Florida State University Libraries

Electronic Theses, Treatises and Dissertations The Graduate School

2005

Neural Network Based Prognosis for Twodimensional Tumor-like Growth

Vidya Rajagopalan

Follow this and additional works at the FSU Digital Library. For more information, please contact lib-ir@fsu.edu

THE FLORIDA STATE UNIVERSITY

FAMU-FSU COLLEGE OF ENGINEERING

NEURAL NETWORK BASED PROGNOSIS FOR TWO-DIMENSIONAL TUMOR-LIKE GROWTH

By

VIDYA RAJAGOPALAN

A thesis submitted to the
Department of Electrical Engineering
In partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Summer Semester, 2005

The members of the Committee approve the thesis of Vidya Rajagopalan defended on April 25, 2005.

Simon Y. Foo
Professor Directing Thesis

Namas Chandra
Committee Member

Anke Meyer-Baese
Committee Member

Uwe Meyer-Baese
Committee Member

Approved:

Leonard Tung,
Chair, Department of Electrical and Computer Engineering

Ching-Jen Chen,
Dean, College of Engineering

The Office of Graduate Studies has verified and approved the above named committee members.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor Dr. Simon Foo for his constant guidance and advice. I would also like to thank him for being my advisor during the course of my degree.

I would like to thank Dr. Uwe Meyer-Baese for his encouragement and support during the research. I am grateful to Dr. Anke Meyer-Baese for her support and guidance in these two years.

I would like to acknowledge the help rendered by my friends at the Machine Intelligence Lab, whose valuable suggestion were instrumental in the successful completion of this thesis.

I feel greatly indebted to Dr. Namas Chandra for the many discussions which reinforced the direction of this work on several occasions. I would also like take this opportunity to thank Dr Usha Chandra who made sure I seldom realized that I was miles away from home.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

In experimental and computational science and engineering, there are a host of practical problems where some tell-tale signs appear well before a critical event of interest surfaces, e.g., the occurrence of tumor is preceded by gradual change in cell density in the neighborhood. If these signs are detected then it is possible to prognosticate the severe conditions that are likely to occur. In this case the tumor can be predicted well before it actually appears and appropriate counter measures can be taken at an earlier stage with a higher degree of success. The effectiveness of such a system is gauged by how early the changes are detected and the accuracy of the predictions. The feasibility of using a combination of wavelets and neural networks as a prognostic tool is explored in this thesis.

There are two phases to implementing a prognosis system for tumor-like growth. The first phase involves characterizing the data so that the tell-tale signs are detected accurately. The next phase involves choosing an appropriate predictive tool to accurately model the growth of the tumor. If this system is built with some learning capability then it could emulate if not surpass the acumen of an expert. Wavelet analyses of incoming signals serve as a preprocessing tool while the neural network is used as the predictor. This wavelets-neural network prognosis tool is directly applied to detect the development of a tumor, well before an expert human eye can perceive the problem. Computer simulations are performed using real and simulated data sets, and conclusions are drawn from the results.

CHAPTER 1

INTRODUCTION

The successful treatment of most critical illnesses is based on early and accurate detection. The use of technology in medicine has changed the face of medical practice completely. But most of this technology is based on detecting the problem only after it has occurred. Even with the availability of superior computing and imaging capabilities, the concept of "*prevention is better than cure*" has not caught on in medical practice.

This thesis explores the possibility of 'Computer Aided Prognosis'. The objective is to design a neural network based predictor that could be used as a prognosis system. The prognosis problem is sub-divided into two units. The first part of the thesis deals with predicting the location of a tumor given a series of mammograms. This involves the selection of an appropriate preprocessing technique and suitable neural network architecture. The second part is a comparative study on the performance of three popular network architectures for the prediction of tumor recurrence. The block diagram of the prognosis system is shown in Figure 1.1.

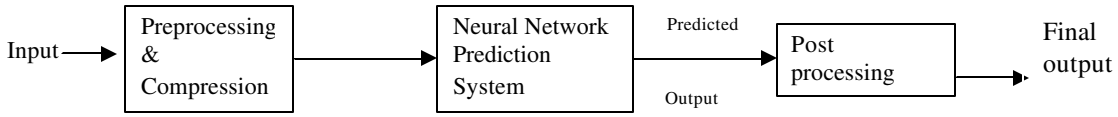


Figure 1.1: System Block Diagram

A prediction system such as the above can be used to monitor 'high-risk' patients. The prognosis system foretells if a particular patient is likely to develop a tumor given the series of mammograms from the annual health examinations. The system uses simulated mammograms in which a specific white area represents a tumor. Once a tumor-like has been predicted in patient by such a system, biopsy techniques can be used to study the problem area further. This facilitates appropriate preventive measure for patients and also to treat recurring and non-recurring tumors differently.

Current Work

Neural Networks have been widely used in medicine for various diagnostic applications. The concept of prognosis using neural networks is a relatively new concept. *Cowburn et al* (1996) [20] have explored the possibility of using neural networks to predict heart failure in patients with heart problems. Laura *and Armando (1999)* [20] showed the advantages of using neural networks to for classification and recognition of digitized medical images.

Organization of Thesis

The thesis is organized as follows. Chapters two and three provide a brief background on wavelets and neural networks respectively. Chapter 4 presents the problem formulation, approach and solution to prognosis using simulated images. Chapter 5 discusses the problem of prognosis using data from biopsy. The results are analyzed and discussed in chapter 6. The final chapter summarizes the thesis and presents the scope for future work.

CHAPTER 2

WAVELETS

Wavelets are basis functions that are limited duration version of a prototype wave called *'mother wavelet'*. They have an average value of zero. Wavelets transform is a windowing technique that provides a varying time and frequency resolution for analysis of non-stationary signals. The signals are broken down into frequency bands small enough to be considered stationary and are then analyzed in time.

Continuous Wavelet Transform

It is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the mother function (?). The continuous wavelet transform coefficients for the signal *s(t)* (with a translation index of *a* and dilation index *b)* are given by

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} s(t)\, \psi\left(\frac{t-a}{b}\right) dt \qquad (2\text{-}1)$$

Discrete Wavelet Transform

Continuous Wavelet transform generates a large amount of data. If the mother wavelet is dilated and transformed according to dyadic (powers of two) then the transform is called *Discrete Wavelet Transform (DWT).*

$$f(t) = \sum_{k} c_{j_0,k}\, \phi_{j_0,k}(t) + \sum_{j=j_0}^{\infty} \sum_{k} d_{j,k}\, 2^{\frac{j}{2}}\, \psi(2^j t - k) \qquad (2\text{-}2)$$

where j is the dilation index , k is the translation index and F (t) is the scaling function.

Applying DWT to a signal is equivalent to passing it through a set of low pass and high pass filters of various cut-off frequencies. Wavelet analysis yields two sets of coefficients: *Approximate Coefficients cA* (obtained by passing the signal through a low

pass filter) and *Detail Coefficients cD* (obtained by passing the signal through high pass filter). The approximate coefficients contain the identity of the signal and the detail coefficients contain the nuances of the signal. Multilevel decomposition (as shown in figure 2.1) is done by applying wavelet transform to successive approximations.

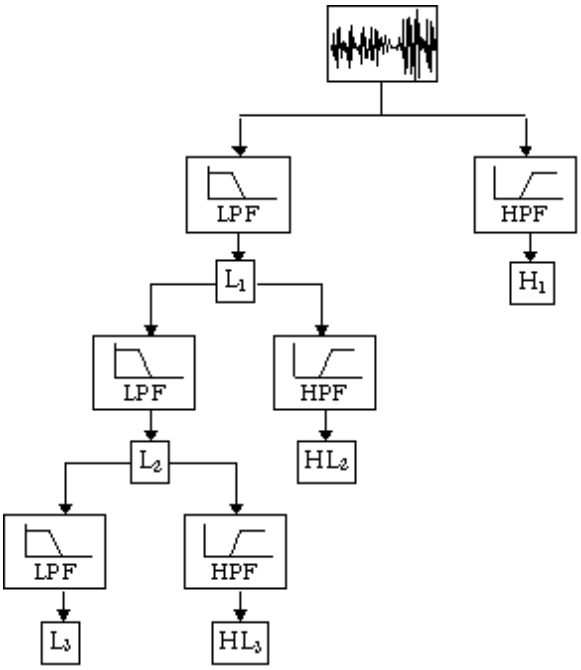


Fig 2.1: Wavelet Decomposition Structure

Daubechies Wavelet Family

The performance of wavelets for a particular application depends largely on the choice of wavelets and the choice of number of levels of decomposition. In this thesis, wavelets are only being used in preprocessing. They are used for compression and noise removal. Biorthogonal filters are popular for this application. But in this case, the picture quality after reconstruction is not a major criterion but the actual application requires that the image be highly compressed without losing much information.

Daubechies wavelets are named after their inventor Ingrid Daubechies [1].

