CHAPTER 6

RESULTS AND DISCUSSION

Simulation details like network topologies tried, learning algorithms used and the convergence behaviour and classification capabilities of various combinations of the learning algorithm and network topology are discussed in this chapter.

6.1 Input Patterns

Sea noise recordings with and without target, taken exclusively at one geographical location (viz. Cochin) were used to generate the training patterns. In this connection, 40 vectors corresponding to ambient noise alone and 32 vectors corresponding to target noise were used. During every iteration, these two sets of patterns were presented to the network in an interlaced manner.

For checking the target detection capability of the network, 248 test vectors were used. Out of these, 196 vectors were generated from ambient noise and 52 vectors from target noise which were recorded at other geographical locations (viz. Andaman and Goa).
6.2 Network Topology

Three network configurations were tried viz. 20-62-2, 20-62-1 and 20-11-2. In the first two networks, the number of neurons in the second layer is more than thrice that in the first layer whereas in the third one, the number of neurons in the middle layer is the mean of the number of neurons in the other two layers. They also differ in regard to the number of neurons in the output layer. The aim of simulating these network variants was to investigate how the above factors influence the network learning behaviour and their classification performance.

6.3 Learning Algorithms

Each of the network was separately trained using two variants of the BP algorithm viz. momentum method and exponential smoothing. In the former, for all the network configurations, the learning rate and momentum coefficient were given the values 0.75 and 0.9 respectively. In the latter case, the exponential smoothing coefficient was given the value 0.9, keeping the learning rate coefficient unaltered. These set of values were found to provide the fastest learning rate without causing oscillation.

6.4 Neural Network Training

Before the commencement of the network training, all the weights and biases were initialized to random values in
the range from -0.1 to +0.1. The training patterns were then presented to the network one after the other and the weights and biases were changed after every presentation according to the learning algorithm used. This process was continued till all the training patterns were exhausted (i.e. completion of one epoch or iteration). After every iteration, the sum of the square of the errors obtained while presenting all the 72 patterns during that epoch was computed. If it was more than the lower limit set viz. 0.01 in the present case, the iteration was continued further until this limit is reached or the prescribed number of iterations are carried out, whichever is earlier. The algorithms were run for a minimum of 1000 iterations in each trial.

6.6 Learning Curve

Sufficient number of iterations will inevitably have to be carried out before the network reaches its converged state where the error is the minimum-most. The quality and ingenuity of a learning algorithm is judged based on its capability to achieve network convergence with minimum number of iterations. The trend of the error during progressive iterations is a reliable index of the learning behaviour of the network. As the sum of square error is an aggregate of the errors during an epoch of training, a plot between this quantity and the corresponding order of iteration is called the "learning curve" for the relevant network-algorithm combination.

The learning curves for the three networks
corresponding to the two algorithms are given in Fig. 6.1 through Fig. 6.6.

Various results obtained in the simulation studies are summarily presented in table 6.1.

6.6 Observations

Since the simulation studies carried out in the present work cannot be claimed as rigorously exhaustive, a generalization of the results obtained therefrom does not bear much relevance in the strictest sense. However, the observations made within the limited scope of the experiments are as listed below:
(a) As can be seen in all the learning curves, the maximum value attained by the sum of square error is relatively moderate. This is attributed to the type of preprocessing employed.
(b) Convergence of the learning algorithm is deeply influenced by the initial weights and biases (which, in turn, are determined by the seed value given to the random number generation routine).
(c) The rate of convergence is dependent on the learning rate coefficient \( \eta \), momentum coefficient \( \mathcal{L} \), and exponential smoothing coefficient \( \beta \). For faster convergence, both \( \eta \) and \( \mathcal{L} \) should be high and for maximum smoothing, \( \beta \) should be high. But, with high \( \eta \), the algorithm either oscillates or diverges. A high \( \beta \) ensures smoother learning which consequently takes more time to converge and this is evident
Fig. G.1 Neural Network Learning Curve
Network Topology : 20-62-2       Learning Algorithm : BP(Momentum)
Fig. 6.2 Neural Network Learning Curve
Learning Algorithm: BP (Exp. Smoothing)
Fig. G.3 Neural Network Learning Curve
Network Topology : 20-62-1  Learning Algorithm : BP(Momentum)
Fig. 6.4 Neural Network Learning Curve
Network Topology : 20-62-1 Learning Algorithm : BP(Exp. Smoothing)
Fig. 6.5 Neural Network Learning Curve
Network Topology: 20-11-2  Learning Algorithm: BP(Momentum)
Fig. 6.6 Neural Network Learning Curve
Network Topology : 20-11-2
Learning Algorithm : BP(Exp. Smoothing)
# TABLE 6.1

