ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
Faculty of Informatics
Department of Information Science

Forecasting Tree Volume Growth using Artificial Neural Networks: the case of Cupressus lustanica Species

BY

FIKADU GEMECHU

Name and Signature of Member of the Examining Board

prof. B.R.K. Rao, Chairman, Examining Board	
Dr. kumudha Raimond, Advisor	
Dr. Lars Asker, External Examiner	

Chairman, Faculty	Signature	Date 07/03/06
Chairman, Graduate Council	Signature	Date

Acknowledgment

First and for most, I would like to thank God for giving me the strength to challenge all the problems I faced during my study periods. Had it not been for his support, it would have been impossible to accomplish my studies successfully.

I would like to express my sincere gratitude and appreciation to my advisor Dr. Kumudha Raimond, for her guidance, support and encouragement during the research undertakings. I also want to thank members of Wondo Genet College of Forestry in providing me with valuable comments and material support during my research work.

I am very grateful to management and staff of the Forestry Research Center for their support in accessing the data and in providing relevant information to my research work.

My special thanks go to my beloved family for their support and encouragement through out my study.

Moreover, my special gratitude goes to my fiancée for her love, support and encouragement through out my career. Lastly, I thank my friends for their unreserved help.

Abstract

Neural networks are computational models with the capacity to learn or generalize complex relationships that exist among data. Although there are different kinds of networks, the multi-layered feed-forward neural network is the most widely used network that is capable of representing non-linear functional mappings between inputs and outputs. The training of this network is accomplished by the method of error back-propagation.

In this paper, the feed-forward neural network with back propagation learning algorithm is presented for forecasting tree volume growth of *Cupressus lustanica* species. The data set for training the neural network was obtained from the Forestry Research Center (FRC). As an input to the neural network, the historical tree volumes are used to train the network. After training the network, the results of forecasting is evaluated using test data set. The result indicates that the model yields good prediction with independent test data set, providing about 86.7% correct forecasts within $\pm 2cm^3$ of the observed values. This suggests that the neural network is a good candidate for forecasting future value of tree volume given properly and accurately measured historical data.

Table of contents

List of Tables

List of figures

Chapter One

Introduction

1.1. Background

Decades ago, forests covered the majority of Ethiopian land. However, these forests keep on disappearing at an alarming rate for fuel wood, cultivation and ruthless exploitation without reforestation.

According to the report made by Pohjonen and Pukkala (1990) cited in Tesfaye (1996) in the early 1990s, 40% of the total land area was covered with forests. Today the percentage of the land covered by forest is less than 3% of the total land area. This figure reveals the rapid rate of deforestation. Although agricultural and livestock expansion and increased demand for commercial and non-commercial forest products have caused the deforestation of the natural forest, the primary cause of deforestation is basically for fuel wood.

The major species of the remaining natural forests are *Podocarpus gracilior, Celtis chloplora, Antiaris, Cordia, Albizia, Juniperus*, with many other species being present in smaller amount. Because of the gradual decline in size of natural forests, the plantations were established in selected priority areas to meet the ever-increasing demand for fuel and industrial wood. The Munessa Shashemene State Forest Project (MSSFP), where the data for conducting this research is collected by the Forestry Research Center, is one of the priority areas. The planted species in this area are predominantly *Cypress, Pines, Eucalyptus* and some indigenous species (Mengistu, 1990 cited in Tesfaye, 1996).

1.1.1. *Cupressus lustanica*

Cupressus lusitanica, also called Mexican cypress, belongs to the family of Cupressaceae and genus *cupressus*. Dayson (1968) cited in Tesfaye (1996) reported that it occurs naturally between latitude 15-45^0 North in Central America, being widespread in central and southern Mexico, Guatemala, Hondurans and El Salvador, where it grows to a height of 30 meters at 1800 – 2400 meters elevation. *Cupressus lusitanica* is the major exotic species planted in the Munessa Shashemene for saw – timer (about 62% of the plantation area). Although its first introduction to Ethiopia is not well documented, it is the most widely planted exotic species next to eucalyptus.

Cupressus lusitanica seedlings can be raised either from seeds or roots and shoot cuttings (Dale, 1953 cited in Tesfaye, 1996). Artificial regeneration through seedling planting is the common practice in Ethiopia.

1.1.2. Forest growth modeling

A stand growth model is an abstraction of the natural dynamics of a forest stand, and may encompass growth, mortality and other changes in stand composition and structure. With suitable inventory and other resource data, growth models provide a reliable way to examine silvicultural and harvesting options to determine the sustainable timber yield, and examine the impacts of forest management and harvesting on other values of the forest.

1.1.2.1 Conventional growth modeling approaches

Forest growth models are classified into a whole stand model, a size class model and a single- tree model, depending on the details provided and utilized by the model.

a) Whole stand models

They are often simple and robust, but may involve complexities not possible in other approaches. Population parameters such as stocking (number of trees per unit area), stand basal area and standing volume are used to predict the growth of the forest. No details of the individual trees in the stand are determined (Vanclay, 1994).

b) Size class models

They provide some information regarding the structure of the stand. Several techniques are available to model stand structure, but one of the most widely used is the method of stand table projection which essentially produces a histogram of stem diameters. This approach is a compromise between whole stand model and single-tree model. When the class size is remarkably large and only one class exists, then the method is a whole stand approach. When the class width is remarkably small and each tree is considered a single class, then the method is the single-tree approach (Vanclay, 1994).

c) Single-tree model

It is the most detailed approach and uses the individual tree as the basic unit of modeling. The minimum input required is a list specifying the size of every tree in the stand. Some models also require tree height and crown class (Vanclay, 1994).

According to Vanclay (1994) irrespective of its detail, a model may be deterministic or stochastic. A deterministic growth model gives an estimate of the expected growth of a forest stand in the same way the mean indicates the expected trend for a population. Given the same initial conditions, a deterministic model will always predict the same result. However, because of natural variation in the

environment, real forest stands may not grow exactly the same amount each year, but may grow more or less than the expected amount. A stochastic model on the other hand attempts to illustrate this natural variation by providing different predictions, each with a specific probability of occurrence. Any one of these estimates may correspond exactly to the growth under some circumstances, but may differ from the expected growth. A single estimate from a stochastic model is of little use, as a whole series of estimates is necessary to provide useful information of the variability of predications.

1.1.4. Neural networks: a short tour

The name Artificial Neural Networks covers a broad range of computational methods. It is relatively a new AI technique that emulates the behavior of biological neural system in digital software or hardware. This network can learn automatically complex relationships among data. This feature makes the technique very useful in modeling process for which mathematical modeling is difficult (Gorni, 1997). Neural networks are applicable in virtually every situation in which a relationship between the predictor variables (independents, input) and predicated variable (dependents, outputs) exists, even when that relationship is very complex and not easy to articulate in the usual terms of "correlations or differences between groups" (StatSoft, 2003).

1.2. Statement of the problem

One of the problems that the forestry sector is facing nowadays is that there is lack of comprehensive computer based models for efficient and effective management of the forest resources. In the country like Ethiopia where the previously existing forest resources are diminishing at a fast rate, the immediate solution to be put into place is a proper way of managing the resources at national

level that in turn maintain soil fertility and increase biodiversity. One way of doing this is by developing a proper growth model that in turn provides information required for forest management.

There are different statistical based approaches of growth model development. Despite their popularity, there exist some drawbacks associated with them. They often do little to further understanding of the processes involved in the system being modeled. While they describe the data satisfactorily, they may give anomalous estimates for values lying outside the range of data on which they are based. In addition limited number of variables will get involved in modeling non-linear growth functions. The regression models currently used are intrinsically linear models (linear in the parameters, although the factors can be nonlinear functions of the original variables), thus they cannot accommodate inherently nonlinear effects in the growth response. The conventional methods are often based on the some assumptions on the distribution of the data such as normality and homogeneous variance. Certain assumptions may be invalid with forest inventory data where classes may display a range of distributions, where relationships between variables may be nonlinear, and where outliers and noise may exist in the data. In addition, the forecast of tree volume is based on volume table, which does not provide good prediction. Hence, the need for an alternative approach of volume growth modeling which, the researcher thinks curb the problems associated with conventional approach. Thus the main reason of conducting this research is to develop volume growth model using artificial neural network with multilayer perceptron, with back propagation, for *Cupressus lustanica* species.

1.3. Objectives of the study

1. 3.1. General Objective

- Exploring the potential applicability of neural network in tree volume prediction with particular emphasis to *Cupressus lustanica* species.

1.3.2. Specific Objectives

In order to achieve the above stated general objective, the study will undertake the following specific objectives:

- To select and extract the target data set for analysis from the Forestry Research Center (FRC).
- To select the appropriate neural network tool to be used to develop the models.
- To collect, clean and prepare the data in a format acceptable by the selected neural network tool.
- To develop models and evaluate their performances.
- Report the findings and forward recommendations.

1.4. Methods

The methodology applied for this research is adopted from the technique used by Stat Soft (2003) and Two Crows (1999). It uses the different data mining steps that need to be carried out to develop a decision support system assuming that the business problem is well understood.

On the way of conducting this research work, constructive comments and experiences were obtained from the forest experts of Wondo Genet College of forestry. The different methods employed for the research are provided below:

1.4.1. Literature Review

A review of literature was made in the course of conducting this research work. Books, journals, manuals, WWW and other pertinent materials on neural network and tree volume growth modeling have been consulted.

1.4.2. Identifying the data source

The data on which the research work was conducted is taken from the Munessa Shashmene site plot, which was established by FRC, Addis Ababa. The data is in an electronic format and contains some measured parameters like height, diameters at breast height (DBH) which was taken annually from permanent sample plots and corresponding to each height-diameter pair, there are tree volumes which can be derived from them and used as input to the neural network. In addition to the tree dimensional variables there are also plot numbers, years of planting, years at which measurements were taken and tree numbers being incorporated in the data. The data for this study is on the *Cupressus lusitanica* species, which is the most widely planted tree species in the project area.

1.4.3. Preparing data for analysis

Before using the raw data for analysis, it had to be converted into a format that was compatible with its end use. Hence, the collected data was cleaned into a form that is suitable for particular neural network software to be used. In spite of being cleaned, the data needed to be transformed to make it suitable for the

algorithm. Among the available neural network software, MATLAB 7.0 is used since it has many algorithms that can be applied for prediction problem.

At this stage pre-processing tasks like handling noisy data, unknown values, missing values, deriving new fields from the existing ones, and summarization of data was performed by taking into account the selected neural network tool.

1.4.4. Training and model building

The next step after pre- processing the collected data was training and model building. Model building was accomplished using the back propagation algorithm on MATLAB. The training was carried out by dividing the whole data set into training, validation and testing sets. In the way of training, many models were built and the one with high performance was selected.

1.5. Scope and Limitation of the study

The scope of this research is limited to particular species namely *Cupressus lustanica* which is among the dominant species that the research center has permanent research plots.

The major limitation while undertaking the research was the shortage of the relevant research papers undertaken in the area, which hinders the researcher from reviewing them in detail. As a second limiting factor to undertake the research, the researcher will not by- pass with out mentioning the total number of data which is less to obtain high accuracy result out of the employed software. In addition, the absences of important variables from the data were also another limiting factor. Lacks of prior experience in ANN techniques as well as the problem domain were also limitations faced by the researcher. As the last limiting

factor while undertaking this research, the researcher mentioned the time constraint faced.

1.6. Organization of the research

This research report is organized into five chapters. The first chapter is an introductory part, which contains background to the research work, statement of the problem addressed, objective of the research, the methodology adopted for the study and the scope and limitation of the study.

Chapter two reviews background materials necessary to understand the basic concepts and results of this research. The fundamental concepts of the ANN are reviewed in detail.

Chapter three gives a brief introduction of tree volume growth forecasting. Methods of measuring tree dimensional parameters and estimating tree volume are also presented. It also presents a short review of literatures.

Chapter four presents the experimentation phase of the study. This chapter discusses the data collection, data preparation, and model building process. Results of training and testing of the neural network models are also discussed in this chapter.

The final chapter provides conclusions and recommendations given based on the findings of the undertaken research.