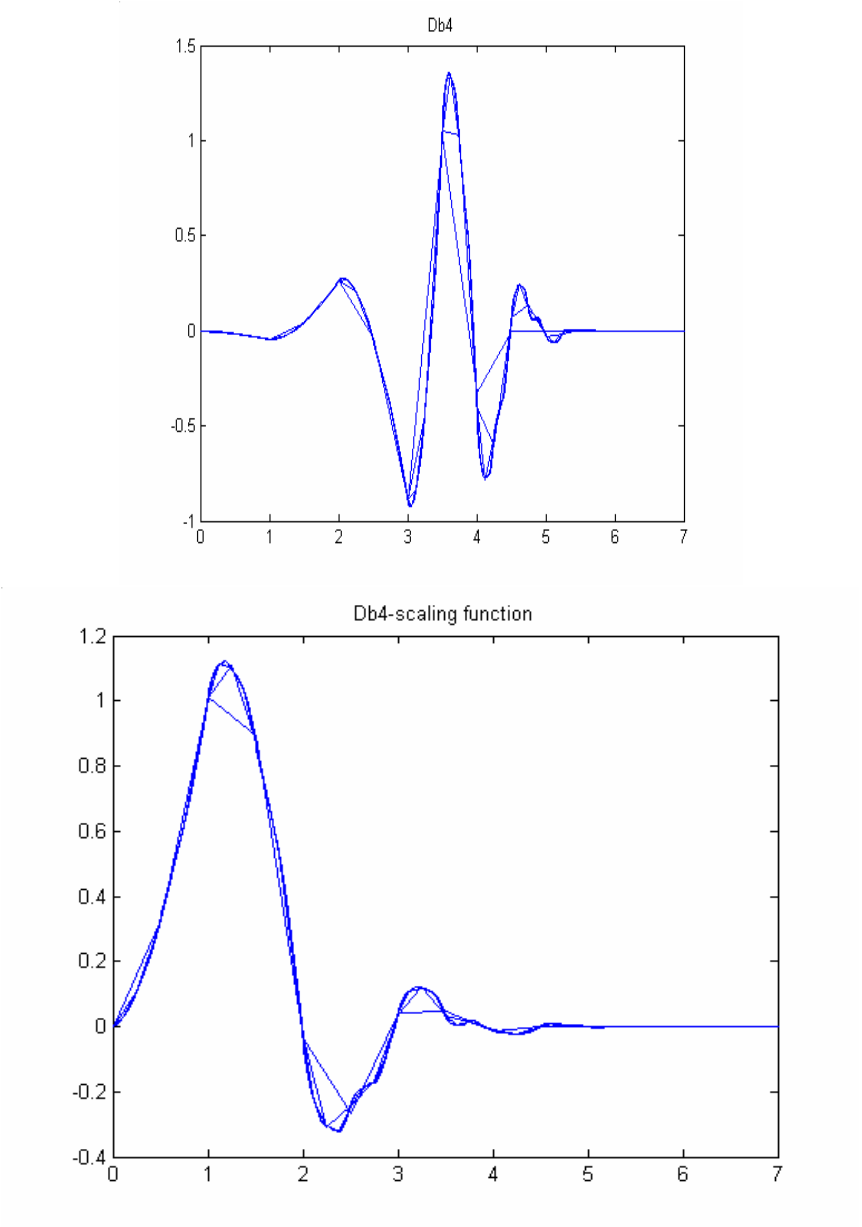


Figure 2.2: Scaling and Mother Function of DB4

Daubechies wavelets are denoted as dBN where N indicates the number of vanishing moments. Any Daubechies wavelet has compact support in the interval 2N-1. Most dBN wavelets are not symmetrical.

The scaling function and the mother wavelet are the two lowest order functions, the recursions of which are used to generate the wavelet family [21]. The equation of the *scaling function* of a Daubechies wavelet is given by

$$\phi(t) = \sum_{k=0}^{3} h_k \phi(2t - k) \tag{2-3}$$

where, h_k are the *filter coefficients*. For dB4 the filter coefficients are:

$$\frac{(1+\sqrt{3})}{4}, \frac{(3+\sqrt{3})}{4}, \frac{(3-\sqrt{3})}{4}, \frac{(1-\sqrt{3})}{4}.$$

The *wavelet equation* (involving the filter coefficients and scaling function is

$$w(t) = \sum_{k} (-1)^k h_{1-k} \phi(2t - 1) \tag{2-4}$$

For dB4 it is given by

$$w(t) = h_3 \phi(2t - 2) + h_2 \phi(2t - 1) + h_1 \phi(2t) + h_0 (2t - 1) \tag{2-5}$$

Two Dimensional Wavelet Transform

When one dimensional (1-D) wavelet transform is used on images, the directionality of the data is lost. One dimensional transform simply acts as a "mathematical microscope ". When required numbers of dimensions are used wavelets provide a useful skeleton for image analysis.

Two dimensional (2-D) wavelet transform requires quad-tree filtering structure. It requires a set of high and low pass filters for the row data and another for the column data. The row and column data can be filtered separately one after another to yield four sets of coefficients i.e. approximate coefficients and three sets of detailed coefficients,

one for each orientation (vertical, horizontal and diagonal). 2-D wavelet transform uses multiresolutional analysis (Figure 2.3) on the signal by passing the input signal through the filter bank more than once.

Consider an image *i* and a wavelet *w*. Both can be subjected to translations (t), rotations (?) and dilations (d). Any wavelet when translated, rotated or dilated still gives another wavelet *w* (t, T, d), thus generating a whole family of wavelets. When the image *i* is decomposed by the wavelet *w*, the outcome is a product of the image and the wavelet w (t, T, d).

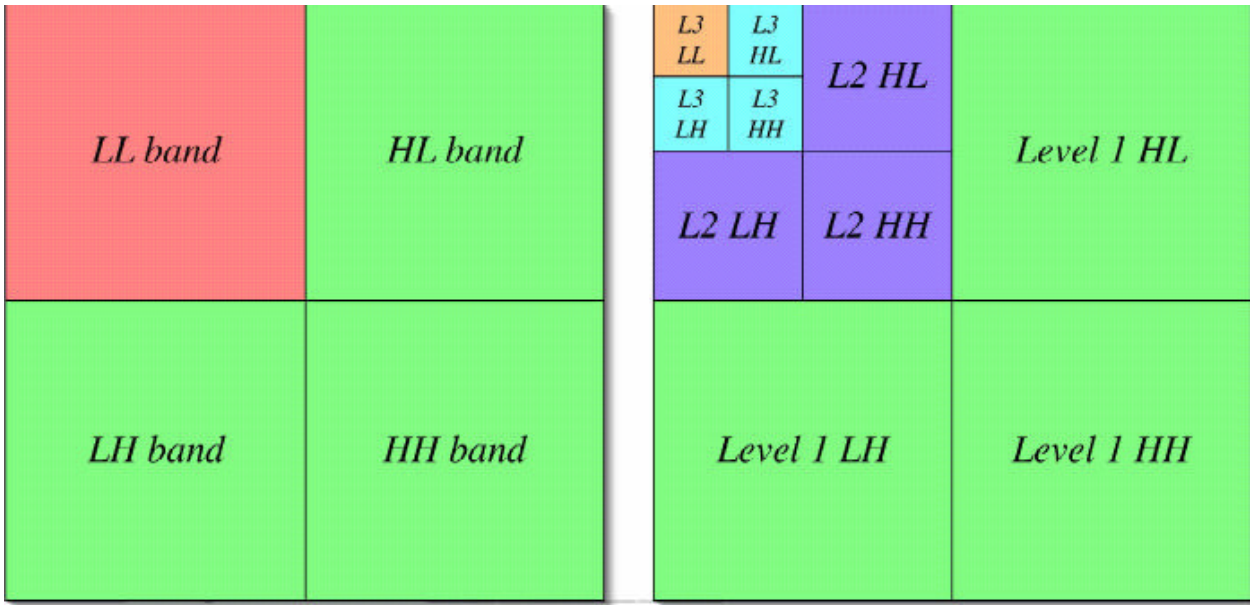


Figure 2.3: 2D transform

Thus 2-D wavelet transform can be interpreted as the magnification of the image by a factor proportional to the inverse of the dilation the direction of orientation being given by the rotation factor.

Wavelet Reconstruction

Signals are reconstructed from their coefficients by using inverse tree filters. When the z-transform of the reconstructed signal exactly matches that of the original signal (i.e. the

signal before decomposition) *perfect reconstruction* is said to have occurred. In order to obtain perfect reconstruction a number of trade-offs are to be achieved between frequency and time components. One of the most important conditions in perfect reconstruction is aliasing cancellation.

Let H_0 be the LPF and H_1 be the HPF of the decomposition filters. Let G_0 and G_1 be the corresponding filters at the reconstruction end. The general setup of the reconstruction filters is shown in Figure 2.4. The problem of perfect reconstruction is the problem of choosing the appropriate filters for H_0, H_1, G_0 and G1.

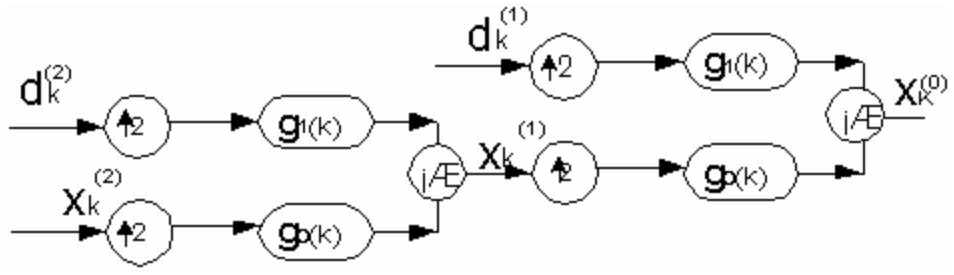


Figure 2.4: Reconstruction Filter Setup

Let S (z) is the z-transform of the original signal and R (z) is that of the reconstructed signal. Aliasing cancellation requires that S (-z) be zero which forces

$$H_0(z)G_0(z)+H_1(-z)G_1(-z)=0 . \qquad (2\text{-}6)$$

For identical z transforms of original and reconstructed signals

$$H_0 (z)\, G0 (z) + H_1 (z)\, G1 (z) = 2 \qquad (2\text{-}7)$$

One of the major hindrances when achieving perfect reconstruction is that wavelets are not immune to spatial or temporal shifts to inputs. This causes aliasing making DWT perform inconsistently when used to detect key features in images. Another common problem is the poor direction sensitivity of wavelets thus causing DWT to perform poorly when used in directionally selective applications.

Wavelet Compression

Wavelet Compression is based on the method of multiresolutional analysis. Unlike traditional compression algorithms wavelet method analyses the image as a whole. This helps to improve the compression ratio and also the quality of the compressed image. Wavelet compression is a non-uniform method where all parts are not compressed with the same ratio. Some parts are compressed more than the others. In this method the image to be compressed is first digitized. It is then decomposed to a set of wavelet coefficients. Then *quantization* or *thresholding* is applied to the signal. The compressed coefficients can then be reconstructed to give the image.