RESULTS OF SIMULATION

<table>
<thead>
<tr>
<th>Network Topology</th>
<th>Learning Algorithm</th>
<th>Number of iterations</th>
<th>CPU time (in secs.)</th>
<th>Minimum Sum of square classification (%)</th>
<th>Misses (%)</th>
<th>False Alarm (%)</th>
<th>Classification performance grading</th>
</tr>
</thead>
<tbody>
<tr>
<td>20-62-2</td>
<td>BP (mom)</td>
<td>1000</td>
<td>10687</td>
<td>8.07884</td>
<td>60.5</td>
<td>17.3</td>
<td>22.2</td>
</tr>
<tr>
<td>20-62-2</td>
<td>BP (exp. smoothing)</td>
<td>2000</td>
<td>26000</td>
<td>6.7763</td>
<td>55.3</td>
<td>15.7</td>
<td>29.0</td>
</tr>
<tr>
<td>20-62-1</td>
<td>BP (mom)</td>
<td>2000</td>
<td>20889</td>
<td>4.27692</td>
<td>58.9</td>
<td>15.7</td>
<td>25.4</td>
</tr>
<tr>
<td>20-62-1</td>
<td>BP (exp. smoothing)</td>
<td>2000</td>
<td>24974</td>
<td>4.33416</td>
<td>55.7</td>
<td>17.3</td>
<td>27.0</td>
</tr>
<tr>
<td>20-11-2</td>
<td>BP (mom)</td>
<td>2000</td>
<td>8041</td>
<td>8.03507</td>
<td>57.7</td>
<td>16.1</td>
<td>26.2</td>
</tr>
<tr>
<td>20-11-2</td>
<td>BP (exp. smoothing)</td>
<td>2000</td>
<td>9318</td>
<td>6.59165</td>
<td>56.5</td>
<td>15.7</td>
<td>27.8</td>
</tr>
</tbody>
</table>
from the learning curves.

(d) Regardless of the network topology, the learning is totally jitter-free in exponential smoothing compared to that in the momentum method.

(e) As is evident from the learning curves, for a given network topology, exponential smoothing takes more time to converge than the momentum method.

(f) Momentum method gives better classification result than exponential smoothing.

(g) For a given number of neurons in the input layer, the classification accuracy improves with the number of neurons in the hidden and the output layer. But this is at the cost of computation and network convergence time.

(h) The minimum value of the sum of square error, which corresponds to the generalized weights and biases of the converged network, need not necessarily be a pointer to its classification accuracy.

6.7 Conclusion

The efficacy of using a neural network for target detection is well inferable from the results of the simulation which was presented in this chapter. Some hardware aspects of neural networks and a few proposals for exploiting them for sonar applications are discussed in the next chapter.
CHAPTER 7

THE NEURAL-BASED APPROACH - A RETROSPECT

A pattern recognition application (viz. detection of an underwater target) of a multi-layered feed-forward neural network was discussed in the previous chapters. It was mainly aimed as a feasibility study where the network was simulated by software on a computer. This being an off-line process, a very limited set of data recordings that were available had to be relied upon for generating both the training and the testing patterns. The compression of data due to the preprocessing method adopted here led to further reduction in the number of these generated patterns.

The efficacy of these types of applications can be conclusively established only through extensive evaluation of the actual neural network hardware with large amounts of practical real-time data. Since a neural network is as intelligent as the way it is trained, elaborate studies involving large varieties of underwater targets against different conditions of the sea background are highly scopeful and all the more relevant.