Chapter Two

Fundamentals of Artificial Neural Networks

An artificial neural network (ANN) is an information-processing paradigm that is inspired by the way biological nervous system, such as the brain, process information. It can also be referred to as an adaptive statistical model based on analogy with the structure of the brain. Its adaptive behavior comes from the way it can learn to estimate the parameters of some population using a small number of examples at a time.

These biologically inspired methods of computing are thought to be a recent development having a resurgence of interest by many researchers and hence a corresponding increases in funding.

The interest in neural networks comes from its potential to find a solution to various problems of application domains. It has been used in large number of applications and has proven to be effective in performing complex functions in a variety of fields. These include pattern recognition, classification, vision control system and predication (Fauset, 1994).

Neural networks take a different approach to problem solving than that of conventional computers, which uses an algorithmic approach. i.e. the computer follows a set of instructions in order to solve a problem.

Various definitions have given by different researchers to artificial neural networks. Among them, the definition given by Haykin (1994), which views neural networks as an adaptive machine, is taken as comprehensive one.

"A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the human brain in two respects:

1. *Knowledge is acquired by the network from its environment through a learning process.*
2. *Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge."*

2.1. Basic structure of a neuron

2.1.1 Human brain

The basic element of the human brain is a specific type of cell that is very large in number and provides us with the abilities to remember, think, and apply previous experiences to our every action. These cells are known as neurons. Each of these neurons can connect with many neurons. The power of the brain comes from the number of these basic components and the multiple connections between them. Basically, a biological neuron receives input from other sources, combines them in some way, performs a generally nonlinear operation on the result, and then outputs the final result

All natural neurons have the same four basic components. These components are known by their biological names-dendrites, soma, axon, and synapses. Dendrites are hair like extensions of the soma, which act like input channels. These input channels receive their input through the synapses of other neurons. The soma then processes these incoming signals over time. The soma then turns the processed value into an output, which is sent out to other neurons through the axon and

synapses (Anderson and McNeil, 1992). Figure 2.1 shows the relationship of these four parts

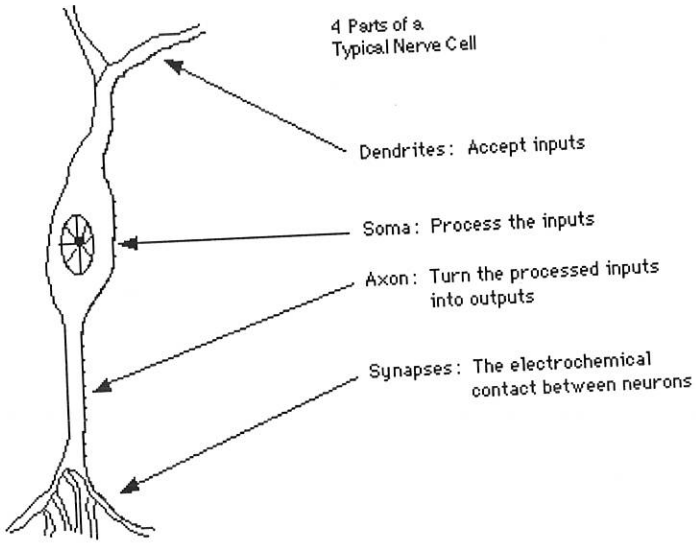


Figure 2.1 A pictorial representation of a biological neuron

2.1.2 Artificial neuron

The biological neuron model is the foundation of an artificial neuron. The basic unit of neural networks, the artificial neuron simulates the four basic functions of the natural neurons. Artificial neurons are much simpler than the biological neurons.

In the ANN literature, the unit analogous to the biological neuron is referred to as a processing element or unit. The general structure of processing element is shown in figure 2.2. The various inputs to the network are represented by the mathematical notation, x_m. Each of these inputs are multiplied by a connection

weight. These weights are represented by $w(km)$. In the simplest case, these products are simply summed and generate an internal activity level u for the process element.

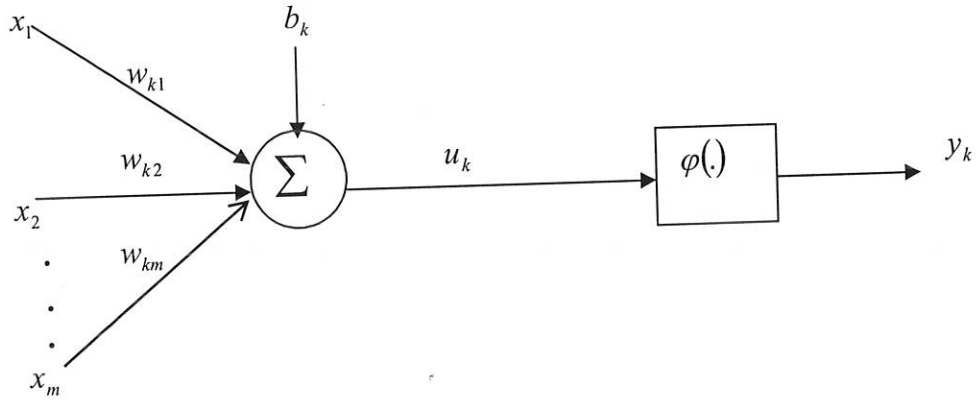


Figure 2.2 A nonlinear model of a processing element

The combined input u_k is then modified by the transfer function. This activation function can be a threshold, which only passes information if the combined activity level reaches a certain level, or it can be a continuous function of the combined output. The output value of the activation function generally passed directly to the output path of the processing element. The neuron model in figure 2.2 also includes an externally applied bias, denoted by b_k. The bias b_k has the effect of increasing or lowering the net input of the activation function, depending on whether it is positive or negative, respectively (Haykin, 1999). The bias b_k is an external parameter of artificial neuron k. In mathematical terms, the neuron k is described by the following pair of equations:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.1}$$

and

$$y_k = \varphi(u_k + b_k) \tag{2.2}$$

Where $x_1, x_2, ..., x_m$ are the input signals; $w_{k1}, w_{k2}, ..., w_{km}$ are the synaptic weights of neuron k; u_k is the linear combiner output due to the input signals; b_k is the bias; $\varphi(.)$ is the activation function; and y_k is the output signal of the neuron.

2.1.2.1 Major Components of an Artificial Neuron

According to Anderson and McNeil (1992), the seven major components that make up an artificial neuron are described as shown below. These components are valid whether the neuron is used for input, output, or is in one of the hidden layers.

1. Weighting factors

A neuron usually receives many simultaneous inputs. Each input has its own relative weight, which gives the input the impact that it needs on the processing element's summation function. These weights perform the same type of function similar to the synaptic strengths of biological neurons.

Weights are adaptive coefficients within the network that determine the intensity of the input signal in the artificial neuron. They are the measure of an input's connection strength. These strengths can be modified in response to various training sets and through its learning rules.

2. Summation function

The first step in processing element's operation is to compute the weighted sum of all of the inputs. Mathematically, the inputs and the corresponding weights are vectors which can be represented as $(x_1, x_2, ..., x_m)$ and $(w_1, w_2, ..., w_m)$. The total input signal is the dot, or inner product of these two vectors. The result is a single number, not a multi-element vector.

The summation function can be more complex than just the simple input and weight sum of products. The input and weighting coefficients can be combined in many different ways before passing onto the transfer function. The specific algorithm for combining neural inputs is determined by the chosen network architecture and paradigm.

3. Transfer function

The result of the summation function, almost always the weighted sum, is transformed to a working output through an algorithmic process known as the transfer function. In the transfer function, the sum total can be compared with some threshold to determine the neural output. If the sum is greater than the threshold value, the processing element generates a signal. If the sum is less than the threshold, no signal (or some inhibitory signal) is generated. Both types of responses are significant.

4. Scaling and limiting

After the processing element's transfer function, the result can pass through additional processes such as scaling and limiting. This scaling simply multiplies a scale factor times the transfer value, and then adds an offset. Limiting is the mechanism, which insures that the scaled result does not exceed an upper or lower

bound. This limiting is in addition to the hard limits that the original transfer function may have performed.

This type of scaling and limiting is mainly used in topologies to test biological neuron models.

5. Output function (Competition)

Each processing element is allowed to output one signal, which it may send to hundreds of other neurons. This is just like the biological neuron, where there are many inputs and only one output. Normally, the output is directly equivalent to the transfer function's result. Some network topologies, however, modify the transfer result to incorporate competition among neighboring processing elements. Neurons are allowed to compete with each other, inhibiting processing elements unless they have great strength. Competition can occur at one or both of two levels. First, competition determines which artificial neuron will be active, or provides an output. Second, competitive inputs help determine which processing element will participate in the learning or adaptation process.

6. Error function and back-propagated value

In most learning networks, the difference between the current output and the desired output is calculated. This raw error is then transformed by the error function to match the particular network architecture. The artificial neuron's error is then typically propagated into the learning function of another processing element. This error term is sometimes called the current error.

The current error is typically propagated backwards to a previous layer. Yet, this back-propagated value can be either the current error, the current error scaled in

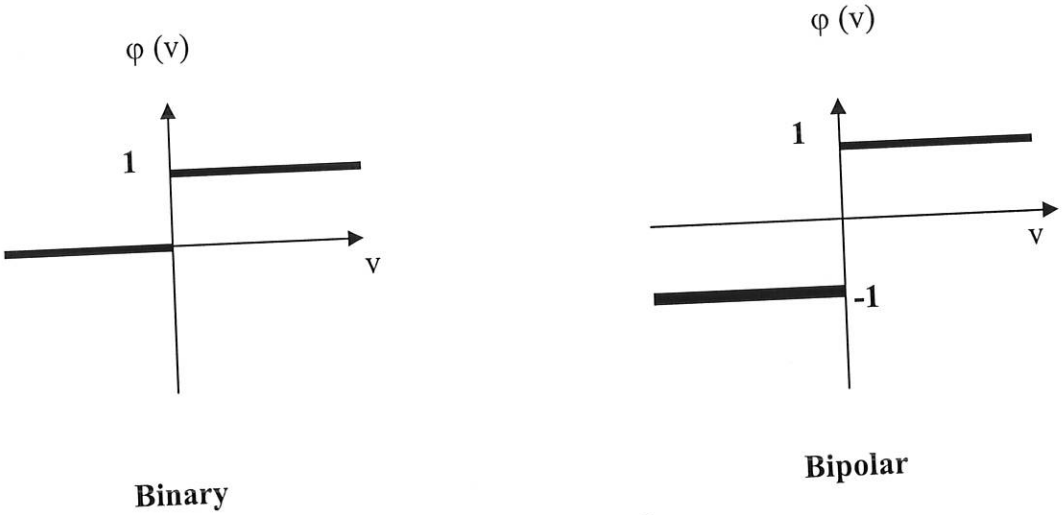
some manner (often by the derivative of the transfer function), or some other desired output depending on the network type. Normally, this back-propagated value, after being scaled by the learning function, is multiplied against each of the incoming connection weights to modify them before the next learning cycle.

7. Learning function

The purpose of the learning function is to modify the variable connection weights on the inputs of each processing element according to some neural based algorithm. This process of changing the weights of the input connections to achieve some desired result could also be called the adaptation function, as well as the learning mode.