Thresholding

Most of the image coefficients are in general close to zero. A very small number of coefficients have appreciable value and are required to reconstruct the image. In this method a threshold value is chosen for the coefficients. Thresholding is of three types. In *hard thresholding* all coefficients whose absolute value is below the threshold value is set to zero. Another method is soft thresholding where if the coefficient is below the threshold value they are set to zero. All others are replaced by the difference between their absolute value and the threshold. The third method is *quantile thresholding* where the coefficients are arranged in descending order of their value. A percentage p is then selected. The lowest $p\%$ of values is then set to zero.

Quantization

In this method every coefficient is represented by a fixed number of bits. The most common method to achieve this is by rounding all the coefficients to the nearest integer. In this method the mapping from the coefficients to the integers is not one to one. This increases the errors during reconstruction. Another method is to normalize the coefficients and then to round them to integers.

CHAPTER 3

ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) can be algorithm or an artificial device that implements the mathematical model inspired by the biological neural system. They can be made up of a number of interconnected processing elements operating in parallel. The essence of neural networks is that they process information collectively. The most important function of the human neural system that the ANN tries to emulate is the ability to learn from past data and extrapolate the knowledge to new data.

The use of advanced imaging techniques in medical practice has created a need for a robust analysis tool that can perform reliably in spite of the size, noise and inconsistency associated with medical images. The major advantage of neural networks is that they compute and approximate a solution even in situations where the input is incomplete, noisy, inconsistent and probabilistic. Their ability to learn from the past data and apply it in solving unknown problems is the main motivation in using neural networks in this application.

The Neuron

The biological neuron is the basic cell that makes up the nervous system and the brain. The neurons are interconnected by small electrical gaps called *synapses*. The electrical signals of the brain are processed in the neuron's cell body and sent to the synapses through the *axons* and *dendrites* (neuron endings). The *synapses* distribute this signal to the neighboring neurons.

The basic functional unit of an ANN is called a *neuron*. It accepts *n* number of *inputs* and delivers one output. Each input(x_1 *to* x_n) has a *weight* (w_1 *to* w_n) associated with it. The *net weight* (u_i) of each neuron is the dot product of all the inputs and their corresponding weights. The net value is then applied to an *activation function* ($f(u_i)$). If output of the

function (y_i) exceeds the *threshold* value of the neuron then the neuron is turned on or '*fired*'. The signal is then passed on in the direction of propagation. The model of a typical artificial neuron is shown in the figure 3.1.

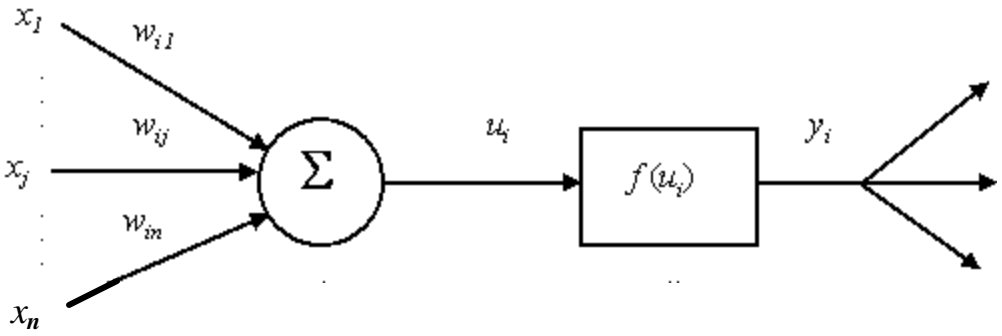


Figure 3.1: Structure of an Artificial Neuron

Architecture of Neural Networks

The basic structure of a neural network consists of one or more layers of neurons. The *input layer* distributes the network inputs to the subsequent layers. It is followed by one or more *hidden layers.* The functioning of the hidden layer is invisible to the user and hence the name. The computation is usually done by the hidden layers and the output is passed on to the *output layer.*

When the network is first designed the weights are initialized at random. A *training input* set is then presented to the network. With each presentation of training input the network adjusts its weights using a training algorithm till the required output is obtained. This point is called *convergence* of network during training and the entire process is called *learning or training* of the network. Each presentation of input to the network is called an *epoch.* Once the network has been trained, the network can be used to compute the output for unknown or *test inputs.*

Neural Networks can be classified into two major categories based on their modes of learning. The two categories are:

Supervised Learning

The training data consists of two sets, namely the input set and the output or *target* set. The network learns from a *teacher*. The network learns by adapting the weights until the output computed by the network matches the desired output. This method is also known as *reinforcement learning*. It is very popular for forecasting and time-series prediction applications.

Unsupervised Learning

There are no explicit teachers or target outputs in this method. The network learns to find patterns and similarities in previous data and applies it to the new input. Unsupervised learning is similar to the way the human brain functions. It is widely used for pattern recognition and classification. They are capable of solving more complex problems as compared to the supervised networks.

Based on the architecture of the network, neural networks can be classified as mentioned below.

Feed Forward Networks

These are the most common type of neural networks. In this type of network, the signal propagation is only in the forward direction and there is no physical feedback. The network is made of a number of layers of neurons namely the input layer, one or more hidden layers and the output layer. The structure of a feed forward network is shown in the figure 3.2. Feed forward networks are called *universal approximators*. It has been proved that with appropriate number of hidden layers, the feed forward networks can be used to approximate any function. This makes the feed forward networks popular for applications such as time series prediction.

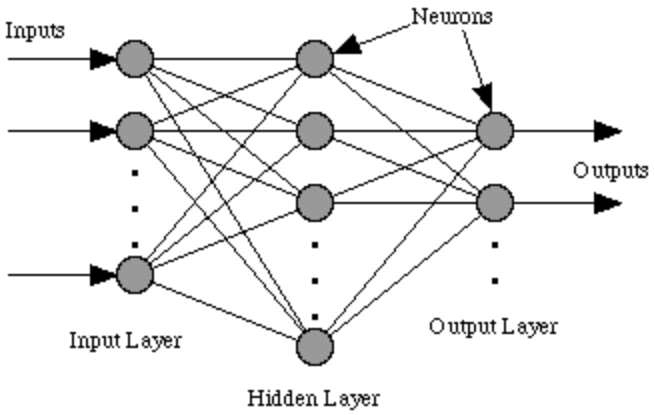


Figure 3.2: Feed Forward Network

The feed forward network functions as follows: each neuron receives the signal from those in the previous layer. The weighted sum of all the inputs to the neuron is then computed. This is then passed on to the activation function of each neuron. The output of the function is compared to the pre-defined threshold value. The output of this function is then passed on to the next layer if the calculated output is greater than the threshold value. The input is applied to the first layer. The signals are then propagated till the final output is reached at the last layer. During the training stage, the computed output is compared to the required output. The error is calculated and the error is propagated backwards. The weights are then adjusted according to the error. The next set of input is then presented to the network and this cycle continues till the error is less than or equal to the target error. This method is called *error back propagation* or simply "*back prop*". The final weights are then stored. This is used to calculate the output when the test inputs are presented.

The most common design parameters for the training of a feed forward network are number of inputs, number of outputs, learning rate, target error and number of hidden layers. Learning rate specifies the amount by which the weight changes each time it is updated. If the learning rate is too small, the network will train very slowly. If learning rate is too high, large oscillations will occur and the network tends too become unstable. Most often learning rate is found in an ad hoc manner by trying a few values until a convenient value is determined. The learning rate in a network need not be fixed. It can

be adapted according to the difference between the computed output and the actual output. This method is called *adaptive learning*. In some situations the network training may never converge and computation does not stop. In order to check such a situation, a maximum number of epochs can be specified, at which training will be stopped if convergence does not occur. The selection of the number of hidden layers decides the efficiency of the network. The number of hidden layers is given by square root of number of input layers.

There are many algorithms which are used for adapting the weights according to the error propagated by the network. These are called *learning methods*. The Levenberg-Maraquadt is the most popular and efficient of these methods. It is an optimization technique used to find the minimum of a multivariate function. It is an iterative method that is a combination of the steepest descent and the Newton's method. This method provides the fastest training among the available training methods. One drawback of this method is that sometimes the intermediate calculations result in very large matrices and hence the algorithm is highly memory intensive.

Radial Basis Function Networks

Radial Basis Functions networks (RBF) are similar in architecture to the feed forward networks. They were introduced by Broomhead and Lowe (1988) and Poggio and Girosi (1989). Their functioning is based on the locally tuned response in human neural system i.e. as seen in the auditory system. They are based on a hybrid learning mechanism where the hidden layer learning is unsupervised but the output layer uses supervised learning.

Similar to feed forward networks, RBF networks are multilayer networks. The network is designed as surface approximators for multidimensional spaces. The network tries to find the surface in a high dimensional space that best matches the input data. The architecture of the RBF network is shown in figure 3.3. They have an input layer, an output layer and only one hidden layer. The input layer is associated with the hidden layer by a nonlinear radial basis function such as a Gaussian function. The hidden layer is associated with the

output layer by a linear function. The neurons of the hidden layer have a *center* associated with them. The distance from the input point to the center is calculated. The basis function at the hidden layer processes the input data based on their distance from the center of the function. This is then propagated to the output layer which calculates the weighted sum of its inputs.

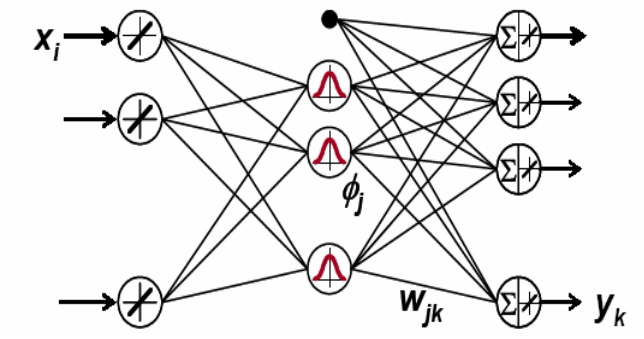


Figure 3.3: Radial Basis Function network

Learning Algorithm of a RBF Networks

The RBF networks find local approximations to a non-linear input to output mapping. The number of input layer neurons is determined by the size of the input vector and the number of output neurons depends on the number of patterns. The number of hidden neurons is usually decided by the problem being solved.

