved by the method of least squares inverses (pseudo-inverses), as shown here:

\[
w = (X^T X)^{-1} X^T z
\] (3.3)

This solution is often unacceptable because of its large error, but this can be alleviated by using the delta rule learning method (section 3.5). Further applications of the FLN are located in [Pao, Sobajic 89].

3.4 Self-organization

Data can often be grouped into labeled subclasses called clusters. This is frequently performed to discover common features in members of the set, thus allowing the creation of representatives of those members called prototypes. These new examples form a reduced data set which can be used to direct any new classification scheme functioning at higher resolution. This procedure is performed without the aid of a teacher and is, therefore, commonly referred to as unsupervised learning.
This paradigm has a self-organizing characteristic in that it is able to
discover and group pattern similarities in a purely autonomous fashion, which is a
valuable methodology because it can reduce the burden on all subsequent processing
by reducing the data overhead, thus increasing overall reliability and precision.
The problem can be stated succinctly as: given a set of patterns P with members \( x_i \),
create a reduced set of pattern representatives C having members \( y_k \), while
following some form of metric criterion. The set C must have fewer members than
the set P. Frequently \( y_k \notin P \), but instead is created by a linear combination of the
input patterns.

3.5 Supervised Learning

Supervised learning is a method with which to *teach* a neural network. This
procedure incorporates the presence of an external teacher that corrects the network
output as training progresses. This produces what may be considered as a
heterassociative mapping, meaning that output responses invoked by cues at the
input need not bear any similarity those cues.

The basic supervised learning paradigm is called the delta rule (DR), which
is nothing more than a gradient descent algorithm which searches for an error
minimum. The generalized delta rule (GDR) is an extension of the DR, but it
propagates the error back through the network in order to determine the weight
adjustments necessary at the lower layers, i.e., toward the inputs. This
methodology is descriptively referred to as *backpropagation of error*. GDR can be
located easily in texts such as Pao [89a] or Rumelhart, Hinton, and Williams [86], so it will not be discussed here.

3.6 Unsupervised Learning

There are several algorithms used for self-organization, and here a few of them are described. Learning Vector Quantization (LVQ) is a nearest neighbor classifier which consists of a two layer net and a predetermined number of processing units, each having a k-element reference vector \( w \) [Kohonen 88], and each being associated with a class from the input set. The algorithm can be stated as follows, using an appropriate metric:

1. Let \( p \) be the processing unit whose vector \( w_p \) is closest to a new input vector \( x_i \). In this case \( x_i \) is assigned to class \( p \).

2. Iteratively update the elements of \( w_p \) in the following manner:

\[
w_p(t + 1) = w_p(t) + \alpha(t)[x_i(t) - w_p(t)]
\]

if \( x_i \) and closest unit belong to same class,

\[
w_p(t + 1) = w_p(t) - \alpha(t)[x_i(t) - w_p(t)]
\]

if \( x_i \) and closest unit belong to different classes,

No change at other processing units.

The factor \( \alpha(t) \) is reduced with time to force convergence.

Geva and Sitte [91] used a variant of this method in their work on adaptive nearest neighbor pattern classification.
Another common clustering algorithm is called ISODATA [Duda, Hart 73]. The basic form of the algorithm is as follows:

1. Choose c initial vectors for the cluster centers.
2. Classify the remaining vectors by assigning them to the closest cluster according to a suitable metric.
3. Compute the mean of each group assigned to a class.
4. If the change in any mean is greater than a vigilance factor go to step 2. Otherwise terminate the procedure.

Details of these and many other methods can be located in [Pao 89a], [Antognetti, Milutinovic 91] or [Kohonen 88].

3.7 Similarity Measures

Cluster algorithms attempt to discover natural groupings by observing similarities in the data. It is, however, necessary to define some form of similarity measurement before one can be certain whether an algorithm has successfully performed its function. Several methods have been used with varying degrees of success but one particularly useful approach based on scatter matrices was studied by Fukunaga and Koontz [70] and is briefly described here:

Let there be N patterns denoted by vectors $X_k$. Let $G_k$ denote the M cluster groups having centers $C_k$.

Define the total scatter of the pattern set as:

$$T \Delta = \sum_{k=1}^{N} X_k X_k^T$$  \hspace{1cm} (3.6)
The scatter within each cluster is defined as:

\[ W_j \triangleq \sum_{x_k \in \mathcal{C}_j} (X_k - C_j)(X_k - C_j)^T \]  
(3.7)

where:

\[ C_j = \frac{1}{N_j} \sum_{X_k \in \mathcal{C}_j} X_k \]  
(3.8)

The total scatter from within the clusters (intracluster scatter) is:

\[ W \triangleq \sum_{j=1}^{M} W_j \]  
(3.9)

The total scatter of the cluster centers (intercluster scatter) is:

\[ B \triangleq \sum_{j=1}^{M} N_j C_j C_j^T \]  
(3.10)

It is necessary to maximize the sum of the eigenvalues of \( W^{-1}B \) in order to get good separation. This is done easily by calculating the trace of the matrix and iterating until this value is at its greatest. In so doing, the ratio of the intercluster scatter to the intracluster scatter is maximized in the direction of the eigenvectors. The primary reason for using this method is that it is invariant to non-singular linear transformations as, for example, a rotation of the data.
3.8 Associative Memory

Neural networks provide an entirely different method of data storage and retrieval than that used in conventional computers. It is commonly known that computers require addresses with which to store and retrieve data. Without the address the data is lost, in fact, all of the data is recovered with an address and all of it is lost without it.

Neural networks behave as content addressable memories. This means that the data is located not by an address, but by a cue at the input. If this cue is "close enough" to a stored pattern then all of the pattern is retrieved, which is significant, because only a partial cue may be necessary for data recovery. These systems have this ability because of their distributed nature. Pattern information is scattered throughout the network providing a strong fault tolerance because, on the average, no individual section is any more important than another. There are many designs pertaining to associative memories [Kohonen 87], but probably the most versatile is the Hopfield network [Hopfield 82]. This configuration (see figure 3.4) is composed of one layer of completely interconnected neurons, but instead of using a specific teacher as in the supervised learning approach these systems converge by means of a Lyapunov energy function. Once this function has been developed and a motion equation derived, the algorithm searches for a location having the lowest energy. These systems can be extremely rapid in their convergence and can be implemented in analog hardware, in which case convergence is only limited by the time constants of the circuitry, as demonstrated by Hopfield and Tank [86].
3.9 Optimization

Many problems exist which require exhaustive searches as, for example, the traveling salesperson problem which is NP-complete. Hopfield nets are frequently used to solve problems of this nature due to their high speed. Energy functions must be derived based on the criteria of the problem, and with this completed, the network searches for a location of minimum energy, indicating at least one solution. These systems can be caught in local minima which has prompted researchers to develop recovery methods to enable the search to continue for the global minimum. Work in this field has been performed by Takefuji and Lee [90] in which they incorporate a hill climbing term which helps kick the system out of local minima.
Chapter 4. Current Neural-Net Approaches

4.1 Target Recognition Problem

The problem being addressed is that of automatic target recognition (ATR). This field encompasses image processing, speech recognition, seismic signal recognition, as well as sonar signal processing. Results from previous methods have yielded some success, as Bhanu's survey [86] indicates, but in practice most fall short of desired goals. This is due in large part to the variability of the target signature and the fact that most previous work has been algorithmic or AI-based. A typical ATR configuration is illustrated in figure 4.1.

In practice, sonar signals from the same class may be very different. The class may be of type A, but actual sources can be dissimilar, furthermore the transfer medium is time-varying and anisotropic, reducing the reliability of a priori information. Using a single hydrophone the process can be stated in the following manner:

\[ Y(t) = F[x,y,z,t,S(t)] \]  \hspace{1cm} (4.1)

where \( S(t) \) is the signal source, \( F(\cdot) \) is the transfer medium in Cartesian coordinates, and \( Y(t) \) is the received signal.

Problems such as these deal with the extraction of uncertain, highly variable data. In this case, more traditional approaches of pattern recognition, e.g., rule-based AI or template matching, are insufficient due to the inherent rigidity of their nature.
Processes dealing with repeatable signatures in stationary backgrounds have historically been identified with the use of, for example, matched filters and correlators, but these are ill-suited for the current task because of the large time-bandwidth products which can occur and the unpredictability of the target signatures [Bouvet, Schwartz 89]. There has been work to enhance the detection of signals in non-Gaussian ocean noise as in [Dwyer 83] where he examines an Arctic data set known as FRAM II. In his paper Dwyer proposes a method of adaptive data partitioning, dependent on the noise distribution prevailing at decision time, which in one example, has permitted a 12 dB reduction in detection threshold for a given false alarm probability. Other work along these lines has been performed by Martin and Schwartz [71], Vastola [84], and many others.