2.1.3. Activation functions

The activation function, denoted by φ (v), defines the output of a neuron in terms of the induced local field v. The most commonly used activation functions are described in figure 2.3 (1-4)



Binary **Bipolar**

Figure 2.3 (1) Threshold functions

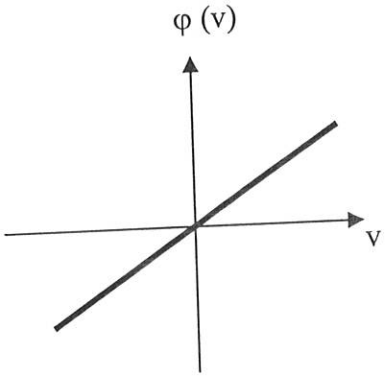


Figure 2.3 (2) linear function

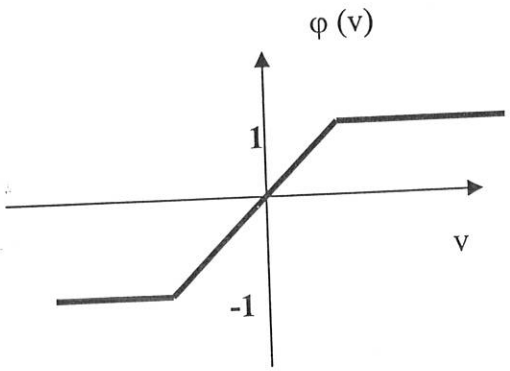


Figure2.3 (3) Piecewise function

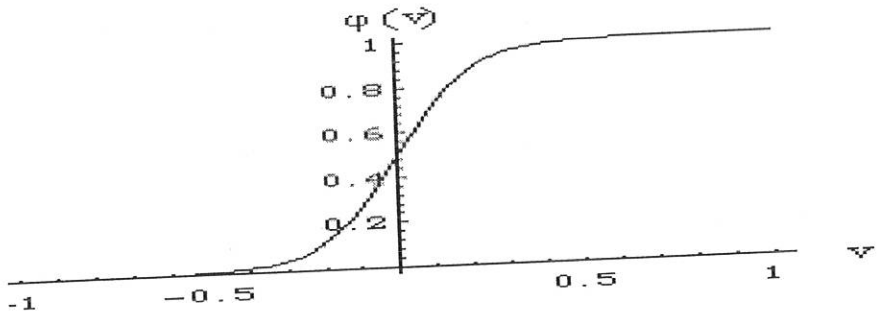


Figure 2.3(4) sigmoid function

1. Threshold function

A threshold function is a type of activation function where the output is set at one of two levels depending on whether the total input is greater than some threshold value. The function is either of binary or bipolar type. The pictorial representation of the threshold function is given in figure 2.3 (1).

$$\varphi(v) = \begin{cases} 0 \text{ if } v < 0 \\ 1 \text{ if } v \geq 0 \end{cases} \quad \text{binary type} \tag{2.3}$$

$$\varphi(v) = \begin{cases} +1 \text{ if } v \geq 0 \\ -1 \text{ if } v < 0 \end{cases} \quad \text{bipolar type} \tag{2.4}$$

2. Linear function

This is a type of function where the output activity is proportional to the total weighted output. Mathematically this relation is expressed as $\varphi(v) = \lambda v$, where λ is the slope of the linear function. When the value of λ becomes one the function reduces to an identity function.

3. Piecewise function

Another type of activation function, ramping function, could mirror the input within a given range and still act as threshold outside that range. It is a linear function that has been clipped to minimum and maximum values, making it non-linear.

$$\varphi(v) = \begin{cases} -1 \text{ if } v < -1 \\ v \text{ if } -1 \leq v \leq 1 \\ 1 \text{ if } v > 1 \end{cases} \tag{2.5}$$

4. Sigmoid (S-shaped) function

For sigmoid units, the output varies continuously but not linearly as the input changes. This function approaches a minimum and maximum value at the asymptotes. The exciting feature of this function is that both the function and its derivatives are continuous. This feature makes it as an activation function of choice for neural networks trained with back-propagation. The general expression of sigmoid function is given as follows:

$$\varphi(v) = \frac{1}{1+\exp(-av)} \tag{2.6}$$

$$\text{Where, } 0 \leq \varphi(v) \leq 1$$

Where, a is the slope of the parameter of sigmoid function. The output of this function is within the range of 0 and 1. Mathematically, this type of function is called the logistic function.

Other activation functions are dedicated to specific network architectures. The choice of the activation function mainly depends on the distribution of the target value, which is generated by the output units. For continuous valued targets with a bound range, the sigmoid functions are preferable, provided that either the output or the targets to be scaled to the range of the output activation function (Sarls, 1977).

2.2. Benefits of neural networks

It is apparent that a neural network derives its computing power through, first, its massively parallel distributed structure and, second, its ability to learn and generalize. These two information-processing capabilities make it possible for

neural networks to solve complex (large-scale) problems that are currently intractable (Haykin, 1999).

The use of neural networks offers the following useful properties and capabilities:

1. **Nonlinearity.** An artificial neuron can be linear or nonlinear. A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear. Moreover, the nonlinearity is of special kind in the sense that it is distributed throughout the network. Nonlinearity is a highly important property, particularly if the underlying physical mechanism responsible for generation of the input signal is inherently nonlinear.

2. **Input-output mappings.** A popular paradigm of learning is learning with a teacher, which involves modification of the synaptic weights of a neural network by applying a set of labeled training samples or task examples. Each example consists of a unique input signal and a corresponding desired response. The network is presented with an example picked at random from the set, and the synaptic weights (free parameters) of the network are modified to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with an appropriate statistical criterion. The training of the network is repeated for many examples in the set until the network reaches a steady state where there are no further significant changes in the synaptic weights. The previously applied training examples may be reapplied during the training session but in a different order. Thus the network learns from the examples by constructing an input-output mapping for the problem at hand.

3. **Adaptivity.** Neural networks have a built- in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions. Moreover, when it is operating in a nonstationary environment (i.e., one where statistics change with time), a neural network can be designed to change its synaptic weights in real time. As a general rule, it may be said that the more adaptive we make a system, ensuring that the system remains stable all the time, the more robust its performance will likely be when the system is required to operate in a nonstationary environment (Grossberg, 1988b cited in Haykin, 1999).

4. **Evidential response.** In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to select, but also about the confidence in the decision made. This latter information may be used to reject ambiguous patterns, should they arise, and thereby improve the classification performance of the network.

5. **Contextual information.** Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt naturally by a neural network.

6. **Fault tolerance.** A neural network, implemented in hardware form, has the potential to be inherently fault tolerant, or capable of robust

computation, in the sense that its performance degrades gracefully under adverse operating conditions.

7. **VLSI implementability.** Artificial neural network computations may be carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

8. **Uniformity of analysis and design.** Basically, neural networks enjoy universality as information processors. This is in the sense that the same notation is used in all domains involving the application of neural networks.

9. **Neurobiological analogy.** The design of neural network is motivated by analogy with the brain, which is a living proof that fault tolerant parallel processing is not only physically possible but also fast and powerful. Neurobiologists look to neural networks as a research tool for the interpretation of neurobiological phenomena. Artificial neural networks are relatively crude electronic models based on the neural structure of the brain. It tries to replicate only the most basic elements of this complicated, versatile, and powerful organism.

2.3. Neural network architectures

Architecture of a neural network is the specific arrangement and connections of the neurons that are organized in the form of layers that make up the network. It is defined in terms of its size, depth and connectivity. In general, artificial neural network has a similar structure of topology. Basically, the neural network topology can be said to consist of an input layer, an output

layer, and some number of hidden layers in-between the input and output layers.

- The input layer is the sensory apparatus of the network. It consists of neurons that receive input from the external environment. There must be sufficient input nodes in the input layer to represent the input data acceptably.

- The output layer is the last layer in the network. It consists of neurons that communicate the output of the system to the user or external environment.

- The hidden layers are the layers in between the input layer and the output layer. Together with the output layer, these do the computational work of the neural network.

The fundamental facet of the network topology is the flow of signals from the input layer to the output layer. Although many schemes employed, taxonomically one draws a line between feed-forward networks and recurrent networks. The manner in which the neurons of a neural network are structured is internally linked with the learning algorithm used to train the network (Stergions and Siganos, 1996).

In general, three fundamentally different classes of architectures are identified.

1. Single-layer feed forward networks

Feed-forward artificial neural network is the type of network, which allows signals to travel one way only; from input to output. There is no feedback (loops). In other words the output of any layer does not affect that same layer. Feed-forward artificial neural networks tend to be straightforward networks that associate inputs with outputs. A feed-forward neural network with no hidden layers is known as a *perceptron*, and is said to be single-layer, or depth one network. The network in figure 2.4 is a single-layer network.

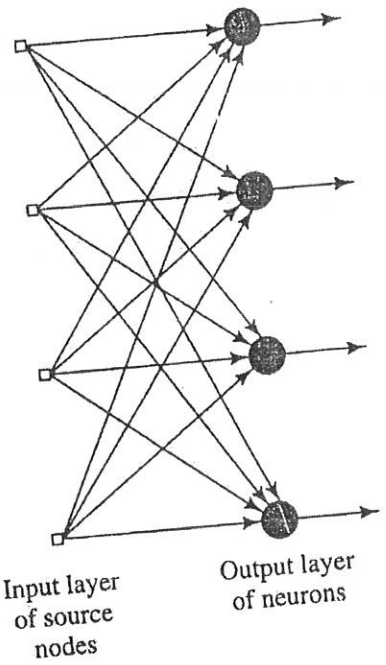


Figure 2.4 Feed-forward network with a single layer of neurons.

2. Multilayer feed-forward networks

The multilayer feed-forward networks are the generalization of the single-layer perceptron. It distinguishes from single-layer feed-forward network by the presence of one or more hidden layers, whose computation nodes are correspondingly called hidden neurons or hidden units. The function of hidden

neurons is to intervene between the external input and the network output in some useful manner. Figure 2.5 illustrates the layout of a multilayer feed-forward neural network for the case of single hidden layer.

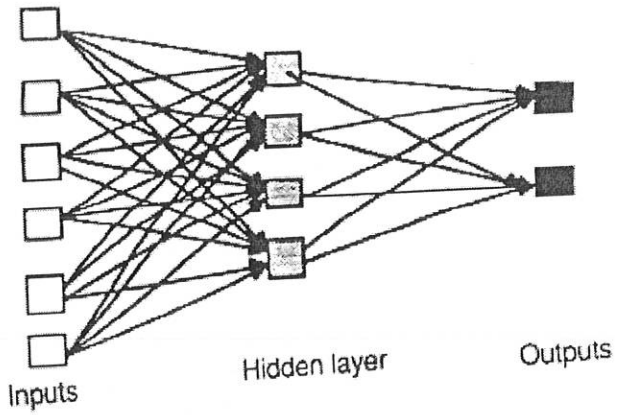
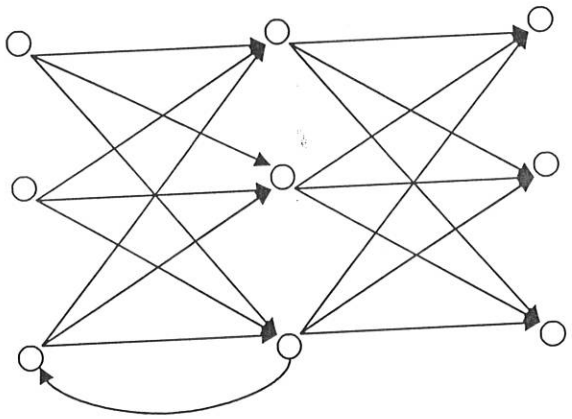


Figure 2.5 A fully connected feed forward network with one hidden layer and one output layer

3. Recurrent networks

Recurrent networks can have signals traveling in both directions by introducing loops in the network. A recurrent network may consist of a single-layer of neurons with each neuron feeding its output signals back to the inputs of all other neurons. In this kind of networks there are no self-feedback loops. They are said to be dynamic in that there state is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Figure 2.6 shows the pictorial representation of recurrent neural networks.



Input layer **hidden layer** **output layer**

Figure 2.6 Recurrent neural networks

2.4. Teaching an artificial neural network

The property that is of primary significance for a neural network is the ability to learn from its environment, and to improve its performance through learning. A neural network learns about its environment through an interactive process of adjustments applied to its synaptic weights and bias levels. Because of the particular interest in the neural networks, the definition of learning that is adapted from Mendel and MaClaren (1970) is presented by Haykin (1999) as follows:

> ***"Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes takes place."***

A prescribed set of well-defined rules for the solution of a learning problem is called a learning algorithm. There is no unique learning algorithm for the design of neural networks. Basically, learning algorithms differ from each other in the way in which the adjustment to synaptic weight of a neuron is formulated. Generally, all the learning methods of an adaptive network can be classified into two major categories.

2.4.1. Supervised learning

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached.

With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. That is, for each input set provided to the system, the corresponding desired output set is provided as well. The training is considered complete when the neural network reaches a user defined performance level.

2.4.2. Unsupervised learning

Unsupervised learning is sometimes called self-supervised learning. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Once the network has become tuned to the statistical regularities of the input data, it develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically (Becker, 1991).