Consider a RBF network with N hidden neurons. Let the input vector be x. The radially symmetric function $\phi(.)$ is usually a Gaussian function with a standard deviation or *width factor* of σ. The output at the j^{th} hidden neuron is given by

$$\phi_j(x) = \exp(-\|x - m_j\|/\sigma_j) \qquad (3\text{-}1)$$

where m_j is the location of the center represented by the j^{th} hidden neuron. The weight connecting the j^{th} hidden neuron and i^{th} output node is given by w_{ij}. The output of the i^{th} output neuron is then given by

$$f_i(x) = \sum_{j=1}^{N} w_{ij}\phi_j(x) \qquad (3\text{-}2)$$

A typical RBF network training process is as follows. The center m_j is chosen for all the *N* hidden neurons. The weight vectors are initialized to random values with very small magnitude. The value of the hidden neurons is calculated from the following relationship:

$$? _i ? \exp(\frac{? d(x_i m_i K_i)}{2}) \qquad (3\text{-}3)$$

The operator $d(x_i m_i)$ denotes the *Mahanoblis distance* and K_i is the *shape matrix* given by $K_{i\,jk} ? \frac{h_{jk}}{? _j * ? _k}$ where h_{jk} represent correlation coefficients. If j = k then h_{jk} is 1 else $\left|h_{jk}\right| ? 1$. The value of the output neurons is calculated according to the equation for $f_i(x)$. The error between the computed output and the target output is then calculated. The weights are then updated based on the calculated error. The process is repeated till the error is less than or equal to the target error.

Strengths and Weaknesses of RBF networks

RBF networks often train faster than most supervised networks. The single hidden layer in the RBF network is as efficient as multiple hidden layers, hence it saves computation. If the radial basis functions have to be changed, then the change can be calculated from a set of equations without having to retrain the network all over again.

On the other hand, the RBF network can get caught in a local minimum like most supervised networks. The choice of parameters like the center and radius are very critical in arriving at the exact solution. The various parameters associated with a RBF network can make design and implementation of the network quite complex.

Recurrent Networks

A feed forward network is comparable to the combinatorial circuits. Their output at any time is determined by the inputs at that instant only. They are not affected by past states of the network. This makes the networks inappropriate for use with dynamic inputs. In

order to overcome this problem, recurrent networks can be used. They are multilayer networks with feedback. They are also use nonlinear neurons and hence they are nonlinear and dynamic systems. Recurrent networks are of many types, those with a single hidden layer and those with multiple hidden layers.

Elman Network

Elman network is a single hidden layer recurrent network. They are also known as *Simple Recurrent Network.* They were introduced by Professor Jeff Elman [4]. The network consists of three layers namely input layer, hidden layer and output layer. The structure of the network is shown in figure 3.4. The output of the network at time *t* is a function of the input at *t* and output at *(t-1).* The feedback is from the hidden layer to the input layer. There is a *context node* in between the hidden node and the input node. The context units act as low pass filters. The data feedback from the hidden node is through the context node. The connection has a standard weight of 1. When an input is applied, signal flow occurs in the standard feed forward method. A learning rule such as *backprop* is used. The copy of the previous state of the hidden node is sent to the context unit before the present rule is applied. The network thus maintains a sense of context. This allows it to distinguish between identical patterns that occur over time. The Elman network combines the advantages of both recurrent networks and learning by back propagation.

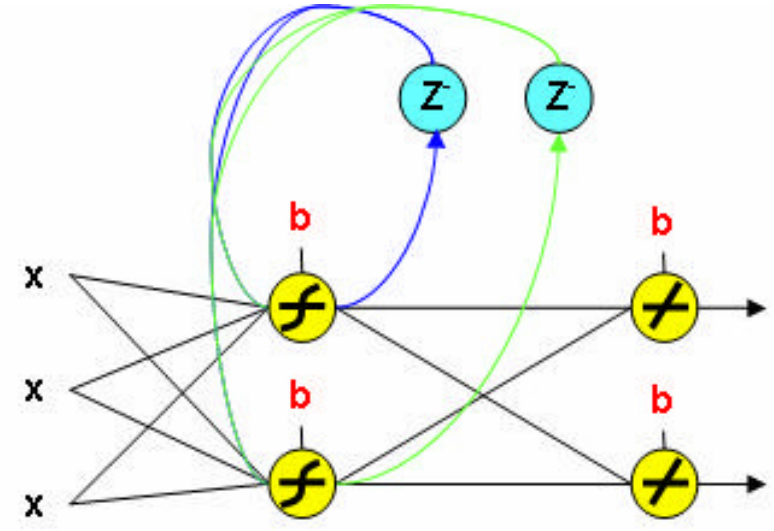


Figure 3.4: Structure of Recurrent Network

CHAPTER 4

PROGNOSIS OF MAMMOGRAPHIC IMAGES

This part of the thesis deals with detecting the location of a tumor-like and forecasting its growth pattern. The input is a series of images giving the changes in the data set over time. The output is another series of images that show the growth of the pattern over time.

Image Database

Like all x-rays of other body parts a mammogram shows soft but dense tissues of the female breast. The breast tissue is not uniform for all patients. The normal mammogram of each patient looks different. Mammograms sometimes contain random white areas that are quite normal and need to be ignored. A tumor (which shall be referred to as area of interest) can be hidden amongst a number of harmless white areas. It is required that the database contain various types of tumors.

The simulated database is made up of about 40 image sets. Each set contains 50 individual images. The images are simulations of mammograms. Each of the image set contains 50 mammograms taken over a period of time specifically the period showing the various stages of growth o a tumor. Each image set are the mammograms of the same person taken at different time instants of uniform time interval. Hence the data base contains the images from 40 such people.

Each image is of size (64 x 64). The background of the image is black like the general background of mammography images. The first image in the database shows the first mammogram taken. A random number of white spots are incorporated at random points to signify variation in the density of flesh which is not abnormal. A random point is then chosen to grow the area of interest. Successive images show the various stages of growth of the area of interest. The random white spots stay the same in the following images. There are no new random spots added. The last image in the dataset usually shows the

fully grown area of interest. The growth pattern of the area of interest in each of the 40 data sets are varied in order to maintain the seemingly irregular growth pattern of a tumor in human body.

Choosing the Growth Model

While finding the exact pattern of growth of a tumor is still a challenge to the medical community, it is evident it is not exactly a random process. A number of mathematical models have been developed to represent tumor growth. In this work some of the classical tumor growth models [12] have been used along with general mathematical equations (exponential, logarithm, nearest neighbor algorithm) and a combination of any two functional forms.

Gompertz Model

This method suggests an exponential growth when time (t) is small and saturation at larger t. V is the volume of the tumor and V_0 is the volume at time t_0 (time at which tumor is first observed):

$$V ? V_0 \exp(\frac{A}{B}(1 ? \exp(? Bt))) \qquad (4\text{-}1)$$

Jansson-Revesz Equation

Cruywagen et al applied the Jansson-Revesz equations to tumor growth. These are simple equations that describe the growth of a tumor under relatively straight forward conditions.

Sherratt and Chaplain Model

This model gives the growth rate of avascular tumors.

Let x denote the spatial coordinate and t the time. The growth factor

$$c(x,t) ? D_c \frac{?^2 c}{?x^2} ? k_1 c_0 [1 ? ? (p ? q ? n) ? k_1 c ? k_2 pc \qquad (4\text{-}2)$$

where p, n and q are the cell densities. The parameter is defined as $? ? (0,1]$. $C_{0\ is}$ the concentration in the absence of a tumor.

Adams Model

In his paper tiled "A simplified mathematical model of tumor growth" J.A Adams [12] describes tumor growth as an ordinary differential equation giving the reaction-diffusion equation of tumor growth.

The other notable works in modeling of tumor growth are those by Ward and King [12] and Bryne [12] and Chaplain [12]. Travis et al [12] have developed models that incorporate the effect of ongoing treatment on tumor growth. This model is not considered because this work assumes that no treatment has been started.

The program is allowed in choose at random (with uniform distribution) from one of the above tumor growth models or the basic growth model such as exponential or logarithmic or a combination thereof. The area of interest is a white spot against the black background. The maximum size to which the spot can grow is restricted to (15 x 15). The spot is allowed to be some shades of gray to keep in touch with the varying color of natural tumors. Consecutive images in each data set show the growth of the tumor over successive time intervals. The database shows a wide variety in the growth pattern, size and color of the spot in order to account for the diversity in actual tumors.

Data Preprocessing

The total amount of data to be handled in this application is enormous. The software used is MATLAB. As MATLAB is an interpreter, it is slower than most computer languages. In order that the dimensionality of the data is reduced it is necessary that the data be processed appropriately before any sort of computing or prediction is done on it.

The image data is a four dimensional array of pixel values from 0 to 256. A pixel value of zero corresponds to white and pixel value of 256 corresponds to black and any value in between corresponds to various shades of gray. The data was stored as unsigned integers which reduced the number of bits occupied by the database by half.

The image was then transformed by using 2-D discrete wavelet transform. It was decomposed down to 3 levels using Db4 wavelet. This wavelet was chosen over *symlet 4*. This was chosen because the details that were lost due to compression using Db4 did not affect the accuracy of prediction. The transform yields four types of coefficients: approximate coefficients, the horizontal, vertical and diagonal detail coefficients. Decomposing the signal to three levels yields one set of approximate coefficients, and three sets each of horizontal, vertical and diagonal detail coefficients.

The type of wavelet compression used in this work was *quantile compression*. This method provides the flexibility of choosing the coefficients that are to be retained unlike the other two. The approximate coefficients were retained completely. Out of the three sets of detail coefficients only the highest level (level 3) were retained in each case. This provided a compression ratio of 1: 100. The information retained after the compression was adequate for the purpose of prediction within the admissible error limits. This compression also retained the necessary features required to detect the area of interest in the image set. The compression ratio was chosen on a trial and error method where it was necessary to maintain a balance between information loss due to compression and the amount of data that could be processed by the system at a reasonable speed and efficiency.