![Figure 4.1 Block diagram of an ATR system.](image)

4.2 Neural Networks and Target Recognition

Neural networks appear to offer a solution to the recognition problem, due primarily to their ability to generalize from examples, remember from experience, and adapt to new situations — three attributes necessary to the success of human interpretation of sonar transients. Studies suggest that neural networks can perform a multitude of classification tasks and that they can behave as good general purpose classifiers [Haley 90].
Today there is much interest in the application of ANNs to the ATR problem. Roth [90] discusses several aspects of the problem and stresses some of the issues and needs. These are consolidated into four major categories:

(1) Good representations of target signatures and backgrounds. Prototypes must be representative of the class structure yet robust to the variations encountered in the environment. A method of reconstruction which has met with moderate success is that of Markov random fields as performed by Geman and Geman [84], but this appears to be most useful with multi-sensor fusion.

(2) Adaptation to target or environmental changes. Many target recognition systems are unable to adapt to environmental changes. If a system is to perform any useful function it must be able to modify its representations while simultaneously retaining acceptable performance at all levels.

(3) Good features for target representation. Good feature extraction is essential to any ATR system. The features must be sufficiently discriminating, limited in number, and usually need to be invariant in some respects [Kohonen 88].

(4) Use of a priori knowledge about target signatures and backgrounds. Specific knowledge of signal shape and the statistical characteristics of the environment are frequently beneficial in automatic target recognition. Neural networks have demonstrated an ability to integrate a priori information automatically through generalization.
4.3 Passive Sonar

This section presents a brief overview of conventional passive sonar systems used to detect and classify underwater acoustic transients. As described, in figure 4.2 passive sonar can be separated into three stages. The first is the transducer which collects the acoustic pressure waves and converts them into electrical energy. More transducers can be configured into an array if needed to provide spatial resolution, but here the emphasis is on single omni-directional hydrophones. The second stage of the sonar system performs the signal processing needed to enhance any signals which may be present. Today this is typically accomplished by spectral analysis and converted to a two-dimensional time-frequency display called a low-frequency analysis record (LOFAR-gram) [Burdic 91]. The final stage is the human listener who determines to which class a received transient belongs.

![Diagram of Passive Sonar Configuration]

Figure 4.2 Basic passive sonar configuration.
Figures 4.3 and 4.4 describe two common signal processing stages, the Blackman-Tukey indirect method and the direct frequency domain method [Oppenheim 78]. The Blackman-Tukey procedure performs an autocorrelation on the data before transforming it to the frequency domain while the latter approach averages an ensemble of spectra. For both systems the final stage is the human listener, which until now, seemed irreplaceable. Current developments in massively parallel systems, i.e., neural-nets are providing the encouragement needed to develop configurations which can, at the least, augment the human listener. As stated earlier this thesis will concentrate on the latter two stages of figure 4.2, particularly stage three (classification) in presenting a neural-net recognition system. As a preamble, some current methods being investigated today are described.

**Figure 4.3** Blackman-Tukey indirect method of spectral analysis. The first step is to autocorrelate the sequence to enhance any signal components which may be present as well as to suppress random factors introduced by noise. Multiplication by a window function is performed next to provide statistical stability, which is followed by a Fourier transform to produce the spectral estimate. As shown here a prewhitener can be included to minimize spectral leakage.
Figure 4.4 Direct method of spectral estimation. The input sequence is partitioned into small, often overlapping, segments which are individually windowed and Fourier transformed. The squared magnitudes of the separate spectra are averaged over each frequency to provide a spectral estimate with lower variance. Averaging across frequency can be performed if additional stability is required.

4.4 An Early System

One of the first papers to seriously discuss the application of neural-nets to the field of sonar was published by Gorman and Sejnowski [88]. This paper addresses a problem concerning classification of sonar returns from two underwater targets. Although concerned with active sonar it is a good starting point for this discussion. Gorman and Sejnowski were concerned with differentiating between a metal cylinder and a cylindrically shaped rock. This was accomplished by impinging a wide band FM chirp upon the targets and collecting the echoes from various aspect angles, and subsequently sending the power spectra of the returns to a neural network for classification. Their best performance, using hundreds of training examples, was over 99% correct classification, independent of aspect
angle. This was certainly a limited application but is included here to illustrate some encouraging results obtained from an early experiment.

The remainder of the work described here is concerned with passive sonar and presents some of the latest methods used to classify underwater acoustic transients. Due to the classified nature of this topic relatively little detailed information is available to the general public, but it is sufficiently abundant to be illustrative.

4.5 Current Systems

Presently, there is much interest in the field of wavelet theory as it applies to signal decomposition. This is, at least in part, due to the spectral variability of most transients as a function of time. The Short-Time Fourier transform is often used in transient analysis to derive local signal characteristics and is used, for example, in the construction of LOFAR-grams. Unfortunately, shorter time windows translate into lower spectral resolution. This limitation can be overcome by using the wavelet decomposition which results in a time-frequency representation of constant relative bandwidth, accomplished by using short time windows at high frequencies and longer windows at lower frequencies. The implementation is illustrated in figure 4.5 as a filter bank tree.
Figure 4.5 A block diagram of the discrete wavelet transform. Input data is highpass and lowpass filtered with filters of bandwidths $\pi/2$. The resulting data is subsampled at half the original rate and the process repeated with filters of bandwidths $\pi/4$, and so on. Subsampling has the effect of increasing the window size used on sequences which have only low frequency components, thus increasing the resolution at lower frequencies, while the filters on these lines eliminate the effects of aliasing. With proper filter design, the high frequency sequences can also be subsampled without loss of information. Details can be found in [Strang 89] and [Rioule, Vetterli 91].

A block diagram of the wavelet transform based system by Desai and Shazeer [91] is shown in figure 4.6. The authors report satisfactory results with this method but point out that they used a subset of the DARPA data from quiet ocean only. Although the results were encouraging this system is computationally intensive. The wavelet decomposition illustrated by these authors converts the single channel input to eight additional channels of constant relative frequency resolution, upon each of which, a Fourier transform is computed (see figure 4.7).
At this point the channels are searched for interesting regions in the spectra from which features are derived. Each feature consists of two elements, the first being the normalized DFT curve length and the second being the relative power within the region. For example, let channels 1 and 5 be chosen to represent a signal, as shown in figure 4.7. The first feature for each channel is computed by dividing the average magnitude variation by the average value of the function within the box (see eqn. 4.2). The second feature is computed by dividing the power within the box by the power within the entire channel (see eqn. 4.3). These features (two for each channel) are input to a neural network for training along with those from several other examples.

\[ \eta_1 = \frac{\| \Delta f(x) \|}{f(x)} \]  

(4.2)

\[ \eta_2 = \frac{f_a(x)}{f_b(x)} \]  

(4.3)

\( f_a \) is the spectrum within the box (figure 4.7) and \( f_b \) is the spectrum within the entire channel. Additional details can be located in Desai and Shazeer [91].

**Figure 4.6** Transient classification system using wavelet preprocessing.
A more traditional approach was attempted by Casselman et al. [91], having a separate configuration for short duration transients and another for tonals (the latter did not employ neural networks). This group reports a 92% probability of a correct classification but the system is extremely complex and does not appear to be very versatile. A section of the block diagram is illustrated in Figure 4.8. This approach sends 64 spectral components to a network which in turn sends its output to a temporal logic unit. It has been my experience that timing units tend to lack adaptability in their interpretations and provide poor generalization. It would seem
that in this case a time-delay neural network would be a preferable choice [Lang et al. 90]. It has also been noted that the large number of input features can produce poorer results due to underdetermination of the system.

![Diagram](image)

**Figure 4.8** Transient classification using windowed FFT.

Further work has been performed by Montana and Theriault [91]. They have concluded that sophisticated detection algorithms should be avoided due to the uncertain nature of the data. Instead, they propose that the majority of efforts be confined to the postprocessing stages which perform high level reasoning. In addition, they concur with the concept of dividing signals into subevents which can be interpreted as disjoint groups. This has similarity to speech processing and has been one of my interests since the project's inception.
Chapter 5. Neural-Net Approach to Transient Classification

5.1 System Description

This chapter presents a new approach to the problem of underwater acoustic transient classification. It is a combination of conventional signal processing and neural network techniques which combine into a robust system capable of classifying a wide variety of signals.

This methodology is based on a comparison between a temporal trajectory of input features and a predetermined trajectory of prototypical features representing a specific signal class. These prototypes are derived from all available class examples by a self-organization algorithm, and provide a method of representing each class by a subset which effectively spans feature space, affording a substantial reduction in data overhead. The trajectory provides a temporal ordering of prototypes, thus furnishing identifiable signatures for class examples, which can be compared with input sequences and used to train neural networks. This concept is displayed graphically in figure 5.1.