2.4.3. Learning rates

It is the rate at which the artificial neural networks learn and it depends on several controllable factors. In selecting the approach there are many trade-offs to consider. Obviously, a slower rate means a lot more time is spent in accomplishing the off-line learning to produce an adequately trained system. With the faster learning rates, however, the network may not be able to make the fine discriminations possible with a system that learns more slowly.

Generally, several factors besides time have to be considered when discussing the off-line training task. Network complexity, size, architecture, type of learning rule or rules employed, and desired accuracy must be all considered. These factors play a significant role in determining how long it will take to train a network.

Usually learning rate is positive and between zero and one. If the learning rate is greater than one, it is easy for the learning algorithm to overshoot in correcting the weights, and the network will oscillate. Small values of the

learning rate will not correct the current error quickly, but if small steps are taken in correcting errors, there is a good chance of arriving at the best minimum convergence.

2.4.4. Learning laws

Many learning laws are in common use. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's rule (Anderson and McNeil, 1992). A few of the major laws are presented as follows:

a) Hebb's rule

The first learning rule was introduced by Donald Hebb. His basic rule is: if a neuron receives an input from another neuron and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened.

b) Hopfield law

It is similar to the Hebb's rule with the exception that it satisfies the magnitude of the strengthening or weakening. It states, "if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate."

c) The Delta rule

This rule is the further variation of Hebb's Rule. It is one of the most commonly used rules. This rule is based on the simple idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of the

processing element. This rule changes the synaptic weights in the way that minimizes the mean squared error of the network.

The way the Delta rule works is that the delta error in the output layer is transformed by the derivative of the transfer function and is then used in the previous neural layer to adjust input connection weights. In other words, this error is back propagated into the previous layers one layer at a time. The process of back- propagating the network errors continues until the first layer is reached.

d) The gradient descent rule

This rule is similar to the Delta rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly.

e) Kohonen's learning law

This procedure was inspired by the learning in biological systems. In this procedure, the processing elements compete for the opportunity to learn, or update their weights. The processing element with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted to output, and only the winner plus its neighbors are allowed to adjust their connection weights. Further, the size of the neighborhood can vary during the training period. The usual paradigm is to start with a larger definition of the neighborhood, and narrow in as the training process proceeds. Because the

winner element is defined as the one that has the closest match to the input pattern, Kohonen networks, model the distribution of the inputs.

2.5. Multi- layered feed-forward neural networks

Multi-layer feed-forward neural network consists of a set of sensory units (source nodes) that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. There is no theoretical limit on the number of hidden layers but typically there is just one or two. The input signal propagates through the network in forward direction, on a layer-by-layer basis as shown in figure2.7. These neural networks are commonly referred to as multilayer perceptrons (MLPs).

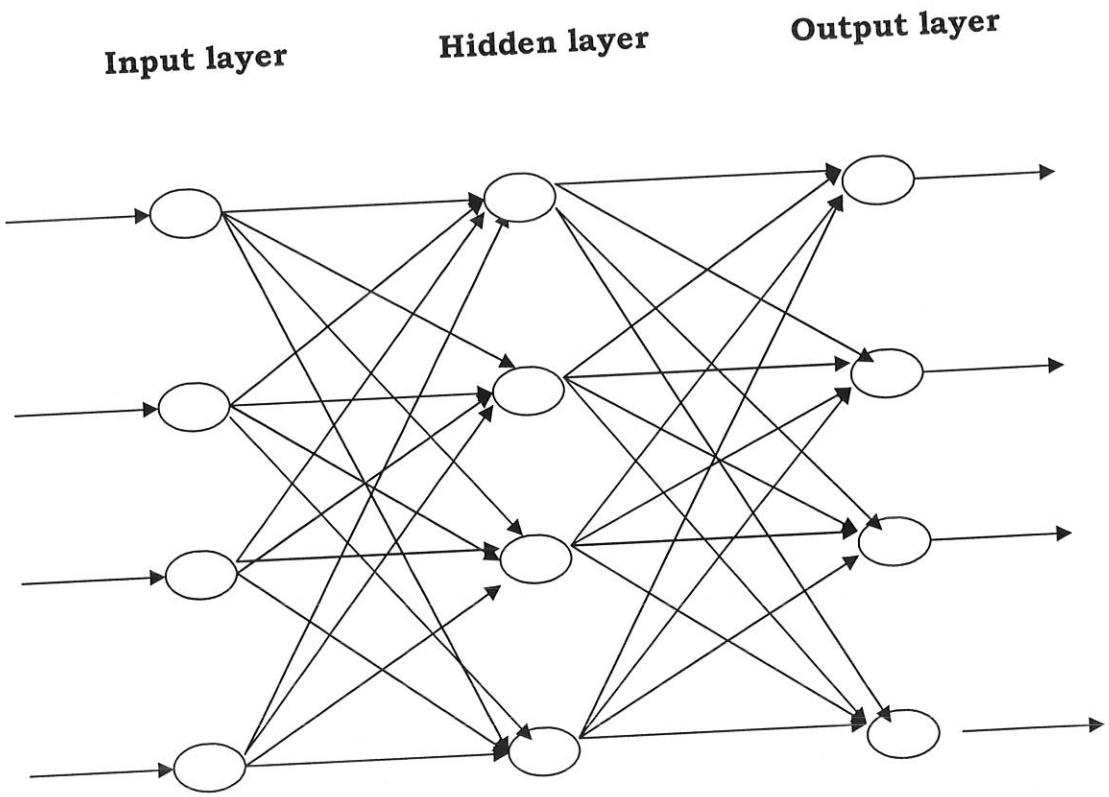


Figure 2.7 Architectural graph of a fully connected multilayer feed-forward network with one hidden layer

Multi-layer perceptrons have been applied successfully to solve some difficult and diverse problems by training them in a supervised manner with a highly popular algorithm known as the error-back propagation algorithm. This algorithm is based on the error-correction learning rule.

The number of layers and the number of processing element per layer are important decisions. There is no quantifiable, best answer to the layout of the network for any particular application. However, the best approach to find the optimal number of hidden neurons is trial and error. It is important to notice that using too few units may lead to under fitting and to the contrary, using

too many units may cause over fitting, in which case the training set will be memorized and no generalization of the data trends will occur, making the network useless on new data set.

2.5.1 Back-propagation

In order to train a neural network to perform some task, we must adjust the weights of each unit in such a way that the error between the desired output and the actual output is reduced. This process requires that the neural network compute the error derivative of the weights. In other words, it must calculate how the error changes as each weight is increased or decreased slightly. The back-propagation algorithm is the most widely used method for determining the error derivative of the weights.

The error back-propagation method can be applied to any feed-forward networks with differentiable activation function. The need for differentiability of the activation function is based on the following reason. The difference between the response of an output unit and expected response is the error made by the network. The units of the output layer use this error directly to correct their connection weights, but this is not the same for the units of the hidden layer since they are not in direct contact with the error. As a result, the units of hidden layers need to estimate their error using error back-propagation method. The amount of error made by the network is first converted into an error signal that is proportional to the rate of change of the nonlinear activation function. In fact, this implies that we want the correction to be proportional to the rate of change of the activation function. From the theory of calculus, the easiest way to fulfill this requirement is to make the correction to be proportional to the derivatives of the activation function,

which justifies the need for differentiability of the activation function everywhere.

The model of each neuron in the network includes a nonlinear activation function. The important point to emphasize here is that the nonlinearity is smooth, as opposed to the hard limiting. A commonly used form of nonlinearity that satisfies this requirement, having values in the range [0, 1], is a sigmoidal nonlinearity defined by the logistic function:

$$\varphi(x) = \frac{1}{1 + \exp(-x)} \tag{2.7}$$

In back-propagation network the units within the hidden layer and output layer are usually nonlinear. This unit computes their activation level by adding all the weighted activations they receive and transfer this level into a response using the non-linear activation, in this case, the sigmoid function.

Basically, error back propagation learning consists of two phases: a forward phase and backward phase.

Forward phase

In forward phase, an activity pattern (input vector) is applied to the sensory nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of output is produced as the actual response of the network. During the forward pass, the synaptic weights of the network are all fixed. This phase ends with the computation of an error signal $e_i = d_i - a_i$

where, d_i is the desired response and a_i is the actual output produced by the network in response to the input x_i.

Backward phase

During this phase, on the other hand, the synaptic weights are all adjusted in accordance with an error –correction rule. Specifically, the actual response of the network is subtracted from a desired (target) response to produce an error signal. This error signal is propagated backward through the network, against the direction of the synaptic connections-hence the name "error back-propagation."

In other words, the error –back propagation is the way of using known input-output pairs of a target function to find the coefficients that make a certain mapping function approximate the target function as closely as possible. The task faced by a back- propagation neural network, such as MLPs, is that of learning supervised mapping: given a set of input vectors and associated target vectors, the objective is to learn a rule that captures the underlying functional relationship between the input vectors and the target vectors. Mathematically, each target vector $\vec{d}_i$ is a function of the input vector $\vec{x}_i$.

$$\vec{d}_i = f\left(\vec{x}_i\right) \tag{2.8}$$

The task of the back-propagation network is to learn the function f. This is achieved by finding regularities in the input patterns that correspond to regularities in the output patterns. The network has a weight parameter

vector, whose values are changed to modify a function computed by the network to be as close as possible to f. .

For a given training set, back propagation learning may be implemented in one of the two basic ways, as presented by Haykin (1999).

1. **Sequential mode**. The sequential mode of back-propagation learning is also referred to as on-line, pattern, or stochastic mode. In this mode of operation weight updating is performed after the presentation of each training example. To be specific, consider an epoch consisting of N training examples arranged in the order (x(1), d(1)),...,(x(N),d(N)). The first example pair (x(1),d(1)) in the epoch is presented to the network, and the sequence of forward and backward computations described previously is performed, resulting in certain adjustments to the synaptic weights and bias levels of the network. Then the second example pair in the epoch is presented, and the sequence of forward and backward computations is repeated, resulting in further adjustments to the synaptic weights and bias level. This process is continued until the last example pair in the epoch is accounted for. This approach is best suited for pattern classification.

2. **Batch mode**. In the batch mode of back-propagation learning, adjustments are made to the free parameters of the network on an epoch – by-epoch basis, where an epoch consists of the entire set of training examples.

The mathematical presentation of the back propagation is as follows:

In the first phase, a forward flow of activation is generated from the input layer to the output layer through the hidden layer. Each unit in the hidden layer computes its activation level as a weighted sum of its inputs. The induced local field $v_j(t)$ produced at the input of the activation function associated with the neuron j is therefore

$$v_j(t) = \sum_{i=0}^{n_x} b_{ij} x_i(t) \tag{2.9}$$

Where n_x is the number of input features (x_i) and b_{ij} is the weight from the input (ith) layer to the hidden (jth) layer. Then an activation function is applied to $v_j(t)$ to compute the output from the hidden nodes, h_j:

$$\begin{aligned} h_j(t) &= \varphi(v_j(t)) \\ &= \frac{1}{1+\exp(-v_j(t))} \end{aligned} \tag{2.10}$$

Next, $h_j(t)$ becomes the net input to the output nodes (q_k) which is calculated by

$$q_k = \sum_{j=0}^{n_h} d_{jk} h_j \tag{2.11}$$

Where n_h is the number of hidden nodes, d_{jk} is the weight from the (jth) layer to the output (kth) layer. Again an activation function is applied to q_k to compute the predicted output, p_k:

$$p_k = \varphi(q_k) = \frac{1}{1 + \exp(-q_k)} \tag{2.12}$$

To accomplish the goal of optimizing the weights in equations (2.9) and (2.11), an error back-propagation algorithm is commonly used to minimize the objective function defined as

$$E = \frac{1}{2} \sum_{i=1}^{n_x} \sum_{k=1}^{n_k} (y_k - p_k)^2 \tag{2.13}$$

Where n_k is the number of the output classes. This is the sum of squared difference between the predicted output (p_k) and the observed output (y_k) averaged over all input and output patterns. Evaluating the partial derivatives of equations (2.13) with respect to the weights in equations (2.9) and (2.11) (i.e., b_{ij} and d_{jk}), we will drive two error terms for the output layer (δ_k) and for the hidden layer (δ_j) as follows:

$$\delta_k = (y_k - p_k)\varphi'(q_k) \tag{2.14}$$

$$\delta_j = \varphi'(v_j) \sum \delta_k d_{jk} \tag{2.15}$$

If the two errors are not sufficiently small, iterative techniques are used to find the optimal set of weights for the network. Each iteration is considered a training period. By updating the weights, the neural network is said to be learning. The connection weights are adjusted using the error back-propagation algorithm based

on the generalized delta rule such that the weights between the two layers are computed iteratively as:

$$\Delta d_{jk}(t+1) = \eta \delta_k h_j + \alpha \Delta d_{jk}(t) \tag{2.16}$$

$$\Delta b_{ij}(t+1) = \eta \delta_j x_i + \alpha \Delta b_{ij}(t) \tag{2.17}$$

Where $\Delta d_{jk}(t+1)$ is the change of the weights between the hidden layer and the output layer at the $(t+1)^{th}$ iteration, $\Delta b_{ij}(t+1)$ is the change of the weights between the input layer and the hidden layer at the $(t+1)^{th}$ iteration, $\Delta d_{jk}(t)$ is the change of the weights between the hidden layer and the output layer at the t^{th} iteration, $\Delta b_{ij}(t)$ is the change of weights between the input layer and the hidden layer at the t^{th} iteration, η is the learning rate and α is the momentum term. The above procedure is repeated for all training samples until the network errors are less than a predefined threshold.