The use of wavelet decomposition also served as a method of denoising the image. Though the primary purpose of the wavelet transform was not to denoise but to compress, it was noted that it also served to denoise the data by removing the really small details from the image data set. The random white spots were added to the image to represent those features which could sometimes be misleading during prediction of the white spot. When most of the spots were removed the task of prediction was made easier as the changes were few.

Once the data was compressed the coefficients were then arranged to make them suitable for input to the neural network. The coefficients are separated into approximate, horizontal, vertical and diagonal coefficients. The data from the image is 2-D. This has to

be changed to 1-D data as MATLAB neural network accepts only 1-D input data. The approximate coefficients from each image set is arranged to form a singe row vector. Thus the input is a matrix of the approximate coefficients from all the image sets. The same is repeated with the detailed coefficients. Hence there are four input matrices to be fed into the neural network system one corresponding to each of the type of coefficients. The four matrices are then fed into the neural networks for prediction.

Neural Network System

The neural network system (Figure 4.1) is at the center of this prediction model. The system is made of four networks. The type of neural network chosen for prediction was Radial Basis Functions. The data to be predicted can be considered as a time series. Feed forward networks and RBF are popular algorithms for this application. The back propagation and Elman networks were found to be unsuitable for this application as the network training never converged. The training data was made up of 25 image sets chosen from the database. The images were chosen to provide ample diversity required for the purpose of good training. The training image sets were then preprocessed to yield four different matrices of 25 rows each.

Each network was required only to predict one particular type of coefficient. This enabled that the networks be designed according to the type of coefficient being predicted. As the approximate coefficients contain most of the information, the error tolerance was very low for this network. The training error target was set to $10^{?5}$. The maximum number of neurons that were allowed to be added to the network was 210. This was chosen by trial and error method. While a higher number provided faster convergence, the performance of the network during training was not consistent. At a lower number, convergence took a long time and sometimes the network did not converge at all. Since the error target was high for this network, the time to convergence as also high.

The parameters set for the networks predicting the detailed coefficients were uniform. The target error during training was set to $10^{?3}$. A higher error can be tolerated as the detailed coefficients do not hold the same amount of information as the previous case.

The maximum number of neurons that were allowed to be added to the network was 175. While the networks predicting the horizontal and diagonal coefficients almost always performed identically the network predicting the vertical coefficients performed less precisely. While the other two networks almost always converged within 150 neurons, the vertical coefficients always took more than 160 neurons before they converged to the target error. The time taken for convergence was also greater in the case of vertical coefficients network. The target error for the detailed coefficients was also raised to its final value because at the original value of $10^{?5}$ the network for the vertical coefficient would not always converge within 175 neurons. A trade off was reached between the size of the network and the error tolerance. Since all the detailed coefficients are of equal priority, all the networks were kept identical.

The test images were the remaining 15 images in the database. They were then preprocessed the same way as the training data. The data was split into the various coefficients and input to the corresponding networks. The predicted data is then post processed and the required result is obtained.

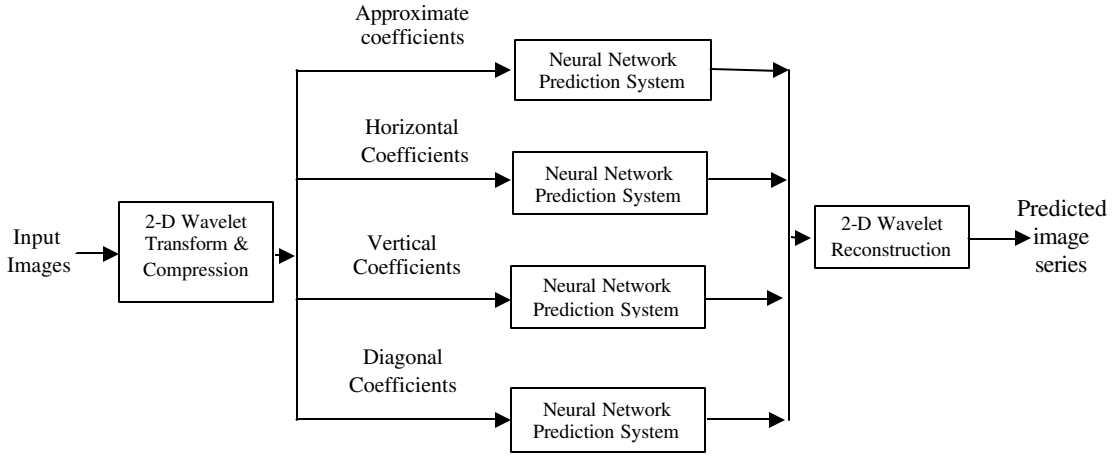


Figure 4.1: Block Diagram of Image Prediction system

Data Post Processing

The outputs obtained from the neural networks are in the form of wavelet coefficients. These are to be reconstructed before the required output is obtained. Before the reconstruction is done the disbanded coefficients of the various images have to be put together. The map of the coefficients is retained from the extraction stage. Using the map the approximate, horizontal, vertical and diagonal coefficients of each image are put together. These are then arranged in order of the images in the series. Then the map of the wavelet decomposition is extracted and zeros are substituted for all discarded lower coefficients in order to maintain the size array to be reconstructed. The coefficients are then reconstructed using Db4 to give the set of image files. The output of the system shows the growth of the spot from instant after the last input image. The images are then saved as an audio video interleaved (*.avi) file. Since the images are reconstructed from only the highest level coefficients the reconstruction is not exact. The errors due to compression and prediction are discussed in further chapters.

CHAPTER 5

PROGNOSIS FROM BIOPSY DATA

While a mammogram may show up an abnormal area in the flesh, it still cannot indicate whether the abnormality is benign or malignant. When mammogram results are ambiguous, they are followed up by other tests such as biopsy. *Fine Needle Aspirate* (FNA) is a needle biopsy method. In this method a few cells are extracted from the area of interest and these are then used to determine the nature of the area of the interest. This enables accurate determination of whether the tumor is malignant or not. However even if the tumor is removed, it is possible that the tumor relapses and if we can determine if the tumor is recurrent it will help in treatment of the patient. In this part of the thesis, the data from the FNA results is used to determine if the given tumor is recurrent or not. If the tumor is recurrent, it then predicts the time to recurrence. Neural networks are used for prediction while Principal Component Analysis (PCA) is used for preprocessing of data.

Database

The data for the thesis was obtained from the Machine Learning Repository of the University of California, Irvine [1]. It was donated by Nick Street in 1995. The database is a follow up on patients diagnosed with breast cancer. The data has been calculated from the digitized image of the FNA of each patient's breast mass. The database has records of 198 patients seen by Dr.Wolberg since 1984. Each record has 34 features. The first three features correspond to ID number of patient, outcome (recurring or non-recurring) and time to recurrence. The other features are results from the FNA analysis of the cell nuclei such as radius, texture, perimeter, area, symmetry etc. the recurrence time was calculated by a method called *Recurrence Surface Approximation* (RSA).The data for each record include the mean, standard error and the worst calculation for each feature computed. The numeric fields are denoted with four significant digits. All the features are real valued except the outcome which is given by *N* or *R* for recurring or

non-recurring. The database has 151 recurring cases and 47 non-recurring cases. The last field is missing in four cases. This is filled during the preprocessing stage.

Principal Component Analysis

Principal Component Analysis (PCA) is a statistical method of finding a pattern in a large set of data by the *Karhunen-Loeve* method. When a large set of data in which the elements are highly correlated is at hand, PCA can be used to reduce the dimensionality of data. PCA is very useful in cases of data with large dimensions and when visual representation is not possible. In a way, PCA also serves as a compression technique.

The algorithm to determine the principal components of a dataset is as follows. The data is at first normalized. It can be normalized by subtracting each element from the mean of the dataset. Any method of normalizing the data can be used. The covariance matrix of the normalized data is calculated. For an n – dimensional data set it is an (n x n) matrix giving the variance of each dimension with respect to the others. The formula for covariance is given by

$$\mathrm{cov}(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{(n-1)} \qquad (5\text{-}1)$$

where X and Y are data points. $\overline{X}$ and $\overline{Y}$ are the mean values of X and Y respectively.

The next step is to find the *Eigen values* and *Eigen vectors*. It is necessary to make sure the calculated Eigen vectors are of unit length. Now the Eigen vectors are arranged in descending order of their Eigen values. This gives the components of the data set in order of their significance, i.e., the first vector is the principal component. It contributes to maximum variance in the data. The matrix containing the principal components is called the *feature vector.* The feature vector is transposed and is then multiplied with the transpose of the normalized data to get the final output. Here it is assumed that the data is linear and that the principal components are orthogonal. The limitation of this method is that it is non-parametric, i.e., the parameters cannot be adjusted according to any apriori

knowledge of the database. The results are unique and independent of the user. If the original data is to be reconstructed from its principal components then the final output is multiplied with the inverse of the feature vector.

Once the Eigen vectors are arranged in the ascending order of their Eigen values the components that contribute to less than $p\%$ of variation in data can be discarded. The value of p is chosen by the user based on the data considered and the compression ratio required. This serves as a data compression technique.

Data Preprocessing

The database does not have an equal distribution of both categories (recurring and non-recurring tumor). The training data set for the neural network was made up of 87 records which consisted of 50 recurring cases and 37 non-recurring cases. Wavelet preprocessing could not be used for this phase as the data was not spatially or temporally related. The data corresponding to the state of the tumor (whether recurring or non recurring) is denoted using a character. This is converted to numeric data by replacing the character by its ASCII value.

PCA is used to reduce the dimensionality of the FNA results and to compress the data. The MATLAB preprocessing function was used for PCA analysis. The data was normalized to have a value between -1 and +1. PCA analysis was then done on this data set. MATLAB uses the method of *singular value decomposition (SVD)* to calculate the principal components.

Mathematical Definition of SVD

Any (m x n) matrix A (where m is greater than n) can be written as

$$A_{mxn} ? U_{mxm} S_{mxn} V^T_{nxn} \qquad (5\text{-}2)$$

where U and V are orthogonal matrices. If the rank of A is r, then the first r columns of V are called *right singular vectors* and the first r columns of U are called *left singular vectors*. S is a diagonal matrix with singular values.