Once the prototypes are formed and the class trajectories established it is necessary to assimilate those attributes used for transient classification. In this work the attributes are functions of the distances, indicating similarity (see section 5.6), from the unknown sequences to the prototypical sequences on a point-to-point basis (see figure 5.2). These parameters are then input to the neural networks in order to provide a decision as to whether a class member has been received.
**Figure 5.1** Temporal trajectories through feature space. The x’s symbolize the ordered prototypes created during the self-organization mode, the dots represent a group of ordered unknown patterns from an input sequence and the dotted lines convey the distances, as defined by a suitable metric, from the components of the input sequence to the prototypes at the time of observation. Referring to the dotted lines, sequence #62, #63, #64, and #65 is visibly similar to the prototype sequence #1, #2, #4, and #3 which represents a class example (numerical labels on prototypes are arbitrary). If the input sequence matches the prototypical sequence reasonably well, it may indicate that a class member has been recognized.

**Figure 5.2** A simple function which converts small radial distances to large levels of similarity.
For example, with reference to figures 5.1 - 5.3 let there be 1000 patterns (numbered 1-1000) designating an unknown input sequence containing noise and possibly a signal from a known class. Prior to run time, a known class has been examined and it has been determined that its four major sequential features are closest (as measured by a metric) to the prototypes #1, #2, #4, and #3, in that order. The goal is to determine if four sequential members from the input data are similar to the ordered prototypes. Suppose observations are collected at times $t-3$, $t-2$, $t-1$, and $t$, yielding patterns #61, #62, #63, and #64 respectively. Each of the distances from the aforementioned patterns to prototypes #1, #2, #4, and #3 are computed separately at time $t$, transformed by the similarity function $f(\cdot)$, and sent to the network. It is noted that temporal misalignment will occur because of the uncertainty of a signal's starting point, therefore, at time $(t+1)$ the distances from input patterns #62, #63, #64, and #65 to prototypes #1, #2, #4, and #3 are now computed, transformed, and sent to the network. During each data acquisition one new pattern is added to the input sequence while the oldest is removed and all of the appropriate similarities are again input to the network. This process is repeated with each observation of the input until the sequence is exhausted, and has the effect of sliding the input trajectory across the prototype trajectory until a strong correlation, if any, is observed. A block diagram of the system architecture is detailed in figure 5.4, with further explanations of all system components and the rationales behind them being located in subsequent sections.
Figure 5.3 Neural network designed to recognize a specific trajectory in feature space. This network is configured for a particular sequence of similarities as, for example, those shown in figure 5.1. At time $t$, the top input is "expecting" to see the similarity measure $f(\cdot)$, as a function of distance $d(\cdot)$, between an unknown pattern #61 and prototype #1, $f[d(61,1)]$. The second input receives the similarity measure from pattern #62 and prototype #2, $f[d(62,2)]$, with the remaining inputs receiving $f[d(63,4)]$ and $f[d(64,3)]$. At time $t+1$, the top input is "expecting" to see the similarity measure between pattern #62 and prototype #1, $f[d(62,1)]$, and the remaining inputs expect $f[d(63,2)]$, $f[d(64,4)]$, and $f[d(65,3)]$. If the inputs at time $t$ are simultaneously within the ranges learned during the training session the network will signal the recognition of a class member at that time. This statement also holds true for comparisons at times $t+1$, $t+2$, $t+3$, ..., as well. One network is required for each class, but multiple examples from within the same class can be accommodated by the scheme related in section 5.6.
Figure 5.4 System architecture of neural-net classifier. Early processing is performed on the input data to provide the necessary pattern structure. At this point the effects of background noise are reduced (section 5.5), followed by extraction of the important features. A bank of neural networks compare the resulting feature space trajectories with those created by class examples. The network with the greatest value exceeding its threshold supplies the decision. This can be performed with a maxnet configuration, if convenient.
5.2 Feature Selection

One of the first considerations in designing the present system was how to describe a transient signal, since the quality of any architecture is significantly dependent on those attributes chosen to represent the input, i.e., the feature selection. Original designs considered a multimodal system using multiple signal processing techniques to enhance the data and permit efficacious feature extraction. This included the short-time Fourier transform (STFT), parametric spectral models, and fractal dimension. Merits of each technique are summarized in Table 5.1.

**Table 5.1** Positive and negative attributes of the three major preprocessing techniques considered in this work.

<table>
<thead>
<tr>
<th>Preprocessing</th>
<th>Positive Attributes</th>
<th>Negative Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short-time Fourier transform</td>
<td>Computed by fast efficient algorithm.</td>
<td>Significant variance in estimates.</td>
</tr>
<tr>
<td></td>
<td>No modeling required.</td>
<td>Spectrum defined by many parameters.</td>
</tr>
<tr>
<td></td>
<td>Similar to method used by human sonar operators.</td>
<td></td>
</tr>
<tr>
<td>Parametric methods of spectral</td>
<td>Accurate if proper model is chosen.</td>
<td>Need good model or highly biased estimates will result.</td>
</tr>
<tr>
<td>estimation</td>
<td>Small number of parameters needed to define spectrum.</td>
<td>Numerically intensive algorithms.</td>
</tr>
<tr>
<td>Fractal dimension</td>
<td>Fast algorithm.</td>
<td>Fails to detect transients in noisy ocean.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Not very selective.</td>
</tr>
</tbody>
</table>
After careful consideration it was decided to retain only the STFT for the preprocessing stage, related by eqn. 5.1, where \( x(m) \) represents the data sequence and \( w(n-m) \) is a shifted Hamming window.

\[
X(n, \omega) = \sum_{m=-\infty}^{\infty} x(m)w(n-m)e^{-j\omega n}
\]  

(5.1)

This method was retained for three major reasons. (1) It localizes the spectra, i.e., one Fourier transform over a long sequence buries any evidence of short term spectral variation, while the STFT consisting of a series of Fourier transforms, can reveal spectra within narrow temporal regions. (2) It is easy to compute. One of the requirements of this system is that it function in real-time, therefore it is advantageous to choose the STFT since it can be calculated with the Fast Fourier transform algorithm. (3) It is similar to the method used to produce LOFAR-grams which are interpreted by human sonar operators. An example of a STFT is illustrated in figure 5.5.

5.3 Preprocessing

The signal conditioning stage performs the STFT as described in the paragraphs above by utilizing 256 length data sequences (records), corresponding to 10ms windows. These sequences are converted to 7 overlapping segments of 64 length (50% overlap), these shorter segments Fourier transformed, and their squared magnitudes averaged to provide a frequency resolution of about 400 Hz and a variance of approximately 1/7 of that produced by a single transformation. A
Hamming window was used for this process, sometimes referred to as the Welch periodogram method [Marple 87].

Further shortening of the window length will continue to reduce the variance, but of course this will also tend to increase the bias, thus smearing the spectrum to a greater extent. Frequency resolution can be sacrificed for a lower variance estimate because the spectral information is augmented by temporal characteristics, as will become apparent in the following pages.

Figure 5.5 This is a plot of the Short-time Fourier transform of a signal, depicting an increase in frequency with each observation window. This is commonly called a spectrogram.
Regardless of the processing method, only those spectral components up to about 6 kHz are retained as the important features, since they contain the frequency components of interest in this application.

Once a spectral estimate has been completed, the system waits 2.5 ms before acquiring an additional 256 point data record, providing a 75% temporal overlap thus ensuring that rapid spectral fluctuations between samples are not missed. This method has proved useful in earlier designs [Pao, Hemminger 91].

5.4 Trajectories

Historically, most signal receiving systems have analyzed target candidates as independent events, as for example, systems based on correlation or matched filters. For this reason a single detection threshold is commonly set to preserve a constant false alarm rate (FAR) and constant probability of detection (PD). To keep the FAR at a low value it is necessary to use a relatively high threshold, but this also has the effect of lowering the PD. With the development of neural networks it should be possible to retain several smaller signal components or "events" and determine whether they can be combined to constitute the detection of a target (section 5.7). For this reason the STFT appears to be an appropriate processing choice because it enables a signal to be parsed into a set of spectral "slices" separated temporally, which in turn can permit the employment of multiple detection thresholds conceivably set from experience. In this way a neural network could observe features from detected events, interpret their relationships, and provide a decision based on the collective information. This concept is not entirely
new as related by Roth [89], but it has not been computationally effective until recently.

5.5 Clustering and Prototypes

With the previous discussions in mind, it is necessary to relate how clusters and their representative prototypes are formed. A simple iterative metric clustering algorithm was implemented [Pao 89a] which determines which cluster center is closest to a new data entry, in this case a power spectrum. A general description of the algorithm follows:

1. Read the first pattern $x_1$ and designate it as cluster center (prototype) $c_1$.
2. Read the next pattern and determine whether it is close to $c_1$ based on some threshold. Average them if the distance is less than the threshold, otherwise create a new cluster center $c_2$.
3. Read the $n^{th}$ pattern $x_n$ at time $t$ and determine whether it is close, as determined by a suitable metric, to any of the $K$ cluster centers created thus far. If it is not, create a new cluster center, but if it is close to $c_j$ then update each element $i$ of $c_j$ by the following formula:

\[
c_j(t + 1) = c_j(t) \frac{m}{(m + 1)} + x_i \frac{1}{(m + 1)}
\]  

(5.2)

where $m$ is the current number of members in cluster $c_j$.