2.5.2 Generalization, overfitting and stopping criteria

In back-propagation learning, we typically start with a training sample and use the back-propagation algorithm to compute the synaptic weights of multilayer by loading (encoding) as many of the training examples as possible into the network. The hope is that the neural networks so designed will generalize. A network is said to generalize well when the input-output mapping computed by the network is correct, for data never used in creating or training the network.

The learning process (i.e., training of a neural network) may be viewed as a curve-fitting problem. The network itself may be considered simply as nonlinear

Reducing the size of neural networks (e.g., reducing the number of hidden neurons and interneuron connections according to prior knowledge) is also an effective way to reduce overlearning.

Chapter Three

Tree volume growth forecasting

3.1. Introduction

Forest management decisions are predicted on information about both past and current resource conditions. Inventories taken at one instant of time provide information on current volumes and related statistics. Forests are dynamic biological systems that are continuously changing, and it is necessary to project these changes to obtain relevant information for prudent decision-making.

Stand dynamics, i.e., growth, mortality, reproduction and associated changes in the stand, can be predicted through direct or indirect methods. Direct methods; such as the stand table projection technique involve field observations in existing stands. Past growth and mortality trends are used to infer future trends in the stands observed.

There are situations in which direct observation of forest growth and mortality are not feasible. Diameter growth, mortality and in growth relationships developed through stand- table projection techniques are not reliable for long periods of time. Also, the costs of direct observation are sometimes prohibitive consequently, foresters often rely on indirect methods of predicting stand dynamics – i.e. growth, mortality, and related quantities of a stand are inferred from the study of other stands. These inferences are made through the use of tables, equations, or computer simulator models. Techniques for forecasting stand dynamics are collectively referred to as growth and yield models (Avery and Burkhart, 1994).

Growth forecasts may be required for a short term or long term basis, for the overall stand volume or volume by product and size classes. With the wide variety of existing stand conditions and the diverse objectives and needs of users of growth and yield models, it is not surprising that numerous approaches have been proposed. These approaches range from models that provide only a specified aggregate stand volume to models with information about individual trees. Regardless of the structural complexity and amount of output details provided, all growth and yield models have a common purpose: to produce estimates of stand characteristics [such as the volume, basal area (BA), and number of trees per unit area] at specified points of time.

3.2. Tree growth

Tree growth is the increase in its size with time. Growth takes place simultaneously and independently in different parts of a tree and can be measured by many parameters for example growth in diameter, in height, etc.

The patterns of the growth of a tree in terms of the changes in diameter, basal area, height form and volume are affected to varying degrees by the site, crop structure, competition and stocking. Even aged crops of one species have very different growth patterns from uneven aged crops of several species (Philip, 1994).

The growth pattern of most living organisms follows a sigmoid pattern with slow initial and terminal growth rates, fastest growth during the middle of their life and a maximum final size. This pattern is mirrored in the commonly used growth model namely the logistic function (Philip,1994).The graphs with such characteristics showing the cumulative size of an object at any age is known to be

sigmoid growth curve. A true growth curve, which shows increment at any age, results from plotting increment over age.

The S shaped form of the cumulative growth curve is evident for individual cells, tissues, and organs, and for individual plants and animals for the full life span. Also, the pattern of growth for short growing periods, such as growing season, tends to follow the S shaped curve. Figure 3.1 shows the pattern followed by the growth of an organism.

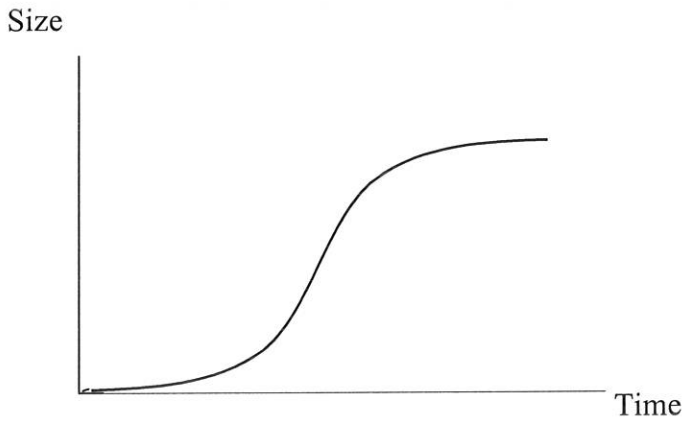


Figure 3.1 A stylized sigmoid shaped growth curve

Although the exact form of the cumulative growth curve will change when the tree dimensions, i.e., height, diameter, basal area, and volume, plotted over age is changed, the cumulative growth curve has the characteristics that hold for all dimensions of a tree.

3.2.1. Diameter and height growth

Characteristically, a cumulative growth curve of height over age for trees shows a juvenile period of less than a decade, a long maturing period when the trend is nearly linear, and leveling off in old age. A cumulative growth curve of diameter over age shows much the same trend, there is, however, more of a tendency toward curvilinearity during the period of maturity. Since diameter is usually measured at breast height, dbh cannot be measured until a tree is over 4.5 feet tall. Consequently, since some of the early growth is before measurement begins, curves of dbh over age may not reflect some of the early growth (Husch et al., 1982).

Past height and diameter growth of individual trees may be determined from repeated measurements of total size at the beginning and at the end of specified growing period and from increment measurements of past growth.

1. Repeated measurements

Height growth of an individual tree can be obtained by measuring the total height of a standing tree at the initiation and at the cessation of a specific growing period and by taking the difference. Tree heights up to 75 feet can be accurately and precisely determined with telescopic measuring poles that can be purchased from forestry and engineering supply companies.

Diameter growth of an individual tree can be obtained by measuring the diameter at the beginning and at the end of a specified period and by taking the difference. Since annual diameter increment is small, when instruments such as calipers and diameter tapes are used, measurements are frequently taken at intervals of several

years. However, for short periods, even for a day diameter growth can be obtained with more precise instruments.

2. Increment measurements

Increment measurements of past height growth can be made quickly if a reference point is marked on a tree. Past increment for intervals during a growing season, for a growing season, or for a specified period may be measured from this reference point. Also, past height increment may be determined by stem analysis. And for species for which the internodal lengths on the stem indicate a year's growth, past height growth may be determined by measuring internodal lengths. In regions where tree growth has a seasonal or annual growth pattern, past diameter increment can be obtained from increment borings or cross sectional cuts. Borings or cross sections can be secured at any point along the stem. However, diameter increment is most often determined at breast height from increment borings.

3.3. Volume of the tree

Volume of the tree is the measure of solid content or capacity, usually expressed in units that are cubes of linear units. There are various methods of determining the tree volume over time. These are direct methods of determining the cubic volume of the tree parts by developing the relationship between the various dimensions and direct methods of determination of cubic volume by using the previously established formulas.

Direct volume determinations of parts of trees are usually made on sample trees to obtain basic data for the development of relationships between the various dimensions of a tree and its volume. Relationships of this type are used to

estimate the volume of other standing trees. In the past, sample tree measurements were often taken on trees cut in harvesting period. But volume relationships development from such measurement may lead to bias because they may not be representative of all the trees in a stand. Thus, there is a growing tendency to take measurements on representative sample of standing trees.

The direct determination of the volume any part of a tree involves clearly defining the part of the tree for which volume is to be determined and carefully taking measurements in accordance with the constrains imposed by the definition.

To use formulas or graphical techniques to determine the volume of a stem, it is necessary to have diameter measurements at various intervals along the stem. Volume tables generated using previously developed volume growth model equations can be used in estimating the future value of tree volume.

3.3.1 Estimation of tree volumes

Tree volumes can be estimated from previously established, relationship between certain tree dimensions and tree volume. Diameter, height and form factor are the independent variables that are commonly used to determine the values of the dependent variable- tree volume. The final result is presented in formulas or table form. The volume formula or volume table, then, gives the average contents of individual trees in terms of one or more of the previously mentioned tree dimensions.

a) Local volume tables

Local volume tables give tree volume in terms of diameter at breast height only. The term local is used because such tables are generally restricted to the local area

for which the height – diameter relationship hidden in the table is relevant. Although local volume tables may be prepared from raw field data- that is, from volume and diameter measurements for a sample of trees- they are normally derived from standard volume tables.

b) Standard volume tables

Standard volume tables give volume in terms of diameter at breast height and merchantable or total height. Tables of this type may be prepared for individual species, or groups of species, and specific localities. The applicability of a standard volume table however, depends on the form of the trees to which it is applied rather than on species or locality. For each diameter height class the form of the trees to which the table is applied should agree with the form of the tree from which the table was prepared.

c) Form class volume table

Form class volume tables give volume in terms of diameter at breast height, merchantable or total height and some measure of form, such as absolute form quotient. Such tables come in sets, with one table for each form class. The format of each table is similar to that of a standard volume table. Note that if a single form class table is chosen as representatives of a stand, volume determinations may be in error because it is unlikely that all trees will be of the same form class. Furthermore, since form class varies with tree size, species, and site, it is unlikely that variations in form class will be random. Thus, it is difficult to obtain an accurate average form class for a stand and is therefore undesirable to use a single form class table for any extensive area.

In this study, the volume table generated with a program developed by Ola (1986) Department of Biometry Swedish University of Agricultural Sciences for Ethiopian case was used. Data are presented for combinations of diameter and height within the limits of the material of trees. The volume table for *Cupressus lustanica* species was generated by making use of the following formula:

$$\ln (v) = -3.2160+1.80960*\ln (D) +1.14920*\ln (H) \quad (3.1)$$

Where,

V is the volume of the tree

D is the diameter at breast height

H is the height

In an applicability check, one should compare the volume of sample trees with the estimated volume from the volume table to be checked.

3.4. Review of literatures

In recent years, neural networks or ANN have become popular as an alternative method to traditional statistical methods. Current ANN applications include forest growth and dynamic modeling, land mapping and classification, spatial data analysis and GIS modeling, soil type classification, plant disease dynamics and insect pest management, marine ecosystem, and global and climate change research (Lek and Guegan, 1999; Peng and Wen, 1999 cited in Chuangmin et al., 2003). Some of the researches conducted elsewhere in areas of forestry and employed neural networks as a tool are reviewed below.

Gordon (1999) examined the applicability of Bayesian Neural Network (BNN) in modeling forest tree growth. According to his findings the ANN parametric form is shown to be a suitable model if each forest tree plot is assumed to consist of several differently growing sub-plots. The predictive Bayesian approach is used in estimating the ANN output. In his study, data from correlated curve trend experiments were used. The ANN predictions were compared with those of one of the best parametric solutions, the Schnute model i.e., growth model. ANOVA methods were used to evaluate whether any observed differences are statistically significant. According to the result he obtained, a Bayesian ANOVA indicates that there is a 93% probability of the ANN approach producing better predictions on the average.