7.1 Neural Networks Perspectives and Potentials

The ever-increasing technological demands of the present-day world require innovative approaches to highly
challenging problems. Artificial neural networks with their outstanding features like massive parallelism, ability to learn, association, generalization, flexibility (plasticity), error tolerance etc. offer the promise of better solutions at least to some of these problems. The unusual and stimulating interdisciplinary nature of neurocomputing spans over neurosciences, cognitive sciences, psychology, computer science, electronics, physics and mathematics. It has already made its impact in the commercial, industrial, medical and scientific fields. The potential defence applications of neural networks include automatic target recognition (ATR), sonar and radar signal processing, vision and image processing, photonics, artificial intelligence, robotics, expert systems etc.

7.2 Sonar Applications of Neural Networks - Some Proposals

Nowadays, the science of anti-submarine warfare (ASW) is emerging as a highly competitive field. With the present-day technology, such integrated and sophisticated defensive measures like sonar systems (hull-mounted, towed array, variable depth and helicopter-mounted types), weapon controls and a variety of electronic and acoustic countermeasures come to the rescue of surface ships. Even then, highly manoeuvrable submarines, which are made more silent with nuclear propulsion and ingenious design technology and which are equipped with computer-controlled wire-guided torpedoes and long-range nuclear missiles, do reign the sea with terror. In the wide open ocean, ultimate success inevitably goes to the one who detects the enemy
first. Hence, the earliest detection and quickest localization of the foe are premier factors and in this endeavour, neural networks are capable of playing their vital role.

7.2.1 Sensor Failure Detection

A sonar system inevitably includes a hydrophone array at its wet end. It is used to transform the hydroacoustic signals into electrical signals which are further processed for detecting the target. The spatially distributed elements (sensors) of this array, by virtue of their arbitrary geometry and dimensionality, spatially discriminates the desired signal against noise and reverberation thereby enhancing the SNR. This process is called beamforming, the effectiveness of which is directly influenced by the width of the beam.

The pattern of spatial distribution of the elements in the array has a direct bearing on the beam width of the array. The failure of an element in the array adversely affects the beam pattern; more the number of failed elements, deeper the extent of this degradation. A neural network may be used for the detection of sensor failure. Signals from all the elements, after appropriate combination and preprocessing, can be fed to a neural network that is already trained with a supervised learning algorithm. The diagnostic output from the network can then be used for either manual or automatic initiation of appropriate remedial measures.
7.2.2 Beamforming

The spatial filtering is accomplished in the beamformer through a series of operations involving the weighting, delay, and summation of the signals received by the spatial elements. The summed-up output is further processed for frequency and temporal discrimination. A time-delay neural network (TDNN), therefore, can directly implement a beamformer. A TDNN has the ability to relate and compare current input to the past history of events. This enables the network to discover acoustic features and the temporal relationships between them independent of position in time so that they are not blurred by temporal shifts in the input [41].

7.2.3 Signal Enhancement

Another area where neural networks may be prospectively employed is in sonar signal enhancement for detection of signals submerged in background noise. The underlying principle upon which neural networks operate being one of pattern recognition, they may be effectively employed for SNR enhancement. Implementation of some of the existing algorithms for signal enhancement through neural network hardware might be practically feasible and worth attempting. Statistical methods, which are more accurate than backpropagation, can be used here for the unsupervised training of the network.
7.2.4 Non-Acoustic Methods of Submarine Detection

Acoustic methods of submarine detection, though widely used and are effective, can be adversely affected by topical conditions. Due to complexities of underwater sound propagation, the acoustic signals are highly susceptible to masking effects.

Non-acoustic methods attempt to detect a submarine by sensing the perturbations it creates in the surrounding physical environment. These disturbances, however, must be measurable and separable from the background of similar naturally occurring disturbances. Phenomena such as wakes, internal waves & disturbances and magnetic & thermal anomalies which are generated by a moving submarine are the important useful parameters in this regard. The perturbations in these being very feeble, devices like superconducting quantum interfacing device (SQUID) are used for measuring them.

Since the characteristics of the submarine and the pattern of perturbations it creates in the vicinity can be mutually correlated, any detection algorithm has to search for these expected signature patterns. A complex neural network stored with all such signatures pertaining to different classes of submarines can carry out a nearest-neighbour search in the sensor data presented to it. These patterns have to be essentially made invariant to the ambient pattern of the sensed parameter and the speed of the searching platform. Optical neural
networks, with their inherently high speed and the potential for massive interconnectivity, stand as the best bet for this application.