4. Go to step 3 until clustering has stabilized.
It is tempting to consider the power spectra as a vector, and by doing so, employ a metric such as a 2-norm in the algorithm above. This is not a good choice for this type of data since vector norms do not consider adjacency of the dimensions, i.e., no dimension is any closer to any other than it is to all the rest. What this implies is that the 16-dimensional vectors in figure 5.6 are equidistant.

\[
\begin{align*}
A &= [0,Y,0,0,0,0,0,0,0,0,0,0,0,0,0,0] \\
B &= [0,0,0,Y,0,0,0,0,0,0,0,0,0,0,0,0] \\
C &= [0,0,0,0,0,0,0,0,0,0,0,0,0,0,Y,0]
\end{align*}
\]

**Figure 5.6** The three noise-free vectors at the top are shown at the bottom as superimposed narrowband spectra. Using a 2-norm, the distance from A to B, \(d(A,B) = Y\sqrt{2}\), as is \(d(A,C)\) and \(d(B,C)\). This indicates that if spectrum B is related to A, as in a Doppler shift for example, a Euclidian norm cannot imply a relationship unless substantial overlap occurs.
Unlike figure 5.6, real spectra will overlap to some degree, which can be attributed to correlation, noise, Doppler shifts, dispersion, etc. and it is useful to retain some of the information contained therein. It has been documented [Kinney, et al 80] that sonar operators, and people in general, tend to have an easier time noticing sharp jumps in data trends, referred to as singularities, rather than subtle or smooth changes. With this as motivation, and with concern regarding the previous discussion, a decision was made to consider the spectra as a 2-dimensional plot rather than as a multi-dimensional vector, and a metric sought that would be more responsive to adjacent components. A measurement method used in fractal imaging has proven suitable for this task. It is the Hausdorff metric, defined by the following equations operating on the sets A and B, [Barnsley 88], [Essex, Nerenberg 90], [Grassberger 85].

\[ h(A, B) = \max \{d^*(A, B), d^*(B, A)\} \]  \hspace{1cm} (5.3)

where,

\[ d^*(A, B) = \max \{d^*(x, B) : x \in A\} \]  \hspace{1cm} (5.4)

and

\[ d^*(x, B) = \min \{d(x, y) : y \in B\} \]  \hspace{1cm} (5.5)

In words, eqn. 5.5 instructs one to find the distance from a point \( x \) in the set \( A \) to all points \( y \) in set \( B \), retaining the smallest. Eqn. 5.4 states that this same operation is to be performed for all of the points \( x \) in \( A \) and that the largest of these minima
be retained. At this point the same operations are to be performed from the opposite set and the largest value also reserved. Finally, eqn. 5.3 says to keep the largest of the two remaining values. This is the Hausdorff distance between the sets, characterized as a method that looks for the greatest dissimilarity between two sets. The measurement $d$ in the equations above represents some suitable metric. Since straight-line distances on a 2-dimensional plot are being used in this work, a 2-norm is appropriate for the point-to-point measurements. Figure 5.7 illustrates the behavior of the Hausdorff metric as related to this theme.

**Figure 5.7** Spectral distances as interpreted by the Hausdorff metric (three superimposed spectra). If measured by a 2-norm the distance $d(A,B) = d(A,C)$ with no clear indication of which distance is less, but with the Hausdorff metric $d(A,B) < d(A,C)$ providing a clear indication of which spectrum is closer to $A$. Note, spectral line 14 is not the closest component of spectrum $C$ to spectrum $A$ in the Hausdorff sense. The closest component of $C$ to $A$ is in frequency bin #2 as shown above (see preceding paragraph).
In utilizing this approach the spectra are considered as patterns, and as a consequence of the noise susceptibility of the Hausdorff metric, as pointed out by Giardina and Dougherty [88], a great deal of consistency in the data is needed. In addition, the signals can become severely distorted from trends in the noise, therefore, a simple scheme is incorporated which attempts to eliminate characteristics due to the noise while retaining most of the signal information (see figure 5.8). Without this procedure, patterns derived from similar signal components, taken from dissimilar backgrounds, can look quite different. The noise cancellation is accomplished by the configuration illustrated in figure 5.9. Its function is in effect, a prewhitening filter which removes most noise trends, thus reducing their influences on the spectral characteristics of the target signal. Subsequently, the spectra are normalized by their 2-norms to force all patterns to the same scale.

Figure 5.8 Upper spectrum is contaminated with background noise, making it difficult to recognize the embedded signal pattern. Lower spectrum has been processed by the configuration in figure 5.9 to reduce the effects of noise.
Figure 5.9 Data prewhitener. This system subtracts the background from the current spectrum while setting negative components to zero. The background estimate is produced by sending each spectral component through a single pole lowpass filter, making the assumption that the noise is at least quasi-stationary over limited intervals. The background is not updated until the current spectrum has been processed and its power computed. Division can be used, but the configuration shown here realizes about 5% lower variance with subtraction.

5.6 Trajectories and the Functional-Link Net

As noted earlier it is necessary for the system to effectively look for temporal tracks or trajectories and determine whether they constitute a signal classification. The tool used in the current pattern recognition approach is the Functional-Link Net developed by Pao [89a]. This network was chosen due to its rapid training time, simplicity, and ease of analysis. An examination of this classifier will follow after a brief description of the configuration. Figure 5.10
illustrates the formulation of cluster trajectories, which are computed for the target signals as well as for a large number of noise files (see also figure 5.1). The result is a list of closest clusters and associated Hausdorff distances corresponding to the temporal sequence of PSDs. For the moment it is assumed that all trajectories from a specific class are similar in order but not distance, allowing for a network to be designated for each class. At this stage the FLN is trained to output a 1 if a signal is present, and a 0 if the input is noise.

**Figure 5.10** Cluster trajectory of signal prototype. The x’s symbolize the prototypes created during the self-organization mode, the dots represent a group of ordered unknown patterns from an input sequence and the dotted lines convey the distances from the components of the input sequence to the prototypes at the time of observation. In this example the input sequence (#1, #2, #3, #4, #5) has traversed pattern space coming nearest to the prototypes #3, #5, #2, #4, #1, respectively, during one of the observations. The similarity measures (here shown as cylindrical functions) with regard to these prototypes are input to a network to determine whether a member from a particular class has been encountered.
As explained earlier, the actual distances $d_i$ are not input to the network, but instead a thresholding function is incorporated $f(d_i, \sigma_i)$ for which smaller distances translate into greater magnitudes indicating similarity, e.g. Gaussian exponential, lowpass filter equation, etc. This provides a smooth transition for cluster membership and allows for the inclusion of multiple trajectories in the same network. For example, if the first cluster center corresponding to class A is #3, 40% of the time, and cluster #7 for the remaining 60%, then the following expression can be computed and sent to network input number one:

$$z_i = 0.4 f(d_i, \sigma_3) + 0.6 f(d_i, \sigma_7)$$  \hspace{1cm} (5.6)$$

This simple expression has the effect of creating a weighted combination of clusters based on relative frequencies in the training set. Others, [Duda, Hart 73] have suggested using networks having multiple outputs when a signal is defined by more than one trajectory, but this requires changing the configuration as new signatures become available. The procedure presented here has several advantages: (1) The network design remains fixed. (2) Weighting factors can be adjusted manually if desired. (3) The size of the neighborhood $\sigma_i$ can be modified at will for greater or lesser selectivity at individual cluster locations. (4) Clusters are not actually combined. This is important for a real-time system in that it holds constant the number of distances which need to be calculated. In the interest of clarity, multiple signatures will not be considered for the remainder of this discussion.
Figure 5.11 illustrates the network during the run-time mode, as shown in the figure the first input to the network is associated with cluster 3, therefore, the similarity measure is input to the net at that location, with the other inputs being derived in a similar manner as time progresses. During each system clock cycle the network output is computed and a binary decision recorded. If the inputs are such that the network produces a low output value it is an indication that no target is present. Conversely a high output signifies a true classification (see also figure 5.3).