In another related study, Zhang et al. (2000) experimented the prediction capability of the ANN in tree-ring growth of Douglas fir in response to climate data. They used the climate data, obtained from the Ministry of Forests, Canada, which include monthly mean temperature and total monthly precipitation from 1891-1992 as an input to the neural network. They employed the three-layer neural network for modeling the nonlinear and complex climate-growth response and also used the two- layer linear neural network for comparison with the performance of the nonlinear models. According to their findings, the ANN models are capable of predicting tree-ring growth for given climatic conditions, and the accuracy of the predictions can be evaluated from the error distributions on the testing sets. The results of their ANN modeling show that with similar memorization ability in the training sets, all the nonlinear neural networks had better generalization capability than the linear models as measured by percentage error distribution and the amount of explained variance in the tree ring series. The errors in nonlinear models, compared to those of linear models, were more concentrated in the small error ranges, and the likelihood of large errors was less.

For the 300 testing ring-width indices, the nonlinear models explained 66.82% of the variance, whereas the linear models explained only 51.32% of the variance, i.e., an improvement of 15.5 % by the nonlinear models. The nonlinear models predicted growth responses more accurately than the linear models. The nonlinear models, developed by using different training sets, varied in their predictions although the variation was not great. This suggests the existence of unknown distributed noise in the different training-testing sets affecting the performance of the neural networks.

In another related study, Corne et al. (2000) used artificial neural network methods to predict forest characteristics in southeast Alaska. Data were collected by field workers by visiting sites arranged on a grid with a separation of 4.8km, although several sites proved to be inaccessible. According to them, modeling the distribution of species and forest characteristics permits the interpolation of such data between survey locations and also allows the prediction of characteristics in unsurveyed regions. They used the learning vector quantization method to generate models based upon simple inventory parameters such as geographical location, elevation, slope and aspect with complementary satellite image data. They generated predictive maps by obtaining input data from digital elevation models. The results obtained reveals that the LVQ neural network is a quick and inexpensive approach to generate models for the spatial prediction of forest characteristics as opposed to the published classification maps based upon the inventory data.

In other study, Liu et al. (2003) used two ANNs and three traditional statistical classification methods to classify Forest Inventory and Analysis (FIA) plots into six ecological habitats in U.S. Northeast. They identified four variables (overstory and understory species composition, hardwood basal area percentage, and current

FIA forest type) from the list of available stand variables for habitat classification. They used error matrix and accuracy indices to assess the classification accuracy of the models and to test the difference between the five classifiers. The result obtained reveals that the ANN model (MLP and radial basis function (RBF)) are superior to the traditional statistical methods such as linear discriminate analysis and minimum-distance classification. The classification accuracy of the ANN models is 90% or higher for overall classification. The K-Nearest Neighbor (KNN) method classifies the six ecological habitats as accurately as the two neural networks. This study shows that the ANN models and KNN method have a great potential for the classification of ecological habitats using FIA data, due to their flexibility of modeling algorithms and robustness to the problems in FIA data such as non-Gaussian distributions, nonlinear relationships, outliers and noise in the data.

According to Hasenanuer, et al (2001), neural network is also used for forecasting individual tree mortality of Norway Spruce Stands. They used multi-layer perceptron, learning vector quantization and cascade correlation networks as a different formalisms for mortality prediction. The data set for parametrizing the model and training the different neural network types comes from the Austrian National Forest Inventory. After training the different network types, they evaluate the resulting mortality prediction using an independent data set. The result they obtained indicates that multi -layer perceptron and cascade correlation networks perform the best predictions.

Tracz (2004), in his study of new analytical methods in forestry, integrated expert systems (ES) and ANN with geographic information system (GIS) in solving complex problems. Results of the research on integrating ES and ANN with GIS prove that functional integration in the sense of good understanding of a spatial

problem and properly addressing functions and data for GIS and modeling module, is possible. Technical integration, which is understood as a higher effectiveness of using various software in one coherent system-application dedicated to chosen problem solving, is also possible.

Chapter Four

Experimentation

4.1. Data collection, analysis and processing

One of the most important components in the success of any neural network application is the data. The quality, availability, reliability and relevance of the data used to develop and run the system are critical to its success. The task of building a model can be successful if the input data has been processed in such a way that it clearly reveals the important information. In contrary, if the input data to be presented to the network is complex, the task of developing a model becomes challenging or even unsuccessful. Designing of any neural network involves the following major steps shown in figure 4.1.

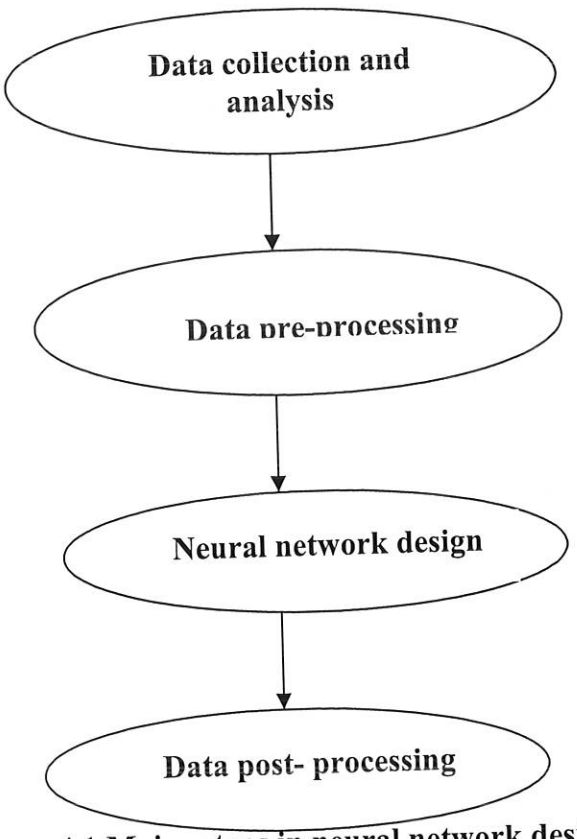


Figure 4.1.Major steps in neural network design

Data processing starts from data collection and analysis, followed by preprocessing and then feeding into the neural network. Finally, post processing is needed for the purpose of transforming the outputs of the network to the required outputs, if necessary.

4.1.1. Data collection

The data collection phase typically consists of the following major duties: -

1. Identifying the data requirements

The primary task to be performed while planning data collection is the decision as to what type of data is needed to solve the problem at hand. To come up with good decision of what type of data is required to solve the problem, the primary discussion was made with the experts in the problem area. In addition, different literatures were reviewed in the problem domain to critically select the relevant data.

Input data

The input data used for forecasting tree volume growth are historical tree volumes of the particular species taken from permanent sample plots from age of six through age of twelve. According to the discussion made with the experts, these tree volumes are computed empirically by employing DBH and height as independent variables. Since these tree dimensional parameters are measured on annual basis, the tree volumes obtained are also annual values. As an input to the neural networks, consecutive seven years historical tree volumes were used to forecast the volume of that particular tree at the forecast year. The inputs to the network correspond to V(6), V(7), V(8),V(9), V(10), V(11) and V(12) and are

given in units of cm^3. These are the volumes of the tree at ages of 6, 7, 8, 9, 10, 11 and 12 years respectively. The tree dimensional parameters are assumed to be insignificant up to the age of five. Table 4.1 shows a sample of the original input data used in the research.

V(6)	V(7)	V(8)	V(9)	V(10)	V(11)	V(12)
84.43	94.68	105.68	117.46	150.61	190.71	231.57
60.94	94.93	137.95	187.22	203.79	237.67	275.02
58.56	76.79	98.04	121.13	136.51	149.87	164.06
69.40	98.39	132.74	172.82	205.63	249.38	295.91
44.51	61.74	82.32	107.83	119.36	152.34	190.24
46.64	77.66	119.39	168.41	183.77	207.22	234.02
71.85	91.24	113.79	136.58	163.48	200.26	243.64
65.89	86.99	110.28	135.77	156.79	185.49	215.46
74.09	117.46	174.34	249.37	294.32	350.42	417.12
62.79	64.90	94.21	128.83	154.96	173.43	191.09

Table 4.1 Sample of the original input data used in the research

Output data

The output of the tree volume forecaster is the tree volume of the forecast year, i.e. V (13), which is the volume of the tree at the age of 13 years. Table 4.2 shows a sample of the original target data set used to train the network

V(13)
274.48
321.06
184.64
339.22
211.27
259.32
271.17
235.67
476.73
231.57

Table 4.2 Sample of the original target data used

2. Identifying data sources

The next step is to decide from where the data will be obtained. Since the model to be developed is a tree volume-forecasting model, the appropriate source to get the data is the forestry research center. The FRC collects and records forest inventory data for the purpose of developing a model using conventional approaches, which in turn enables for the management of forest resources. This is the institution in the country highly engaged in conducting different forestry research activities. Therefore, the relevant and reliable data for conducting this research work is available only at the aforementioned government institution and hence the data is secondary data.

3. Determining the data quantity

It is important to make a reasonable estimate of how much data will be needed for developing the neural network model. The amount of data used in the training process should be properly determined to enable the NN to learn different properties of the data. If the data collected is limited in number, it may not reflect the full range of properties that the network should be learning and this will in turn reduces the performance of the network with new data. In contrary, collecting too much data will introduce extra expenses. Therefore, based on the experts' opinion, 8 years of forest inventory data for 1155 individual trees of *Cupressus lustanica*, i.e. from 1994 to 2001, were collected for developing a model using the neural network.

4.1.2 Data preprocessing

Today's real world databases are highly susceptible to noisy, missing and inconsistent data due to different reasons, such as attributes of interest may not always be available, relevant data may not be recorded due to misunderstanding of the subject under consideration, instrument used may be faulty, etc.

There are a number of data preprocessing techniques. Data cleaning, data integration, data transformation and data reduction are the most commonly used techniques that help to improve the overall quality of the data.

1. Data cleaning

This routine cleans the data by filling in the missing values, smoothing noisy data, identifying or removing outliers, and resolving inconsistencies. Dirty data can cause confusion for the mining procedure, resulting in unreliable output (Han and Kamber, 2001).

and v(12)) and the volume of the forecast year(i.e., v(13)) is computed. The correlation coefficient obtained between past volumes from the age of six through twelve and the volumes of the forecast year are 0.77, 0.83, 0.86, 0.88, 0.90, 0.92 and 0.93 respectively. These figures show that there exists a strong positive correlation between the historical volumes and volume of the forecast year. Therefore, the past tree volumes are good predictors of the future value of tree volume and hence used as an input in the development of the forecast model. Table 4.3 presents the original sample input data with all attributes.