7.3 Neural Network Hardware

Neural network models in software generally consist of many very densely interconnected processing elements, each of which performs a simple computation in parallel with its neighbours. These models and the learning algorithms are computationally intensive on general-purpose computers. However, because of the computational simplicity of the basic processing element, neural networks are implemented on special-purpose massively parallel hardware which can vastly outperform implementations on even the most powerful serial computer. The neurocomputer hardware has been an essential ingredient for the development of practical applications of neural network technology.

Only VLSI processor arrays can realize the true computing potential of massively parallel neural networks. This realization follows one of the two approaches: (1) general purpose neurocomputers that consist of programmable processor arrays for emulating a range of neural network models (2) special purpose neurocomputers that are dedicated hardware implementations of a specific neural network model. Any programmable neurocomputer is order-of-magnitude slower than its directly fabricated hardware version which has got very poor
generality. In fact, far more dedicated special-purpose neural network hardware is being developed than programmable neural processors [19]. The fabrication technology is broadly classifiable as electronic, optical and electro-optical implementations where the second and third are rapidly outgrowing the first one. Researchers now consider molecular devices, still very much in their infancy, as a new basis of neurocomputers.

Silicon implementation can be considered the first step toward large neurocomputers. While considering the possible architectures for the basis of a neurocomputer, the important design issues are parallelism, performance, flexibility - and their relationship to the silicon area. These issues, which are directly influenced by the node complexity, lead to radically different systems which range from simple traditional RAMs to programmable processors and special-purpose dedicated hardwares [18]. For example, "80170 ETANN" which is a VLSI electrically trainable artificial neural network developed by Intel Corporation, USA, is the fastest device commercially available as on date [43]. It has 64 neurons with a total of 10,240 programmable analog weights. They perform calculations, known as "connections" simultaneously, resulting in a performance of more than two billion connections per second (A connection is a multiplication-and-sum operation). Newer and more powerful neural network chips are also being developed by this company.

The increased circuit density possible in VLSI makes it most suited for neurocomputer implementation on silicon.
But, the main design problems encountered here are: massive interconnectivity of the processing elements, complex adaptivity for the synaptic weights, realization of learning algorithms, network size & geometry, processing and communication speeds and data representation. The number of cells (i.e., neurons and synapses) in a fully interconnected neural network grows phenomenally with the number of neurons. Also, the area required to route connections and to contain the average length of interconnections increases at unacceptable rates as more processing elements are added. Reduction of size of the basic cells even despite a loss of precision, wafer-scale integration and three-dimensional integration are only some of the answers to these problems. For better solutions, the architectural properties of VLSI have to be further explored and exploited thoroughly.

Analog optoelectronic hardware implementation of neural nets, which was first introduced in 1985, has many advantages over the electronic implementation. Primary among these is that the optoelectronic (photonic) approach combines the best of the two worlds: viz. (i) the massive interconnectivity and parallelism of optics and (ii) the flexibility, high gain and decision-making capability offered by electronics. It seems more attractive to form analog neural hardware by completely optical means where switching of signals from optical to electronic carriers and vice versa is avoided. However, in the absence of suitable fully optical decision-making devices, the capabilities of the optoelectronic approach remain quite attractive and stand
competitive with other approaches when considering the flexibility of architectures possible with it [42].

Molecular electronics is a vision that promises to solve all the technological problems of neural networks. Its self-building and self-organizing capabilities in three dimensions offer prospects of huge neural networks occupying compact space and rendering highly superior performance. In reality, however, the development of biological molecular electronics will be very slow and the field of physical molecular electronics is just starting with the first test structures [19].

7.4 Conclusion

Biological neural nets were evolved in nature for one ultimate purpose: that of maintaining and enhancing survivability of the organism they reside in. Embedding artificial neural nets in man-made systems, and in particular autonomous systems, can serve to improve their survivability and hence reliability. Neurocomputers can be expected to play an important role in the modeling and study of highly complex systems and problems with enhanced flexibility and speed offered by integrated optoelectronic techniques.