**Figure 5.11** Adjusted Hausdorff distances input to FLN. There is a separate network for each signal class. During a system clock cycle $t_i$ the network output is computed and a binary decision recorded. The function $f(\cdot)$ is used as a method to indicate membership or similarity, being a Gaussian or any function similar in shape to a lowpass filter. The parameters $d_i$ and $\sigma_i$ denote the Hausdorff distance from the input pattern to the prototype and the width of the function, respectively. At the next clock cycle one new pattern is added to the sequence while the oldest is removed and all of the appropriate distances are again input to the network.
5.7 Theoretical Analysis

It is now related why neural networks have been an efficacious design choice and why the concept of distributing the signal information into clusters and interpreting trajectories is advantageous. The following investigation could be performed with any neural network, but the nature of the FLN permits a relatively simple analysis. The results are based on maximum likelihood and will be compared to those from earlier work by Roth [89].

Consider two types of events, noise \( n \) and target \( t \), which can be assembled into trajectories in feature space. Also let \( x_i \) denote whether an event has become a member of a particular cluster \( (x_i = 1) \) or not \( (x_i = 0) \). Assume that these data points can be members of each of a total of \( N \) clusters and that they are independent. In addition, let \( M \) of these clusters be associated with a signal that has generated a distinctive sequence and that all signals \( S \) have equal probability, which is the usual assumption for maximum likelihood analysis. The probability of noise becoming a member of any one cluster \( i \) is \( \alpha_i \) while the probability of a target component becoming a member is \( \beta_i \).

To begin the analysis start with:

\[
P_i(x_i) = \begin{cases} 
\alpha_i & x_i = 1 \\
1-\alpha_i & x_i = 0 
\end{cases} \tag{5.7}
\]

which is the probability that a noise component is associated with a particular cluster and:

\[
P_i(\alpha_i) = \begin{cases} 
\beta_i & \alpha_i = 1 \\
1-\beta_i & \alpha_i = 0 
\end{cases} \tag{5.8}
\]
which is probability of a target component \( t \) being associated a cluster. Noise can produce random trajectories, where the probability of any one of these is:

\[
P_n (\mathbf{x}_0) = \prod_{i=1}^{N} \alpha_i^{x_i} (1 - \alpha_i)^{1-x_i}
\]

(5.9)

The probability of target segments producing a particular trajectory, i.e. a signal, when noise is present is accordingly:

\[
P_i (\mathbf{x} | S) = \prod_{i=1}^{M} \beta_i^{x_i} (1 - \beta_i)^{1-x_i}
\]

(5.10)

In reality, noise will still be present for the duration of the signal and beyond, therefore, the total probability of a particular pattern being present is:

\[
P_i (\mathbf{x} | S) = \prod_{i=1}^{N} \beta_i^{x_i} (1 - \beta_i)^{1-x_i} \prod_{j=1}^{N} \alpha_j^{x_j} (1 - \alpha_j)^{1-x_j}
\]

(5.11)

A receiver based on the likelihood ratio yields a minimum probability of error when the transmitter input events \( S \) are equally likely [Wozencraft, Jacobs 65]. The likelihood ratio can be stated as the probability of a sequence in the presence of signal and noise (eqn. 5.11) divided by the probability of noise alone (eqn. 5.9). A receiver which maximizes this ratio is said to be optimal in a maximum likelihood sense. This reasoning is illustrated below in eqn. 5.12, and it will be shown later how the FLN attempts to maximize this expression.
\[ L(x|S) = \frac{ \prod_{i=1}^{N} \beta_{x_i} (1-\beta_i)^{1-x_i} \prod_{j=1}^{M} \alpha_{z_j} (1-\alpha_j)^{1-x_j} }{ \prod_{j=1}^{M} \alpha_{z_j} (1-\alpha_j)^{1-x_j} } \]  

Maximization of expression 5.12, can be accomplished by maximizing its logarithm shown here.

\[
\ln[L(x|S)] = \sum_{i=1}^{N} \left[ x_i \ln(\beta_i) + (1-x_i) \ln(1-\beta_i) \right] \\
+ \sum_{j=1}^{M} \left[ x_j \ln(\alpha_j) + (1-x_j) \ln(1-\alpha_j) \right] \\
- \sum_{i=1}^{N} \left[ x_i \ln(\alpha_i) + (1-x_i) \ln(1-\alpha_i) \right] 
\]  

Eqn. 5.13 can be reduced to the following form:

\[
\ln[L(x|S)] = \sum_{j=1}^{M} x_j \ln \left[ \frac{\beta_j (1-\alpha_j)}{\alpha_j (1-\beta_j)} \right] + \sum_{i=1}^{M} \ln \left[ \frac{1-\beta_i}{1-\alpha_i} \right] 
\]  

This is the expression which should be maximized. All signal sequences are equally probable and cluster numbers are assigned arbitrarily, therefore, with no loss of generality the discussion is continued with a sequential set of cluster numbers. This simplifies the examination and does not alter the final result. Note the similarity of equations 5.14 and the general form of the FLN shown in eqn 5.15:

\[
Y = F \left\{ \sum_{i=1}^{K} \lambda_i X_i + \theta \right\} 
\]
Looking at equation 5.14 a strong relationship is seen between the weights learned by the network and the unknown a priori probabilities of the individual clusters. As the first term of eqn. 5.14 demonstrates, the ratio of the component probabilities has a direct relationship with the magnitudes of the corresponding weights. If $\beta_i/\alpha_i$ increases, meaning greater certainty that a target component is present, the weights will increase in magnitude. Conversely, a lower ratio means that the cluster population is unreliable and should have a negative influence on the outcome. Equal probabilities render weights having magnitudes near zero. In practice, the inputs to the net are enhanced to some extent by the methods of functional enhancement or joint activation, and this shown in eqn. 5.16 where one level of joint activation is included (adjacent terms only), but this could be generalized to higher orders:

$$Y = F \left\{ \sum_{i=1}^{M} w_i x_i + \sum_{j=1}^{K} e_j x_j x_{j+1} + \theta \right\} \quad (5.16)$$

The first summation represents the $M$ original pattern elements while the second represents the $K$ joint activations. The objective is to maximize the likelihood ratio by maximizing the value of the summations during the training process, thus generating the decision boundaries. At this point the performance of this system is demonstrated by determining the signal-to-noise ratio (SNR) for the receiver. $F(\cdot)$ is a threshold logic function, or low temperature sigmoid, that is used for training and can be removed for this investigation due to its being monotonic. In the discussion that follows $w_i$ are the weights on the original $M$ pattern attributes and $e_j$ are those on the $K$ enhanced dependent inputs. The constant
\( \theta \) is a bias that drops out of the analysis. This results in the modified FLN equation below.

\[
\Psi = \sum_{i=1}^{M} w_i x_i + \sum_{j=1}^{K} e_j x_j x_{j+1} \tag{5.17}
\]

The SNR can be calculated by:

\[
SNR = \frac{E\{\Psi_{s+n}\} - E\{\Psi_n\}}{\sigma_n} \tag{5.18}
\]

where \( E\{\Psi_{s+n}\} \) is the expected value of the function when signal and noise components are present and \( E\{\Psi_n\} \) is the expected value with noise alone. If the presence of a pattern is considered, the expected value of the summation with signal and noise is:

\[
E\{\Psi_{s+n}\} = \sum_{i}^{M} \beta_i w_i + \sum_{j}^{K} \beta_j \beta_{j+1} e_j \tag{5.19}
\]

while the expected value without the signal is:

\[
E\{\Psi_n\} = \sum_{i}^{M} \alpha_i w_i + \sum_{j}^{K} \alpha_j \alpha_{j+1} e_j \tag{5.20}
\]

At this point the variance can be calculated as:

\[
\sigma^2 = E\{\Psi_n^2\} - E\{\Psi_n\}^2 \tag{5.21}
\]
where

\[ E \{ \Psi_i^2 \} = \sum_{i}^{M} \alpha_i w_i^2 + \sum_{i}^{M} \sum_{j}^{M} \alpha_i \alpha_j w_i w_j (1 - \delta_{ij}) \]

\[ + 2 \sum_{i}^{M} \sum_{j}^{K} w_i \alpha_j \alpha_{j+1} e_j \]

\[ + \sum_{i}^{K} \alpha_i \alpha_{i+1} e_i^2 + \sum_{i}^{K} \sum_{j}^{K} \alpha_i \alpha_{i+1} e_i \alpha_j \alpha_{j+1} e_j (1 - \delta_{ij}) \]  \hspace{1cm} (5.22)

and

\[ E \{ \Psi_j^2 \} = \sum_{i}^{M} \alpha_j w_i^2 + \sum_{i}^{M} \sum_{j}^{K} \alpha_i \alpha_j w_i w_j (1 - \delta_{ij}) \]

\[ + 2 \sum_{i}^{M} \sum_{j}^{K} w_i \alpha_j^2 \alpha_{j+1} e_j \]

\[ + \sum_{i}^{K} \alpha_i^2 \alpha_{i+1}^2 e_i^2 + \sum_{i}^{K} \sum_{j}^{K} \alpha_i \alpha_{i+1} e_i \alpha_j \alpha_{j+1} e_j (1 - \delta_{ij}) \]  \hspace{1cm} (5.23)