Planting year 1988																					
plot number 5																					
	Years of measurement																				
	1st 1994			2nd1995			3rd 1996			4th 1997			5th 1998			6th 1999			7th 2000		
TREE NO.	DBH	HT	V(6)	DBH	HT	V(7)	DBH	HT	V(8)	DBH	HT	V(9)	DBH	HT	V(10)	DBH	HT	V(11)	DBH	HT	V(12)
1	15.15	10.5	81.83	16.20	11.75	105.12	17.25	13	132.28	18.30	14.50	166.89	19.2	16	203.85	19.85	16.25	220.40	20.5	16.5	237.76
2	15.6	11.5	95.79	16.90	12.75	124.65	18.2	14	158.72	19.50	15.00	194.66	20.3	17.25	245.83	21.2	17.38	268.21	22.1	17.5	291.46
3	14	8	51.90	15.00	9.50	71.63	16	11	95.28	17.00	12.50	123.16	17.5	13	135.77	17.75	13.5	145.48	18	14.25	158.77
4	14.5	11	79.73	15.50	12.50	104.20	16.5	14	132.91	17.50	15.50	166.19	17.9	16.25	182.79	18.45	17.25	206.79	19	18.25	232.67
5	13.6	11.5	74.73	15.80	12.50	107.88	18	13.5	149.21	20.20	14.50	199.56	21.2	15.75	239.51	22.6	17.00	293.56	24	18.25	355.09
6	11.8	11.25	56.35	13.20	12.50	77.92	14.6	13.75	104.33	16.00	15.00	136.08	16.9	17.5	179.37	17.58	18	198.98	18.25	18.5	219.73
7	11.75	9	43.27	12.10	10.00	51.51	12.45	11	60.51	12.80	12.00	70.32	13.7	15	102.76	14.23	16	118.54	14.75	17	135.63
8	10	7.75	27.22	11.20	10.00	44.78	12.4	12.25	67.98	13.60	14.50	97.54	13.9	15.25	107.52	14.7	15.38	120.14	15.5	15.5	133.42
9	11.6	10	47.72	13.40	12.00	76.40	15.2	14	114.57	17.00	16.00	163.55	17.7	16.25	179.11	18.85	16.5	204.27	20	17.25	239.29
11	14	9.5	63.23	15.00	11.25	87.00	16	13	115.45	17.00	15.00	151.86	17.2	16.25	170.06	17.48	17	184.42	17.75	17.5	196.03
13	13.05	10	59.06	13.70	11.25	73.83	14.35	12.5	90.63	15.00	14.00	111.85	15.3	16.5	140.03	16.9	17.25	176.43	18.5	18	218.22
14	14.45	10.5	75.11	15.50	12.00	99.42	16.55	13.5	128.17	17.60	15.00	161.70	18.7	16.75	204.85	19.1	17	216.50	19.5	17.25	228.58
15	8.3	7	17.28	9.30	9.00	28.34	10.3	11	42.94	11.30	13.00	61.53	12	13.75	73.16	12.3	14.5	81.32	12.6	15	88.32
16	9.6	9	30.02	11.00	10.75	47.10	12.4	12.5	69.58	13.80	14.00	96.19	14.8	16.25	129.56	15.65	17.25	153.52	16.5	18.25	180.25
17	10.5	8	30.83	12.00	9.50	47.84	13.5	11	70.06	15.00	12.50	98.19	15.4	13.5	112.51	16.2	14.5	133.86	17	15.25	154.77
18	11.5	7.5	33.75	11.90	9.25	45.69	12.3	11	59.20	12.70	13.00	76.01	13.2	13.5	85.12	13.45	14.25	93.71	13.7	15	102.76
19	13.4	8	47.94	14.60	9.75	70.28	15.8	11.5	98.02	17.00	13.50	134.54	17.6	14.5	155.52	18.05	155	175.76	18.5	16.25	194.02
20	19.25	10.25	122.77	21.00	11.50	164.02	22.75	12.75	213.46	24.50	14.00	271.79	25.5	15	316.30	26.2	16	357.77	26.9	16.75	395.53
21	10.85	8.75	36.27	12.00	10.50	53.67	13.15	12.25	75.61	14.30	14.00	102.59	14.4	14.75	110.31	14.95	16	129.62	15.5	17	148.36
22	13.55	8.5	52.45	13.70	10.25	66.34	13.85	12	81.10	14.00	13.50	94.68	16	14	125.71	16.1	15	137.63	16.2	16	149.89
23	13.15	8.25	48.00	14.10	10.00	67.93	15.05	11.75	92.01	16.00	13.50	120.56	16.2	14	128.57	16.3	14.5	135.36	16.4	15.25	145.03
25	15.5	12.5	104.20	16.00	14.00	125.71	16.5	15.5	149.40	17.00	17.00	175.35	18.1	19	223.21	18.4	19.25	233.43	18.7	19.5	243.95
26	13.6	12.5	82.24	15.00	14.00	111.85	16.4	15.5	147.77	17.80	17.00	190.57	18.9	17.5	219.61	19.95	18	250.15	21	18.5	283.26
27	14.35	10.25	72.15	15.90	12.50	109.11	17.45	14.75	156.17	19.00	17.00	214.45	19.5	17.5	232.39	20.45	18	261.61	21.4	18.5	293.10
28	11.95	9.25	46.04	12.70	10.50	59.46	13.45	11.75	75.07	14.20	13.00	93.02	14.4	14	103.89	14.55	15	114.59	14.7	15.75	123.47

Table 4.3 Sample of the original data with all attributes

testing data, even if it works perfectly on the training data, i.e., the network is overtrained. In order to keep the model simple, only one hidden layer with 30 hidden neurons is used. This number was arrived after analyzing 10, 15, 20, 25, 30, 35, 40, 45 and 50 neurons in the hidden layer. The architecture with 10, 15, 20 and 25 neurons in the hidden layer was faster in computation but the convergence rate was very low. The architecture with 35, 40, 45 and 50 neurons in the hidden layer was converging equally well as that with 30 neurons. Therefore, the architecture with 30 neurons in the hidden layer was selected. Figure 4.2 shows the structure of the annual tree volume forecast model.

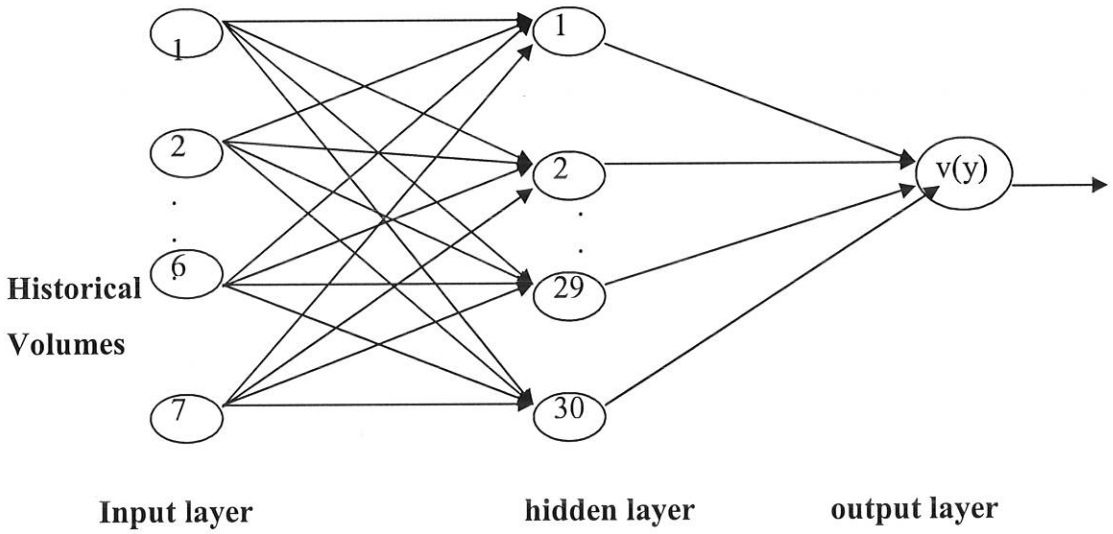


Figure 4.2 Architecture of the volume forecast model

4.2.1 Splitting of data sets

The final result of any neural network application depends entirely on the quality and quantity of the data being fed into the network. Therefore, large data samples are required for training purpose. The selected historical volumes and the forecast output volume from the period of 1994 -2001 formed the database for developing the neural network model. In this research work, the total data size employed for developing the model is 1155. This

input –output data pairs are partitioned into training, validation and testing sets. Training set is used actually to train the network and must have enough data to be representative for the overall problem. The validation set is used to check the generalization ability of the network with a data set that is not part of training set and the testing data set is independent of the training set and used to examine the final performance of the network in forecasting the volume of a tree. Finding an optimal split of the entire data into training, validation and testing sets is somewhat challenging. But according to the literatures reviewed, the forecasting accuracy of the ANN model is achieved when the training data is approximately 60% of the entire data, and the validation and testing data are 40% and should be representative of the existing patterns of the entire data. The validation and testing data are also further divided into the ratio of 1:2 respectively. Such division was done to compromise the two facts: the larger the training set the better the forecasting and the larger the test set, the more accurate the error estimate. The partitioned data with their corresponding number is shown in the table 4.6

Sets	Inputs size	Targets size
Training	693	693
Validation	154	154
Testing	308	308

Table 4.6 Partitioned data set

4.2.2 Training and testing the networks

After the network architecture has been designed, it must be trained. The training algorithm used in the research is the back propagation algorithm based on the steepest descent method that performs stochastic gradient descent on the error surface.

The transfer function being employed both in the hidden and output layer is sigmoid. This is because of the fact that the sigmoid activation function maps the data between 0

and 1. For the training process to converge in reasonable time, the desired output values should be scaled within this interval (Haykin, 1999).

Since the algorithm used for the training process is iterative in its nature; there has to be some stopping criteria. Training can be stopped after a fixed number of iterations (epochs) or after the error decreases below some specified limit. In this work, fixed number of epochs was followed as the stopping criterion.

The learning rate parameter and momentum term were also adjusted intermittently to speed up convergence. For various combinations of the learning rate parameter and the momentum term, the convergence rate was very slow after 25000^{th} iteration. Hence, the maximum number of epochs is set to 25000 as the stopping criterion and the network training performance was also checked. Table 4.7 shows only some of the results obtained out of a large number of attempts made by varying both the values of learning rate and momentum term during the training process.

Learning rate	Momentum term	MSE
0.1	0.6	0.0056
0.2	0.3	0.0043
0.3	0.3	0.0047
0.4	0.5	0.0036
0.5	0.3	0.0035
0.6	**0.5**	**0.0025**

Table 4.7 Sample network results of the tree volume forecaster with 30 hidden neurons and various combinations of learning rates and momentum term.

From the table, it can be viewed that the network achieved the highest training accuracy when the learning rate and momentum term are 0.6 and 0.5 respectively. Therefore, the network architecture having one hidden layer of 30 neurons with learning rate 0.6 and

momentum term 0.5 was found to be an optimal model for the problem resulting in the training performance of 0.0025(i.e. the network with least generalization error). Figure 4.3 shows the network convergence for $\eta = 0.6$ and $\alpha = 0.5$.

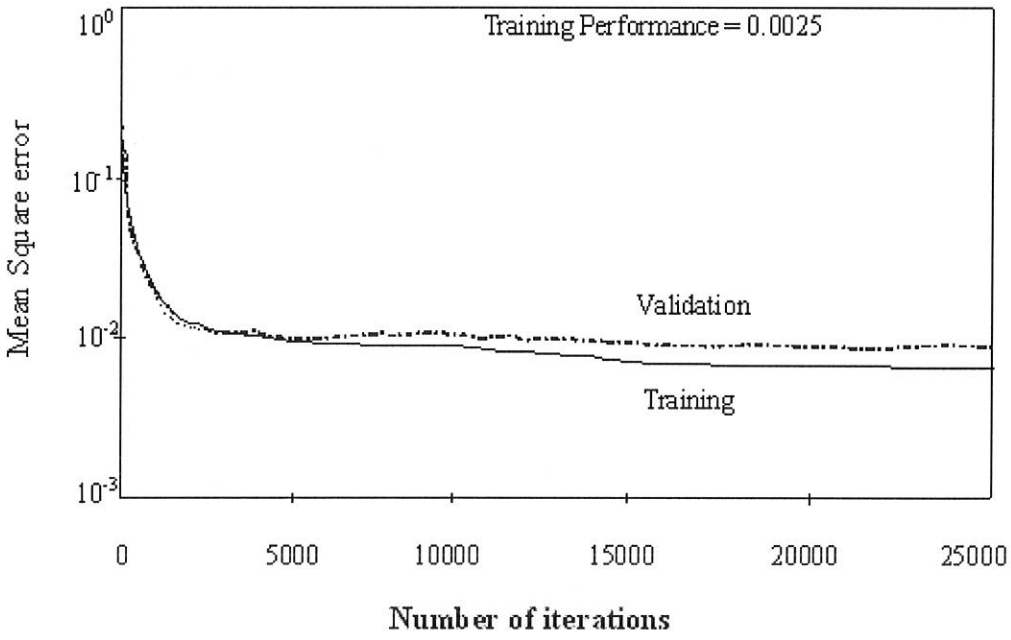


Figure 4.3. Convergence of the training algorithm

4.2.3 Analysis of the results

Once the network with optimal network architecture is determined, it will be evaluated with the independent data set reserved originally for testing purpose. At this point, the difference between the desired and the actual output of the network is computed. The performance of the network is evaluated based on the different network performance measures which use the desired and actual volumes as the key variables. Table 4.8 presents some of the normalized desired and actual outputs with their corresponding error, and table 4.9 presents some of the unnormalized desired and actual outputs with their corresponding error.