Data Compression using PCA

Once the principal components were computed, all those contributing to less that 2% variation in the data were discarded. Only 10 out of the 34 features were retained after compression. This shows that the data was highly correlated and redundant. The transformation matrix corresponding to the data was stored so that it can be used every time to ensure uniformity in preprocessing.

Neural Network System

The neural network is the heart of the prognosis system. In this part of the thesis three different neural network algorithms were used and their performances were compared. Since this was a prediction problem, supervised learning was chosen. The three networks chosen were Feed forward networks (FF), Radial Basis Functions (RBF) and Recurrent networks (Elman network). As in the previous case the neural network was split into one predicting the condition of recurrence and another to predict the number of months to recurrence. If the case is non-recurring then it predicts the disease free time. The parameters for each network can thus be adjusted individually in order to decrease the error during prediction. The system is configured as shown in the figure 5.1.

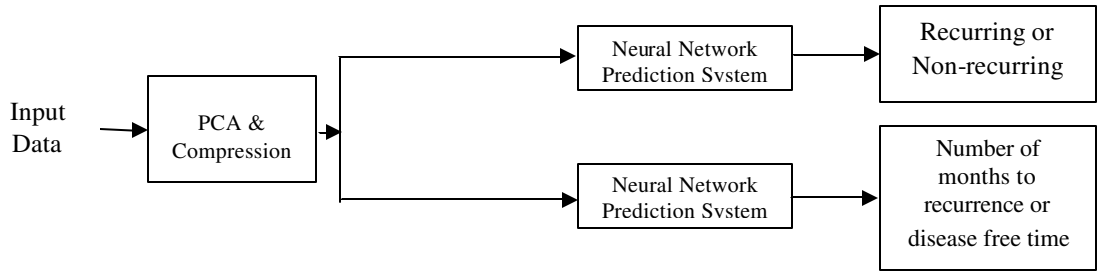


Figure 5.1: Recurrence Prediction System

Prediction of Recurrence or Non-Recurrence

The input to the neural network is the vector of the principal components of the FNA data (as shown in the figure). The neural network is trained to predict whether the particular case of tumor is recurring or not. The training set had 87 records. The output was one of the two values of 78 or 82 which corresponds the ASCII value of *N* and *R* respectively. The compressed data was passed into the neural network.

Feed forward Network

The Levenberg-Maraquadt learning algorithm was chosen to train the network because of its speed. It is proven to be the fastest training algorithm for medium sized networks with up to a few hundreds of weights. The only drawback of this algorithm was that it is very memory intensive. In this algorithm the learning rate was set to be 0.07 and the target error was set to be $10^{?5}$. The prediction of recurrence is very crucial and hence the error tolerance for this network is kept very low. The maximum number of training epochs allowed was 10^3. The test set consisted of 10 records. The Levenberg-Maraquadt algorithm took only 10 epochs to converge to the target error.

Radial Basis Function

The Radial basis functions (RBF) network was also provided the same target error as the previous network. The maximum number of neurons that were allowed to be added to the network was 85. The network always converged with just 35 neurons. The RBF was faster and more computationally efficient. While the feed forward network can sometimes to get stuck in a local minima this did not happen with the RBF.

Elman Networks

The recurrent network chosen was Elman networks. Elman networks are just two layer back propagation networks with feedback from one layer to the previous one. Their similarity to feed forward networks makes them more suitable for the current application. The design parameters of the Elman network were set to be identical to the feed forward network in order to maintain uniformity for the purpose of comparison of performance.

The learning rate was set at 0.07 and the target error was set at $10^{?5}$. The maximum number of training epochs allowed was 10^{3}. The Elman network required more number of epochs to converge than the feed forward networks. They required an average of 130 epochs to train on the data. This can be attributed to the speed of the Levenberg-Maraquadt algorithm. The Elman network was not as memory intensive as the feed forward network.

Prediction of time to recurrence

The neural network is required to predict the time for recurrence or disease free time for corresponding to the decision in the previous case. The problem of prediction of time is quite different from the other one though the inputs are the same. The time unit is months. The error tolerance is not as low as the previous case. The size of the training and the test data sets was the same as the previous case. The input data was the same as the previous case. The target during training stage was a vector of the time (in months) for recurrence of tumor (if the decision was recurring in the previous case) or the disease free time (if the decision was non-recurring).

Feed Forward Network

The back propagation method with the Levenberg-Maraquadt algorithm was used in this method too. The learning rate was set to 0.56 and the error tolerance was set to 10^{-3}. The maximum number epochs was set at $10^{4.}$ The network took an average of 500 epochs to converge.

Radial Basis Functions

The RBF was given the same error target as the feed forward networks. The maximum number of neurons that were allowed to add was 35. The RBF used all the allowable neurons in order to converge. The network was not able to train completely if the neurons were added two at a time. For complete training any other arrangement for addition of neurons failed.

Recurrent Networks

The Elman network was used with the same design parameters as the feed forward networks. The learning rate was set to 0.56 and the error tolerance was set to 10^{-3}. The maximum number epochs was set at $10^{4.}$ The Elman network took 400 epochs to converge to the target error. In this case the performance during the training stage was more consistent than the feed forward network.

CHAPTER 6

RESULTS AND DISCUSSION

Prognosis of Image Data

The test data set contained 15 records. The input was images 1 to 15 from a series of 50 images. The system was required to predict the rest of the images in the series. In order to reduce the amount of data to be handled by the neural network, the input images were decomposed to wavelet coefficients. Only the highest level coefficients were retained and all the others were set to zero. The remaining coefficients were then separated into approximate, horizontal, vertical and diagonal coefficients. They were them made as inputs to four different networks. The task of each network was to predict one type of coefficients for all the images. These predicted coefficients were then reconstructed in order to obtain the required images. The task of the neural networks was finally reduced to one of predicting the coefficients.

The RBF network was used in all the neural networks. The performance of the network was evaluated using the following metrics:

Average Error

Average error was calculated during the output stage of the neural network. Only the prediction error was considered. The error due to pre processing and post processing was not considered as it did not contribute to any significant error.

$$\text{Average\_error} = \frac{\text{expected\_output - observed\_output}}{\text{numberofrecords}} \qquad (6\text{-}1)$$

Correlation

This parameter calculated the correlation between the expected and the predicted output. Correlation was calculated using the in built MATLAB function. Correlation for two variables X and Y (with mean of $\overline{X}$ and $\overline{Y}$ respectively) is given by

$$\mathrm{cov}(X,Y) \;?\; \frac{?_{i?1}^{n}(X_i ? \overline{X})(Y_i ? \overline{Y})}{(n ? 1)} \qquad (6\text{-}2)$$

Power Spectral density

Power spectral density (PSD) of a signal gives how the power (or variance) of a signal is distributed over a frequency spectrum. PSD plots can be used to study the correlation between the expected and observed outputs. The in built MATLAB functions were used for this purpose.

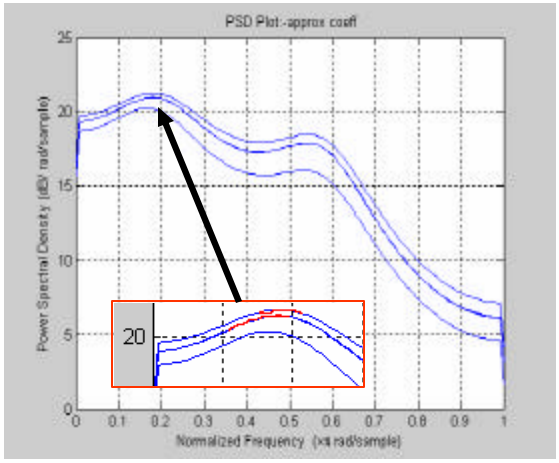
Analysis of Results

The error due to wavelet compression was not considered during the error calculation as the loss due to compression was expected and the system was designed so that loss due to compression will not cause error predicting the location or the size of the area of interest (white spot). The performance of the prediction system is essentially the performance of the four neural networks.

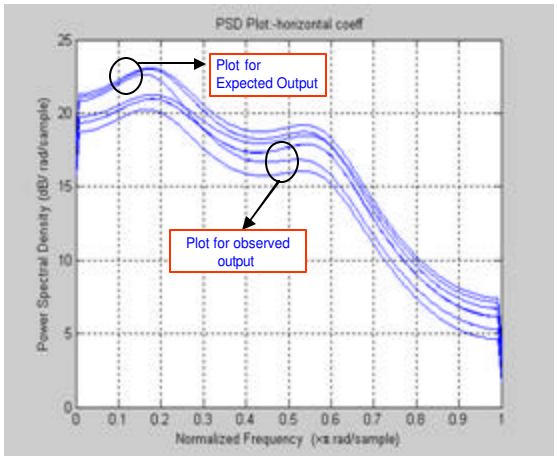
The RBF network was used in all four cases as it was the only architecture where the training converged all the time. The average error of the four networks during the test period is listed in the Table-1 below. It is noted that the performance of the network predicting the vertical coefficients is very poor. Therefore during the reconstruction stage the vertical coefficients were disregarded. The reconstruction without the vertical coefficients did not show any significant loss or error. Hence at all trials, the vertical coefficients were not used for reconstruction.