These are combined into eqn. 5.21 and simplify to yield:

\[ \sigma^2 = \sum_{i}^{M} (\alpha_i - \alpha_i^2) w_i^2 + 2 \sum_{i}^{K} (\alpha_i \alpha_{i+1} - \alpha_i^2 \alpha_{i+1}) w_i e_i \]

\[ + \sum_{i}^{K} (\alpha_i \alpha_{i+1} - \alpha_i^2 \alpha_{i+1}^2) e_i^2 \]  \hspace{1cm} (5.24)
Eqns. 5.19, 5.20, and 5.24 can be placed into eqn. 5.18 yielding the final result.

\[
SNR = \frac{\sum_{i=1}^{M} (\beta_i - \alpha_i)w_i + \sum_{i=1}^{K} (\beta_i \beta_{i+1} - \alpha_i \alpha_{i+1})e_i}{\sqrt{A + B + C}}
\]

\[
A = \sum_{i=1}^{M} (\alpha_i - \alpha_i^2)w_i^2
\]

\[
B = 2\sum_{i=1}^{K} (\alpha_i \alpha_{i+1} - \alpha_i^2 \alpha_{i+1})w_i e_i
\]

\[
C = \sum_{i=1}^{K} (\alpha_i \alpha_{i+1} - \alpha_i^2 \alpha_{i+1}^2)e_i^2
\]  

(5.25)

This is a formidable expression which is explained by the fact that the enhancements are dependent on the linear terms, but it can be useful if some simplifying assumptions are made. This analysis has assumed that all \textit{a priori} cluster probabilities are different, but if \(\beta_i \to \beta\) and \(\alpha_i \to \alpha\), as in many analyses, the expression is much more compliant. In this case the SNR can be written as:

\[
SNR = \frac{(\beta - \alpha)\sum_{i=1}^{M} w_i + (\beta^2 - \alpha^2)\sum_{i=1}^{K} e_i}{\sqrt{\alpha(1-\alpha)\sum_{i=1}^{M} w_i^2 + 2\alpha^2(1-\alpha)\sum_{i=1}^{K} w_i e_i + \alpha^2(1-\alpha^2)\sum_{i=1}^{K} e_i^2}}
\]  

(5.26)

The probability of noise being associated with a cluster is typically much less than that of a signal component since the clusters were formed from examples of the signal. As a consequence some approximations can be performed which will enable a more useful interpretation. It is noted that one term in the numerator has the very small coefficient of \(\alpha^2\). In addition, all of the terms in the denominator have coefficients of \((1-\alpha)\) and \((1-\alpha^2) = 1\) but two of these also have coefficients of
$\alpha^2$, therefore, as an approximation the right two terms of the denominator and the right term in the numerator will be neglected with eqn 5.27 as the result.

$$SNR = \frac{(\beta - \alpha) \sum_{i}^{M} w_i + \beta^2 \sum_{i}^{K} e_i}{\sqrt{\alpha(1 - \alpha) \sum_{i}^{M} w_i^2}} \quad (5.27)$$

If the enhancements are removed for a moment this expression can be written as:

$$SNR = \frac{(\beta - \alpha)}{\sqrt{\alpha(1 - \alpha)}} \frac{\sum_{i}^{M} w_i}{\sqrt{\sum_{i}^{M} w_i^2}} \quad (5.28)$$

The left factor is dependent only on the *a priori* probabilities, while the right factor is the ratio of the 1-norm of the numerator (noting that by experience the weights are usually positive) to the 2-norm of the denominator. By the triangle inequality the result must be greater than or equal to one, meaning that the right factor has an enhancing effect. If this were not the case it would infer that more observations could produce inferior results to that of a single observation — an unacceptable condition. If the weights and input values $x_i$ are considered strictly as binary, eqn. 5.28 will reduce to:

$$SNR = \frac{(\beta - \alpha)}{\sqrt{\alpha(1 - \alpha)}} \sqrt{M} \quad (5.29)$$

which is in complete agreement with Roth’s [89] analysis for the Hopfield net. A geometric analogy can be drawn from observing eqns. 5.27 and 5.29. From eqn.
5.29 It is seen that a fixed threshold exists for a specific vector dimension $M$. This can be visualized as a spherical threshold which the vector must exceed in magnitude before reporting a target detection, as illustrated in figure 5.12. A slightly different interpretation can be derived from eqn. 5.27. In this case the threshold is not only dependent on the vector dimension, but also on the values of the weights. This produces an asymmetrical hypersurface synthesized by the network during training, as illustrated in figure 5.13. In this case the network is not only able to weigh the different inputs accordingly, but also permitted to judge if any correlations between the inputs are significant. These correlations can be important because they indicate whether more than one input is always a member of a cluster or clusters. If this is true then the network is able to place more emphasis on those enhancements. In practice these systems are trained with several examples of the prototype as well as a large number of noise files, thus enabling the FLN to determine where the dichotomies exist and to make changes to the weights accordingly. If the cluster distance at a particular input varies little while training within the groups (intracluster) of signal and noise, but simultaneously varies substantially between the groups (intercluster) of signal and noise, that weight will tend to have a relatively large value, indicating a greater information content. Conversely, high intracluster variance and low intercluster variance induces smaller weights by comparison meaning little information is gained at those inputs. Due to the directional nature of this system, the right side factor of eqn. 5.28 (or 5.27) can change in value from slightly above $\sqrt{M}$ to below it depending on the training set.
Figure 5.12 Simplified geometrical interpretation of signal detector described by equation 5.29. If the number of matches exceeds a given threshold the detector signals that a target is located.

Figure 5.13 Simplified geometrical interpretation of neural network signal detector as described by equations 5.27 and 5.28. The threshold is directional, having been learned during the training process.
Chapter 6. Experimental Results

This chapter presents results obtained from running the neural-net classification system on several transients, all of which are described in the following sections. Most of these signals are from the DARPA standard transient data set introduced in chapter one, but some additional examples were computer generated, while still others were derived from acoustic experiments in the laboratory. All of the results will be compared to those obtained by correlators, where applicable, for comparison.

6.1 Prototypes

The first step in implementing this system is to choose an appropriate set of class examples and form a prototypical set of cluster centers. As stated earlier, there is usually a very poor set of specimens within a real data set, which can lead to the use of every target signature regardless of quality, resulting in performance degradation. In this work less than ten examples from each class existed with which to train the system. This, no doubt, reduced performance to some extent, but overall, the results were very encouraging.

In order to build a cluster set several signatures are accumulated from each class. In addition, a large collection of noise files is amased to provide the system with a background group to use later for counter training (training for non-target outputs). All examples are then processed with the clustering algorithm described in chapter 5, until the cluster centers stabilize. There are three questions in regard
to this process which should be addressed. (1) The spectra are normalized with respect to their 2-norms, meaning that the magnitude of the largest component is less than one, yet the adjacent frequency bins are separated by integers, so how is the discrepancy in horizontal and vertical scale handled? (2) What is the cluster radius and how is it determined? (3) What criterion was used to halt the cluster routine? In answering the first two questions the fluctuation of each signal was observed in both types of noise revealing that strong spectral peaks varied by about 15% after normalization, and that they only shifted by one spectral bin (+/- 400 Hz) at the most. With this as a guide a vertical multiplier of 8.0 and a radius of 1.2 were chosen, i.e., 15% (see figure 6.1), although neither parameter is overly sensitive.

![Graph showing magnitude vs. frequency](image)

**Figure 6.1** Vertical axis is multiplied by 8.0 with radius 1.2. These parameters allow for a 15% magnitude deviation and a 400 Hz frequency deviation while still being considered in the same class.
The last concern is the criterion used to halt the cluster algorithm. Chapter three presented a technique studied by Fukunaga and Koontz [70] which uses scatter matrices to determine when a cluster algorithm has converged to a stable solution, but this method is a variation of a 2-norm, making it inappropriate when using Hausdorff distances. Therefore, in this work the cluster algorithm is iterated while trying to maximize the following ratio:

\[
\Omega = \frac{1}{M} \sum_{k=1}^{M} d(C_k, \mathcal{C})
\]

\[
\frac{1}{M} \sum_{j=1}^{M} \left[ \frac{1}{N_j} \sum_{i=1}^{N_j} d(X_i, \mathcal{C}_j) \right]
\]

(6.1)

This is the ratio of the intercluster dispersion to the intracluster dispersion in terms of Hausdorff distances, \(d(A, B)\). The \(C_k\) are the cluster centers, \(\mathcal{C}\) is their centroid, \(M\) is the number of clusters, \(X_i\) is an individual cluster element, and \(N_j\) are the number of elements in the \(j^{th}\) cluster. In maximizing this ratio the greatest separation of the prototypes is achieved while minimizing the dispersion of individual cluster members within groups. As cluster iteration continues, the outliers are removed from their current groups and permitted to associate themselves with new groups. Suitable convergence is usually realized after 5 or 6 iterations when \(\Omega\) is near one of its maximum values, producing between 10 and 20 cluster centers.
6.2 Network Training

The next procedure requires training on class representitives. At this point all suitable examples of a signal class, as determined by DARPA, are sent to the cluster algorithm while in the consult mode. The output is a list of nearest prototype segments (power spectra) and their corresponding distances. The cluster numbers provide an ordering for the network while functions of the distances are the actual inputs (section 5.6). This configuration utilizes as many network inputs as the class length necessitates. For example, 2 records (256 samples / record) may be the average sequence length of a signal class. Using 75% overlap (to provide sufficient temporal invariance) this produces five power spectra, hence five nearest cluster centers, indicating five network inputs, as shown in figure 6.2. When using full joint activation the number of enhanced inputs is \( n(n - 1)/2 \), suggesting that the network designed for this class should have 15 weights.