Desired output	Actual output	Error
0.2368	0.2851	-0.0483
0.1063	0.1044	0.0019
0.3000	0.3245	-0.0245
0.2697	0.3076	-0.0379
0.1582	0.1465	0.0117
0.2327	0.2810	-0.0483
0.1514	0.1493	0.0021

Table 4.8 Sample normalized desired and actual outputs with corresponding error

Desired output	Actual output	Error
252.62	251.76	0.86
137.49	136.29	1.2
308.49	307.99	0.5
281.71	281.51	0.2
183.23	183.43	-0.2
249.02	248.07	0.95
177.23	177.63	-0.4

Table 4.9 Sample unnormalized desired and actual outputs with corresponding error

There are different measures of performance of the network as cited in different literatures. Most of the literatures reported only the Mean Absolute Percentage Errors (MAPE); few also reported the standard deviation of the errors (Alfuhad et al., 1997). Measures based on squared errors are sometimes suggested, as they penalize large errors. Some papers reported that the Mean Absolute Errors could be useful (Papadakis et al., 1997).

Whatever the case, error measures are only intended as summaries for the error distribution. In this research undertaken, three measures of performance were conducted; MAPE (Mean Absolute Percentage Error), MAD (Mean Absolute Deviations) and MSD (Mean Squared Deviations).

MAD is the average amount by which the forecast deviates from the desired volume and is given by

$$MAD = \frac{\sum |V_d - V_f|}{N} \qquad (4.2)$$

Where V_f are the forecasted volumes, V_d are the desired volumes, and N is the number of test samples. The difference between V_f and V_d is the error term denoted by **e**.

Similarly, MAPE and MSD are given by,

$$MAPE = \frac{\sum \left| \frac{V_d - V_f}{V_d} \right|}{N} \times 100 \tag{4.3}$$

$$MSD = \frac{\sum (V_d - V_f)^2}{N} \tag{4.4}$$

The model is tested using the test dataset and its performance was evaluated by comparing the forecasted output with the desired output. The overall performance of the network assessed by MAD, MAPE and MSD for the forecast year is shown in the table 4.10. In addition, the forecast error distribution is presented in table 4.11 and figure 4.4.

Performance criterion	Network performance in forecasting
MAD	0.9789
MAPE	0.0426
MSD	1.0253

Table 4.10 Volume forecaster's performance

Absolute error range	Frequency	Cumulative frequency	Relative frequency
$0 \leq \lvert e \rvert \leq 1$	192	192	62.3%
$1 < \lvert e \rvert \leq 2$	75	267	86.7%
$2 < \lvert e \rvert \leq 3$	23	290	94.2%
$3 < \lvert e \rvert \leq 4$	11	301	97.8%
$4 < \lvert e \rvert \leq 5$	4	305	99.1%
$5 < \lvert e \rvert \leq 6$	2	307	99.7%
$6 < \lvert e \rvert \leq 7$	1	308	100%
Total	308		

Table 4.11 Frequency distribution of absolute forecast errors for volume forecast model

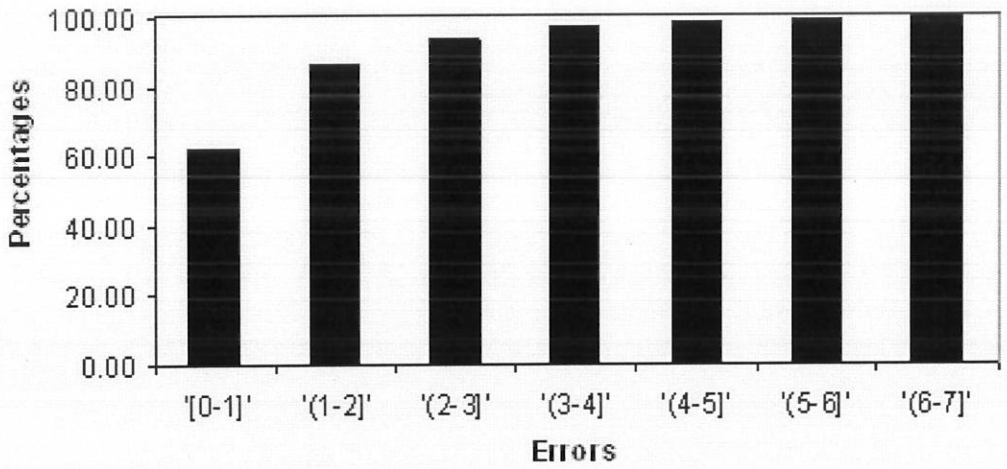


Figure 4.4 Histogram of absolute error distribution for the volume forecast model.

From table 4.11 and figure 4.4, it can be inferred that 86.7% of the absolute forecast errors are less than or equal to 2, while the absolute forecast errors with values greater than three constitutes only 5.8%. Hence, the ANN model is considered to be a good candidate for forecasting future values of tree volumes given properly and accurately measured historical data.

Chapter Five

5.1 Conclusions and recommendations

The application of ANN technique in forecasting is an advance in modeling of the relationships between the input and output. First, the ANN is able to recognize the relationships between the historical and target volumes. Furthermore, each input signal is spread over several hidden neurons where it is processed together with received signals. This parallel distributed processing gives neural networks the flexibility to model complex systems (Haykin, 1994). Second, the ANN does not pre-assume an explicit functional relationship to describe the observed input and output data, but rather it uses a layer-structured computational system to approximate the yet unknown functional relationship by constructing an input-output mapping from the observed tree volume database. Thus, the ANN avoids the possible mistakes of inadequate model specification encountered with multiple regression techniques (Wittink, 1988 cited in Haykin, 1994).

This paper reports on the research project that was carried out to assess the applicability of artificial neural network for forecasting the tree volume growth. The results obtained are significant to institutions dealing with forest researches in that, 1) it enables them to predict the growth of the tree against selling and harvesting plans; 2) it enables them to predict growth on a particular site to make rational decisions.

In this work, the neural network model generates the forecasted value of tree volume of *Cupressus lustanica* species at the age of 13. With the independent set of test data, forecast model predicted the volume, in which 62% of the cases have uncertainties within $\pm 1 cm^3$ and about 86.7% of the cases have uncertainties of $\pm 2 cm^3$.

Thus, an established volume-growth forecaster appears to operate like a "black-box" that takes in an input of historical volumes and gives an output of forecasted volume without knowing to understand the relationship between the input and the output values.

In this study on the way of developing the tree volume forecast model, the researcher has employed only historical values of tree volumes as an input to the neural network. Other variables that affect the tree volume growth directly or indirectly, i.e., climate, site quality and etc, are not included in building the model. Thus, as a further study, the researcher recommends to make use of these variables in developing the volume forecast model and see their effects on the performance of the model developed.

The performance of any neural network is promoted by presenting sufficient amount of training data to the network. In this study, few years of data are used in developing the tree volume forecast model (i.e., 8 years). Thus, the researcher's recommendation for further study in this area is to use many years' data to get good results out of the network employed.

As a last recommendation, it also possible to use other network types (i.e., Radial Basis Function Networks and etc) to check their performance in forecasting tree volume growth and make a comparison with the MLP.

In this study on the way of developing the tree volume forecast model, the researcher has employed only historical values of tree volumes as an input to the neural network. Other variables that affect the tree volume growth directly or indirectly, i.e., climate, site quality and etc, are not included in building the model. Thus, as a further study, the researcher recommends to make use of these variables in developing the volume forecast model and see their effects on the performance of the model developed.

The performance of any neural network is promoted by presenting sufficient amount of training data to the network. In this study, few years of data are used in developing the tree volume forecast model (i.e., 8 years). Thus, the researcher's recommendation for further study in this area is to use many years' data to get good results out of the network employed.

As a last recommendation, it also possible to use other network types (i.e., Radial Basis Function Networks and etc) to check their performance in forecasting tree volume growth and make a comparison with the MLP.

References

Alfuhad, A.S., et al.1997. Cascaded Artificial Neural Networks for short term load forecasting: in IEEE Trans. On power Systems, 12(4).

Anderson, D. and McNeil, G. 1992. Artificial Neural Networks Technology: Data and Analysis Center for Software, Rome Laboratory. Available at URL: http://www.dacs.dtlc.mil/techs/Neural/Neural.title.html

Avery, T. E. and Burkhart, H., E.1994. Forest Measurements, 4th ed.: McGraw Hill publishing.

Becker, S. 1991. Unsupervised Learning Procedures for Neural Networks: in International Journal of Neural System, 2(1).

Corne, S.A.; Carver, S.J.; Kunin, W.E.; Lennon, J.J., & S.Van Hees, W.W.2000.Using artificial neural network methods to predict forest characteristics in Southeast Alaska. Available at URL:
http://www.asgdc.state.ak.us/data/adnr/physical/coast/alsaka.html

Fauset, L.1994. Functional Neural Networks. New Jersey: Prentice-Hall.

Gorden, C.1998. Artificial neural network modeling of forest tree growth. University of Witwatersrand, South Africa, (MSc thesis)

Gorni, A.1997. The Application of Neural Networks in the Modeling of Plate Rolling Processes: in Journal of Modeling, 49(4).

Han, J. and Kamber, M. 2001. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publisher.

Hasenanuer, H.; Merkel, D. & Weignartener, M.2001. Estimating tree mortality of Norway spruce star with neural networks. Available at URL: http://www/rise.wien.ac.at/people/merks/papers/AEROL.pdf.

Haykin, S.1999.Neural networks: A comprehensive foundation. Ontario, Canada: Prentice Hall, Inc

Haykin, S.1994. Neural Network: A Comprehensive Foundation. Ontario, Canada: Prentice Hall, Inc

Husch, B; Miller, C., I.; and Beers, T., W.1982. Forest Mensuration. 3rd ed. New York: Wiley

Liu, C., Zhang, L., Davis, C.J., Solomon, D.S., Brann, T.B., and Caldwell, L.E.2003.

Comparison of Neural Networks and Statistical Methods in Classification of Ecological Habitats Using FIA Data: in Journal of American Forest Science, 49(4).

Ola, L.1986. Volume Tables for Ethiopia. Swedish University of Agricultural Sciences Department of Silviclture. UMEA.

Papadakis, S.E., et al, 1998. A novel approach to short term load forecasting using fussy neural networks: in IEEE Trans. On Power Systems: 13(2).

Philip, M.S. 1994. Measuring trees and forests. Dareselam, Tanzania: Cab International Publishing.

Sarls, W.S.1997. Neural Network FAQ, Periodic posting to the Usenet newsgroup comp. ai. Neural-net. Available at: URL:ftp//ftp.sas.com/pub/neural/FAQ.html

Senjyu, et al. 2002. One-hour a head load forecasting using Neural Network: in IEEETrans. On power System, 17(3)

Stat Soft . 2003. Neural Networks. Available at URL: http://www.StatSoftinc.com/textbook/steneunet.html

Stergions, C., & Siganos, D., 1996. Neural networks. Available at URL: http://www.doc.ic.uk/~nd/surprise- journal/vol4/cs11/report.html

Tesfaye, T.1996. A growth and yield model for *Cupressus lustanica* in Munessa Shashemene State Forest Project (MSSFP), southwest Ethiopia. Aberdeen: Aberdeen University, UK (PhD thesis).

Tracz, W. 2004. New Analytical Methods in Forestry: Integration of Expert Systems and Artificial Neural Networks with GIS: in Electronic Journal of Polish Agricultural Universities, 7(1).

Two Crows Corporation, 1999. Introduction to Data Mining and Knowledge Discovery. Available at URL: http://www.twocrows.com

Vanclay, J.K.1994. Modeling Forest Growth and Yield: Applications to mixed Tropical Forests. Royal Veterinary and Agricultural University, Denmark: CABI Publishing.

Zhang, Q.B.; Hebda, R.J.; Ahang, Q.J. & Alfaro, R.I. 2000. Modeling Tree-ring Growth Responses to Climatic Variables Using Artificial Neural Networks: in Journal of American Forest Science, 46(2).

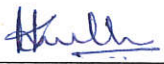
Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of material used for the thesis have been duly acknowledged.

Fikadu Gemechu Erba

March, 2006

The thesis has been submitted for examination with my approval as university advisor



Dr.Kumudha Raimond

March, 2006