Table 1: Average Error in prediction of Coefficients

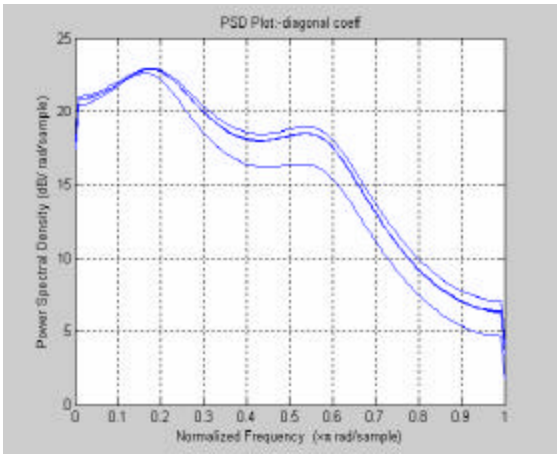
Coefficients	Average Error for all images
Approximate	7.9211×10^{-7}
Horizontal	5.0718×10^{-7}
Vertical	5.79259
Diagonal	2.440×10^{-6}



(a)



(b)



(c)

Figure 6.1: PSD Plots of Expected and Observed Outputs

The PSD plots of the approximate, horizontal and diagonal coefficients are shown in the figure 6.1 above. The performance of the network is determined by how closely the PSD plot of the observed output follows that of the expected output. The consistent performance of the system in spite of a very high compression ratio can be attributed to the very high accuracy in the prediction of the approximate coefficients. It can be seen from the plot that for three different test data sets, the expected and the predicted outputs overlap each other showing only three lines. The PSD plot of the horizontal curve shows that while the general shape of the expected output is followed by the predicted curve, there is a certain loss in power. The diagonal coefficients show the same amount of accuracy as the network for approximate coefficients. The network predicting the vertical coefficients did not keep up this accuracy trend. Even changes in the network parameters such as number of neurons to be added and reduction in target error rate did not improve the performance of this network significantly. The PSD plot (Figure 6.2) shows that not only the predicted output lost the required shape but there is also a huge power loss in the predicted output. Since this corresponds only to detailed coefficients, the performance of the reconstruction system was not affected when the output of this network was disregarded.

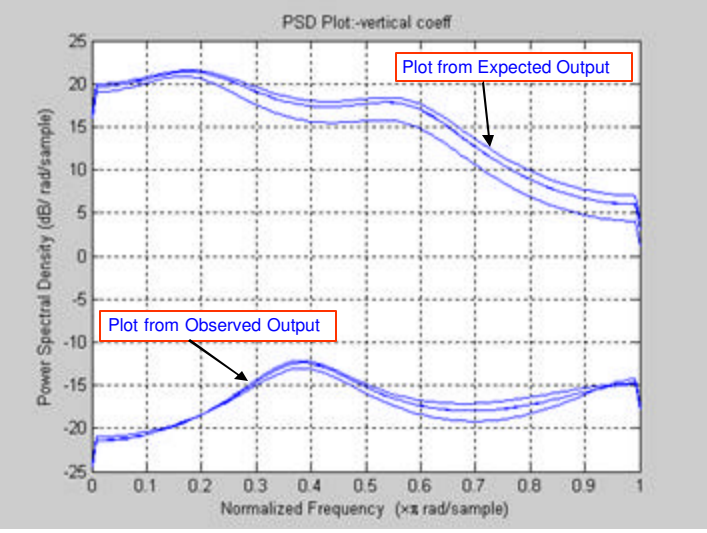


Figure 6.2: PSD Plot of Vertical Coefficients

It is thus seen that if the performance of the network predicting the approximate coefficients is highly accurate a certain amount of error from the other networks can be tolerated. The inclusion of the vertical coefficients degraded the performance more than their omission. A highly accurate prediction of approximate coefficients also facilitates a high compression thus saving valuable computation resources and training time.

Prognosis of Biopsy Data

The prediction problem was considered to be two-fold. One network was created to predict if the tumor was recurrent. Another network was created to forecast the number of months. This was done in order to observe if any particular worked better for one of the problems and not the other so that an optimum combination can be used to achieve better accuracy. The implementation was done using the Neural Networks Toolbox in MATLAB. Two different training sets were used, one where there was an equal number of recurrent and non-recurrent cases. In the other the distribution was not equal, there were more non-recurrent cases.

Feed Forward Networks

- The feed forward network was trained using the Levenberg-Maraquadt algorithm. In both cases the training took only 7-9 epochs to meet the error goal.

- The number of hidden nodes was set to 20. This was chosen by process of trial and error. The normal rule of square root of number of input nodes did not seem to work for this application

- A sigmoid activation function was used.

Elman Network

- The activation functions in an Elman Network were a sigmoid function in the first layer and a linear function in the next layer.

- Here the number of hidden nodes was chosen to be 15, also by trial and error.

- If the same input is presented to the sigmoid more than once the result is not always the same. This occurs due to the feedback property of the recurrent network.

Radial Basis Function Network

- The RBF function was set to the default transfer function in MATLAB.
- The network was allowed to adaptively add neurons till the error goal was met.

The three metrics used to compare the performances of the three networks were average error, correlation and power spectral density. (as explained in the previous section)

Analysis of the Results

To predict if the Tumor is Recurring

The performance of the networks with respect to the average error is shown in Table2.

Table 2: Average Error and Correlation Coefficients of Predicted Coefficients

Method	Average Error	Correlation
RBF	0.1228	0.8004
Elman	0.3097	0.6260
FF	0.5330	0.4738

It is seen that the average error is the lowest for the RBF network. The correlation between the required and the predicted output is also the highest for the network. The feed forward network which is a common predictor performed poorly for this application. This can be attributed to the fact that combination of linear and non-linear layers in the RBF network was able to catch the pattern in the input data better than the traditional *backprop learning*. The performance of the Elman Network was not very far behind the RBF network. The feedback actually worked against the Elman network as each case does not depend on the result of the previous record. If during the implementation of this system for a similar application, the RBF network does not seem feasible, the Elman can be a close second.

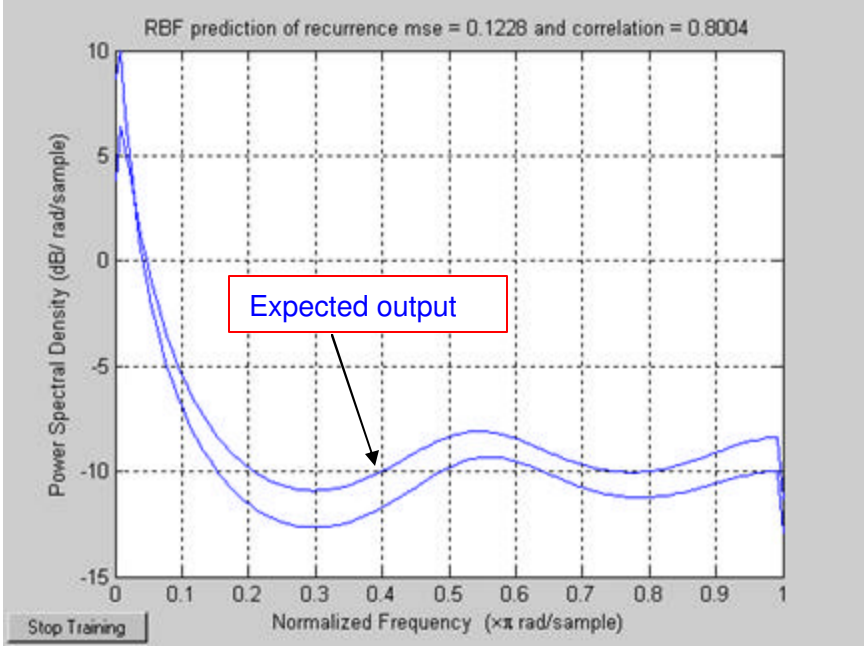


Figure 6.3: PSD Plot for RBF Network

For the RBF network PSD plot (Figure 6.3) of the predicted output follows the expected output except for a small loss of power throughout. In the first few samples the expected and predicted outputs overlap.

The next plot is that of the Elman network (Figure 6.4). It is seen that during the initial stages the Elman network follows the required curve much closer than that of the RBF

network. At one point the Predicted curve overshoots the expected curve. This is due to the prediction of a false negative. The penalty for predicting a false negative was set to be higher than that of a false positive. This is because in real world, the consequence of missing a recurrent tumor are graver than those for predicting one when there is none. The curve then falls back at a point where it predicts a falls negative. It starts following the other curve from this point on though with a loss of power.

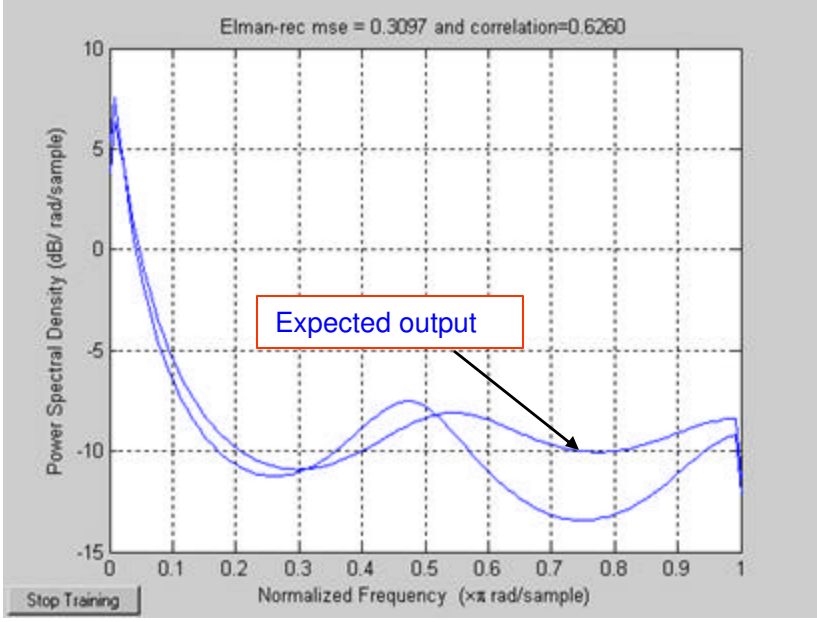


Figure 6.4: PSD plot for Elman Network

The PSD plot of the FF network (Figure 6.5) does not follow the expected output curve at any point. This is also reflected in the low correlation coefficient. The prediction of two false positives and a false negative in a row does not let the system error recover. It also had the highest average error rate.