The procedure above is also performed on a large set of noise files (30 to 40) from quiet and noisy ocean, having the same temporal length. This second group is necessary to permit counter training so that the network can not only determine if a signal is present, but when it is not. Without the noise, the network would train to zero error very rapidly, but have no noise immunity during the consulting phase.

While training the network, signals and noise are associated with targets of 1.0 and 0.0 respectively. Typically, training requires approximately 5000 iterations to converge to a mean squared error of \( 10^{-5} \).
Figure 6.2 Input records are processed with 75% overlap to produce a smooth spectral flow and to eliminate the possibility of missing any significant features. Two 256 point records produce 5 power spectra. In general, $N$ records produce $4N-3$ spectra.

6.3 Run-time Evaluation

With training completed, the classifier is ready for consultation. The objective is to detect and correctly classify members from each class in real time, independent of signal strength and ocean noise level, with low sensitivity to disturbances encountered in the medium. The first set of test transients was supplied by DARPA and consists of six classes sampled at a rate of 25 kHz, and referred to as A - F. Classes A and B have complex signatures while the remaining four are tonals. All of the signals were supplied in quiet and noisy ocean and at various signal-to-noise ratios (SNRs), in addition to those used for training, there is a similar set for testing purposes. Results from signal classes A, B, and C are displayed on the following pages, while those from D, E, and F are included at the end of this section in the form of a table.
Figure 6.3 illustrates two examples of transient A in quiet ocean and noisy ocean. Class A signals have a length of approximately 400 samples corresponding to 16 ms duration and can have SNRs from about 1.5 dB to 7 dB. For this reason samples from class A are considered as being two records in length. Referring to

![Signal Duration](image)

**Figure 6.3a** Transient from class A at highest SNR in quiet ocean (7.0 dB).

![Signal Duration](image)

**Figure 6.3b** Transient from class A at lowest SNR in noisy ocean (1.6 dB).

Both figures are at the same scale.
the discussion above this produces a 15 weight network with which to identify the trajectories. The results of this work on class A signals are illustrated in figure 6.4 with an ROC (receiver operating characteristics) curve. These curves plot the probability of signal detection against probability of false alarm while varying the detection threshold. Each curve represents a different SNR.

![ROC curve](chart.png)

**Figure 6.4** Receiver operating characteristics for neural-net classifier trained on seven examples from class A, and a correlator using the cleanest signature from class A. The SNR is 2.0 dB. The poorer results for the correlator are due to the noisy signature that was used in the correlation operation.
The results above reveal that the neural network system provides an improvement of about 0.5 dB over the correlator. This is explained by the fact that the correlator relies on a noise free example of the signal, which in the case of underwater acoustic transients is not realistic. On the contrary, the neural network system only requires several examples from which to synthesize an appropriate discriminant function.

![Probability of Detection vs Probability of False Alarm](image)

**Figure 6.5** Receiver operating characteristics for neural-net classifier trained on seven examples from class A, and a correlator using the cleanest signature from class A. The SNR is 1.0 dB. The poorer results for the correlator are due to the noisy signature that was used in the correlation operation.
Signals from class B are composed of two similar components separated by about 30 ms of background noise, possibly indicating the reception of a multipath signal (see figure 6.6). This type of signal has proven to be nearly intractable due to its dual nature and low SNR, and the possibility of temporal variance between the two components. In this work signal B was considered as one entity consisting of 5 records, thus providing 17 network inputs and originally 153 weights. The system was pruned before training to remove many of those inputs in the noise region. Further pruning was performed during and after training when weights exhibited very small values, indicating little or no contribution to the result. The ROC curve for this class is displayed in figure 6.7. The nature of this signal makes it inappropriate for direct correlation due to the large noise region, therefore, there is no ROC comparison for class B.
Figure 6.6a Transient from class B at highest SNR in quiet ocean (4.3 dB).

Figure 6.6b Transient from class B at lowest SNR in noisy ocean (1.9 dB for non-noise segments). This figure has at the same scale and positioning as figure 6.6a.
As figure 6.7 illustrates, the results for class B are not as favorable as those from class A, but this is not really surprising when one considers the nature of the signal. Certainly better performance would have prevailed had there been a larger data set since, in this case, only four examples were considered useful for training, but nevertheless the results were in accord with those attained by other researchers working on the same data set [Proc. IEEE Conf. N-N for Ocean Eng. 91].

![Graph showing ROC curve with SNR = 1.0](image)

**Figure 6.7** Receiver operating characteristics for neural-net classifier trained on signals from class B. Four class examples were used in training.
Class C signals are tonals. As figure 6.8 demonstrates, these are not realistic underwater acoustic transients, but their inclusion helps complete this performance evaluation.

Figure 6.8a Transient from class C at highest SNR in quiet ocean (14.3 dB).

Figure 6.8b Transient from class C at lowest SNR in noisy ocean (3.5 dB).
The classification system was trained on 8 examples from class C, including that shown in figure 6.8b. These signals are considered as being two records in length, thus providing 5 network inputs and 15 weights, as in the case of class A. The ROC curves are shown below in figures 6.9 and 6.10. As expected, the curve from the correlator illustrates slightly better performance, but this is not entirely unexpected since filters of this nature maximize the signal-to-noise ratio and class C contains many relatively noise free examples. It should also be remembered that class C transients would not be typically encountered in underwater acoustics.

![ROC Curve](image)

**Figure 6.9** Receiver operating characteristics for neural-net classifier trained on eight examples from class C, and a correlator using the cleanest signature from class C. The SNR is 2.0 dB. The superior results for the correlator are due to the use of a nearly **noise free** signature in the correlation operation.
Figure 6.10 Receiver operating characteristics for neural-net classifier trained on eight examples from class C, and a correlator using the cleanest signature from class C. The SNR is 1.0 dB. The superior results for the correlator are due to the use of a nearly noise free signature in the correlation operation.

The remaining transients in the DARPA set are protracted sinusoids. The duration of class D is 100ms at 3 kHz, that of class E is 1.0 second at 150 Hz, and of class F is 8.0 seconds at 250 Hz. A separate network was trained on shortened examples from these classes as well as those discussed previously, producing six classification networks. Although the problem of cross classification is a reality, no
difficulties with false alarms were experienced on non-target signals, i.e. networks trained on one signal typically produced little or no output when subjected to alternate transients. To further reduce any possibility of cross classification the networks were run competitively, meaning that if more than one classifier produced a high output the net exceeding its threshold to the greatest extent was the winner. Table 6.1 summarizes the results from the DARPA test set.

**Table 6.1** Summary for DARPA test set. This configuration does not produce false alarms due to the presence of a competitive algorithm.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>Missed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>71/72 98%</td>
<td>51/58 88%</td>
<td>40/40 100%</td>
<td>12/12 100%</td>
<td>20/21 95%</td>
<td>15/15 100%</td>
<td>1/72 2%</td>
</tr>
<tr>
<td>Truth</td>
<td>1/72 2%</td>
<td>7/58 12%</td>
<td>0/40 0%</td>
<td>0/12 0%</td>
<td>1/21 5%</td>
<td>0/15 0%</td>
<td></td>
</tr>
</tbody>
</table>

A signal that is used frequently in active sonar systems because of its good range resolution is the FM chirp given by:

\[ s(t) = \alpha(t) \cos[2\pi f_c t + \pi \mu t^2] \]  
(6.2)
where $\mu$ is termed the chirp rate. Several examples of this signal were generated, noise was added, and all were tested with the current system and with a correlator. An example of this transient is illustrated in figure 6.11 and ROC curves are displayed in figures 6.12 and 6.13.

Figure 6.11a FM chirp.

Figure 6.11b FM chirp with noise added to simulate real class example.