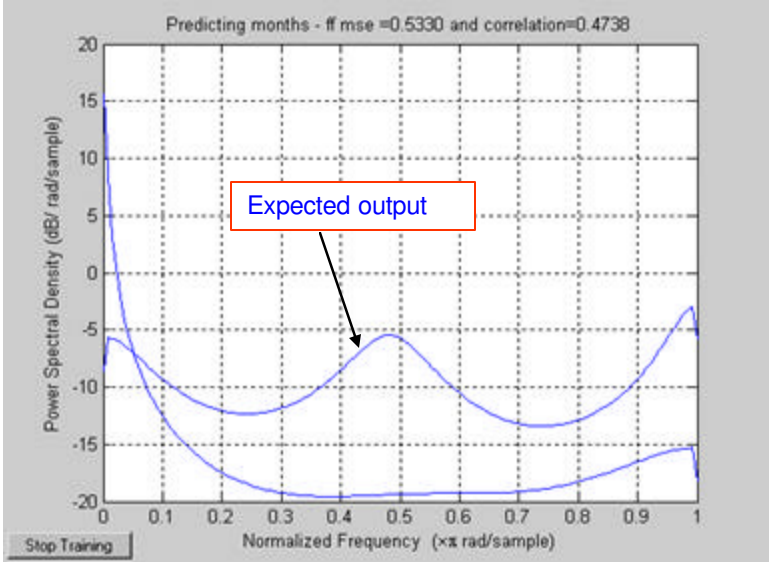


Figure 6.5: PSD Plot for Feed Forward Network

It is seen that the RBF network has the best performance for this application and the error rate was low and where all predictions were correct.

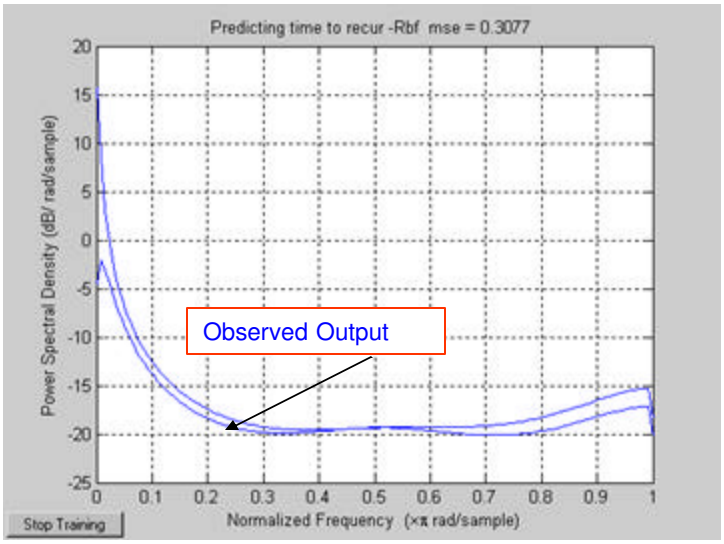
Predicting the number of months to recurrence

The performance of the networks in this case was not very much unlike the previous case.

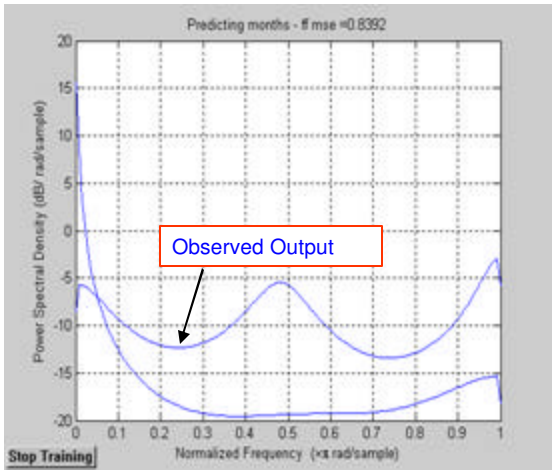
Table 3: Average Error in the Prediction of Months

Method	Average Error
RBF	0.3077
FF	0.5683
Elman	0.4785

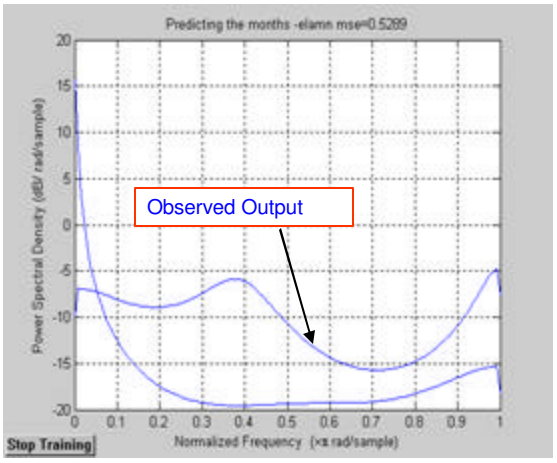
The RBF network's performance improved from the beginning and perfectly matched the required output till the fifteenth case. The case was a wrong prediction. Hence the curve is set off by a small amount but it does not affect the performance much as the network continues to predict accurately after that.



(a)



(b)



(c)

Figure 6.6: PSD Plots for all Networks

The PSD plots of Elman and FF networks are shown in Figure 6.6. The network outputs are nowhere close to the required outputs. The best performance of the Elman network was with 20 hidden nodes while for the FF it took 25 hidden nodes to achieve it best performance. In spite of varying different parameters the performance of the networks did not improve significantly. The Elman network changed the performance at different runs of the program because of the inherent nature of the feedback. The two networks are unsuitable for this application.

CHAPTER 7

CONCLUSION AND FUTURE WORK

This thesis introduces the concept of *Computer Aided Prognosis*. This system can be applied to detect tumor-like growth much before they are visible to the human eye. This improves the chances of early diagnosis. Patients who have been predicted to have recurring tumors can be monitored closely for better treatment.

The system was successfully implemented in MATLAB. In the first unit i.e. the prognosis of image data, wavelets were used to achieve a very high compression ratio (1:100). It is required that level of compression be decided earlier so that maximum compression is achieved without loss of information. Compression ensures that the computation of the neural network is faster and less time consuming. It was shown that dedicating an individual network to each type of coefficient ensured that the networks be better customized according to the priority of the coefficient. In the first case i.e. prognosis from image data it was shown that a very high level of accuracy in the prediction of approximate coefficient is critical. This ensures a very reliable prediction from the system on the whole in spite of prediction errors form other networks and reconstruction errors. RBF network is best suited for this purpose because the combination of linear and non-linear activation functions allows the network to train with very low target error.

In the second unit (prognosis from BIOPSY data), three networks were compared for prediction capability. When the data is not related in space or in time PCA is a good method of compression. As in the previous case the trade off between compression and loss of data has to optimized. From the performance of the three networks it was observed that the RBF showed the best and the most consistent performance in both the cases. Though the Elman network was a close second in terms of performance, the Elman network was not consistent because the presence of the feedback worked against the

network's performance when successive inputs were not related to each other. The FF network, which is universally popular as a predictor, showed the worst performance among the three. This could be attributed to the fact that simple FF learning system was not sufficient for the application considered as it would often be trapped in a local minima and was not able to converge to the global minima.

This work is just an exploratory tool for the application of neural networks as a prognosis tool. The work can be expanded to suit many applications. The number of images in the data base could be increased. This would provide a larger training set for the network to learn and hence could improve its performance. The system can be made to center on the area of interest once it has been spotted by the system thus reducing the data to be processed further. Original mammogram images can be used. This requires more sophisticated network and training. It also requires sophisticated feature extraction techniques in order to detect the required patterns and ignore the rest. The system can be extended for use in other applications such as failure prediction in materials etc. It can also be extended to three-dimensional images

REFERENCES

1. University of California Machine Learning Repository

Books

2. . Daubechies, *Ten Lectures on Wavelets.* Philadelphia, PA: SIAM, 1992.

3. Burrus, C. S., Gopinath, R. A., and Guo, H.: Introduction to wavelets and wavelet transforms, Prentice-Hall International Inc., Texas, 1998.

4. Haykin, S. (1999). *Neural Networks. A Comprehensive Foundation*, Second Edition, Prentice-Hall, Inc., New Jersey.

Publications

5. Tsui, F.-C., Li, C. C., and Sun, M. et al.: A comparative study of two biorthogonal wavelet transforms in time series prediction, IEEE, 1997.

6. C.J.Vyborny and M.L.Giger, "Computer Vision and Artificial Intelligence in Mammography," *American J, Radiology,* Vol. 162, 1994, pp. 699-708.

7. A.G.Gale, et al., "Computer Aids to Mammographic Diagnosis," *British J. Radiology,* Vol. 60, 1987, pp. 887-891.

8. A.P. Dhawan, Y. Chite, C. Bonasso and K. Wheeler "Radial-Basis-Function Based Classification of
9. Mammographic Micro calcifications Using Texture Features", *IEEE Engineering in Medicine and Biology & CMBE*C, *pp.535-536,* 1995.

10. Jain, A.K. and Mao, J. (1996). *Artificial Neural Networks: A Tutorial*, IEEE Computer, vol.29, N: 3, pp.31-44.
11. R. P. Lippmann. An introduction to computing with neural nets. IEEE ASSP Magazine, (2):4{22, Apr. 1987.

12. Kansal A.R., Torquato S., Harsh IV G.R., Chiocca E.A., and Deisboeck T.S.: Simulated brain tumor growth dynamics using a three-dimensional cellular automaton. J. Theor. Biol. 203: 367-382, 200

URL

13. www.dacs.dtic.mil/techs/**neural/neural**_ToC.html
14. www.nd.com/neurosolutions/products/ns/whatisNN.html
15. www.dacs.dtic.mil/techs/**neural/neural**.title.html
16. www.fdm.uni-freiburg.de/groups/medstat/annoncol

17. engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html
18. www.**wavelet**s.com/ -
19. www.owlnet.rice.edu/~elec301/ Projects00/**wavelet**_image_comp/
20. http://www.generation5.org/content/2004/MedicalDiagnosis.asp
21. http://documents.wolfram.com/applications/wavelet/SummaryofFunctions/3.1.2.html

BIOGRAPHICAL SKETCH

Vidya Rajagopalan was born in the year 1982. She finished her high school education in Chennai, India. She received her undergraduate degree from the University of Madras in the year 2003. She majored in Electronics and Communication Engineering.

She was admitted to the Master's program in the Department of Electrical and Computer Engineering in Fall 2003. She specializes in the field of Image processing specifically Medical Images.