Figure 6.11c FM Chirp at typical SNR.
Figure 6.12 Receiver operating characteristics for neural-net classifier trained on ten examples of FM chirp, and a correlator using the cleanest example. The SNR is 2.0 dB. The poorer results for the correlator are due to the noisy signature that was used in the correlation operation.
Figure 6.13 Receiver operating characteristics for neural-net classifier trained on ten examples of FM chirp, and a correlator using the cleanest example. The SNR is 1.0 dB. The poorer results for the correlator are due to the noisy signature that was used in the correlation operation.
As shown in figures 6.12 and 6.13 the neural classifier exhibits greater performance than the conventional approach due in large part to the poor transient signature used in correlation.

To summarize, this approach has demonstrated robust and reliable qualities in that it is not overly sensitive to parameter changes and has exhibited consistent behavior in its treatment of a wide range of class members. It is also very fast when considering that each new class only requires the addition of a single FLN without the need to re-train previous nets. This is in sharp contrast to many other systems such as the correlator above which would require continuous correlation with each signal class. This architecture also requires much less processing than, for example those outlined in chapter four and should, therefore, be much easier to implement in real time.

A wide range of signals and noise types can be encountered in real world situations. This system is by no means intended to handle all of them, but it can deal with many classes which are difficult to process by conventional means, particularly those that are temporally extended or have temporally fluctuating spectra. In general, the average class SNR can drop to about 0.5 dB and still be successfully classified, but a precise measure is difficult to ascertain because temporal augmentation helps reduce the dependence on signal strength. Signal duration is also difficult to quantify for much the same reason, but experience has demonstrated that signals from 10 - 200 ms can be processed efficiently.

Actual system output is included in the appendix.
6.4 Run-time Considerations

Although clustering and training can be performed off-line, consultation is a real-time process, requiring efficient algorithms and high-speed computers. With this as motivation all of the software was written as efficiently as possible. The FFT algorithm is designed for a specific sequence length which uses look-up tables for the trigonometric functions permitting them to be calculated at system start-up. In addition, these sequences are comprised of real data, therefore, the FFT algorithm was written for real sequences only. Both procedures provided an overall reduction of approximately 20% in execution time over a general FFT algorithm.

Calculation of the Hausdorff metric is of order $N^2$, but tremendous improvements in speed can be realized by noting equation 5.4 which is repeated here for convenience:

$$d^{**}(A,B) = \max\{d^*(x,B) : x \in A\}$$  \hspace{1cm} (6.3)

It states that the maximum of the minima from all $x \in A$ to set $Y$ is to be retained. This enables the calculation of minima to be terminated for any $x \in A$ at the point in which its distance is less than the maximum value attained thus far. In modifying the algorithm for this criterion and using peaks as starting points (greater distances) the average execution time was reduced by about 80%.

All of the tests were performed on a 486-PC running at 33 MHz, producing 8 to 10 times real-time operation. This is excellent performance considering the data is sampled at 25 kHz. Currently, sections of this procedure are being converted for implementation on a digital signal processing board which will permit true real-time operation and provide extra design latitude.
Chapter 7. Discussion and Conclusion

7.1 Discussion

The system described in this thesis is based on two major neural network paradigms, unsupervised and supervised learning. Unsupervised learning performs a self-organizing function in which power spectra are arranged into clusters predicated on a similarity measurement based on Hausdorff distance. With this completed, a feedforward network is used to determine if previously unseen inputs constitute a signal from the class in question, or if they emanate from noise. This latter task is performed by measuring the Hausdorff distance from the unknown spectral patterns to the cluster centers and mapping a sequence of these to a one-dimensional output space.

One of the strengths of this system is its real-time capability. This is facilitated by the self-organization performed before testing which serves the function of reducing the input sequences into a subset representative of attributes from all classes, and allows the determination of whether trajectories within this cluster space constitute a signal from the target class.

Another strong feature is its adaptability. The clusters can be updated in real time, thus always providing a basis set that reflects the current status of the environment or the signal classes. In addition, a vigilance factor can easily be incorporated permitting the production of new cluster centers while in the consultation mode. This presents the capability of remembering heretofore unseen input patterns, which may at some future date be employed to produce sequences for additional, yet unknown classes.
Several experiments were performed incorporating signals exhibiting substantial differences both temporally and spectrally. It has been learned that, while performance gains over conventional techniques are not significant when concerning strong tonals, they can be realized to a high degree when the transient contains many spectral components and/or varies in temporal length. This is apparent when considering the results obtained from class A and the outputs included in the appendix, these being the types of signal for which this system has been designed.

Some justification has been included concerning the concept of temporal trajectories and how they can contribute significantly to classifier performance. And it has been demonstrated how spectral clarity can be exchanged for low variance in this same context.

The Functional-Link Net has established itself as well suited for this project. It exhibits the capability of generalization, heterassociative memory, and a high learning rate. It also permits analysis to a much higher degree than the conventional feedforward network.

A summary of system characteristics and some observations follow:

1. Partitioning a signal into components increases the ability of a classifier to detect its presence in a manner proportional to the square root (on average) of the number of partitions.

2. The Hausdorff metric has proven to be an excellent choice in determining the similarity of spectra. It provides a measurement not entirely unlike that of human interpreters in that it is sensitive to pattern singularities.
3. Systems based on extended analysis of low resolution attributes can provide high success rates in the classification of protracted signals with highly variable spectra.

4. Classifiers based on neural nets provide generalizations that cannot be realized by previous techniques such as artificial intelligence.

5. Neural networks are an efficacious choice for problems of this nature particularly due to their ability to assimilate extended sequences.

6. There is nothing in the design which inhibits its use with more than one sensor.

7.2 Conclusion

This work demonstrates that adaptive pattern recognition as implemented by neural networks is a viable approach to the problem of underwater acoustic transient detection and classification. It also marks the first time that the Hausdorff metric has been used in transient classification, and validates the contention that choice of the proper metric is an extremely important step in system design. In addition, it has incorporated an uncommon technique in signal processing, that of observing reduced resolution information and determining how it fits into a larger temporal structure. Furthermore, this method utilizes a much simpler network than is commonly used, and has achieved very good results.

Currently, classification of underwater acoustic transients is usually performed by humans, so it seems only natural that neural networks should be employed for the task, but certainly much work remains to be done. Researchers are still attempting to unlock the secrets of the human cognitive process and until that goal is reached we will never be entirely certain just which pattern attributes
provide the best signal representation. Human beings are trained on LOFAR-grams augmented aurally, but future work may reveal this to be inappropriate.

As this thesis demonstrates, neural networks are providing at least a partial solution to the problem of signal classification, and the research should continue. This writer is firmly convinced that neural network technology is an appropriate choice for the detection and classification of underwater acoustic transients, but in the future some additional questions need to be addressed, as for example:

1. Is one preprocessing method sufficient for all signal classes and backgrounds or is it necessary to incorporate several methodologies?
2. Can one system architecture be utilized to classify all types of transients?
3. Can better ocean modeling help to improve performance or reduce system complexity?
4. How much benefit can be derived from multi-sensor fusion?
5. Which types of neural networks are best suited for this type of application?

These are some of the topics which need to be studied in future research before a reliable system architecture can be operational.
References


Appendix

This section presents system output during the run-time phase. All illustrations are in the form of a four column spectrogram with the neural network output displayed immediately to the right of each column.
Figure A1  Several examples of class A transients in quiet ocean. These SNRs range from 4 - 6 dB except for the last signal which is at 0.5 dB.
Figure A2  Several examples of class A transients in noisy ocean. These
SNRs are all approximately 4 dB.
Figure A3  Several examples of class B transients in quiet ocean. These SNRs are all approximately 5 dB.
Figure A4  Several examples of class C transients in noisy ocean. These SNRs range from 7 - 10 dB.
Figure A5  Ten examples of unclassified, unknown biological transients referred to as N1 and N2, including representatives from classes A and C; the order and quantity of each is listed as follows. N1 (2), class A (6), class C (6), N2 (4), N1 (4). This network was trained on two examples from N1 but classified five of them. The final N1 is in the last column and is substantially different from the others, resulting in a missed classification.
Figure A6  Same set of transients as displayed in figure A5. This network was trained on two examples from N2 and classified all four of them. Note the slight amount of false classification on class A. This is compensated for by the competitive module which examines the outputs from nets trained on class A and determines that these are not valid N2 outputs.
Figure A7  FM chirp as described in chapter 6. The SNR is approximately 2 dB.
Figure A8 Transients from classes A and C. This network, trained on class A signals, classified all examples correctly while not experiencing any false alarms on transients from class C.
Figure A9 Transients from classes A and C. This network, trained on class C signals, classified all examples correctly while not experiencing any false alarms on transients from class A.
Figure A10 Transients from classes A and C utilizing the 2-norm instead of the Hausdorff metric. This network, trained on class A signals, classified all signals (and some noise) as belonging to class A.