Modeling of Neural networks using Lab VIEW

A Dissertation Submitted

In Partial Fulfillment of the Requirements for the Award of the Degree Of

MASTER OF ENGINEERING
IN

(CONTROL AND INSTRUMENTATION)

Submitted By

Ramanjaneyulu Gali *(16/C&I/07)*

University Roll No.: - 12244

Under the Esteemed Guidance Of

Dr. PARMOD KUMAR
(PROFFESOR & HEAD)

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI COLLEGE OF ENGINEERING

UNIVERSITY OF DELHI, DELHI-110042

2009

DELHI COLLEGE OF ENGINEERING

DELHI

Department of Electrical Engineering

CERTFICATE

This is to certify that **Mr. Ramanjaneyulu Gali** student of final semester **M.E. (C&I), Electrical Engineering Department, Delhi College of Engineering**, during the session 2008-2009 has successfully completed the project work on **Modeling of neural networks using Lab VIEW** and has submitted a satisfactory report in partial fulfillment for the award of the degree of **Master Of Engineering** (**Control and Instrumentation**) in **Electrical Engineering Department.**

( )

Prof. Parmod Kumar

Project Guide

HOD, EED

Delhi College of Engineering,

Delhi

__ Acknowledgement__

I would like to extend my sincere gratitude and thanks to my guide **Dr. Parmod Kumar**, Professor and Head of Electrical Engineering Department, Delhi College of Engineering, Delhi for his valuable guidance, constant encouragement and helpful discussions throughout the course of this work. He initiated the founding ideas behind this thesis and provided insight and valuable information into many of the problems that were encountered. His positive attitude throughout the duration of this thesis always put things into perspective whenever difficulties arose. This thesis would not have been possible without him.

I would also like to thank respected Aaanchal maheswari and Payal Bhuptani madam from National Instruments they are provided free training in Lab VIEW and helped in all ways whenever I am facing difficulty in Lab VIEW and all my friends mainly Mr. Suresh , Harish vakalapudi and santosh for immense help, advice and cooperation to complete this work.

I am extremely grateful to my parents for their encouragement, support and dedication which have helped me in great way to complete this work. Without their blessings, this work would not have been possible.

Lastly, I thank Almighty GOD for his countless blessings.

Ramanjaneyulu Gali
M.E(C&I)

Abstract

An Artificial Neural network (ANN) is a collection of neurons in a particular arrangement or configuration. It is a parallel processor that can compute or estimate any function. Basically in an ANN, knowledge is stored in memory as experience and is available for use at a future time. The weights associated with the output of each individual neuron represent the memory which stores the knowledge. These weights are also called inter-neuron connection strengths. Each neuron can function locally by itself, but when many neurons act together they can participate in approximating some function. The knowledge that is being stored will also be referred to as “numeric data”, which is transferred between neurons through weights. ANNs learn through training. There are various training algorithms for different types of ANN, based on the specific application. Based on the training, the weights associated with each neuron adjust themselves. By training, it is meant that a set of inputs is presented repeatedly to the ANN and the weights adjusted so that the weights reach an optimum value, where optimum means that weights either tend to minimize or maximize. An ANN is trained repeatedly and at one point it reaches a stage where it has ‘learned’ a particular desired function. With proper training it can generalize, so that even new data, which was not part of the training data, will yield the desired output.

Artificial Neural Networks have gained wide-spread popularity over the last few decades. Their application is considered as a substitute for many classical techniques that have been used for many years. Some of the most common applications of an ANN would be pattern recognition, plant modeling, plant control, and image compression. A large variety of neural network architectures have been developed. These ANN models have the capability to use more than one learning algorithm for training purposes. Different aspects of ANN such as efficiency, speed, accuracy, dependability and the like have been studied extensively in the past. Many approaches have been explored to improve the performance of neural nets. In this thesis, a new approach is proposed to build neural net architectures. Lab VIEW is graphical programming software developed by National Instruments. Using Lab VIEW, ready-made Virtual Instruments (VI) can be developed for various applications. Lab VIEW is basically an Application Development Environment (ADE) for building user friendly applications. This thesis concentrates on a Lab VIEW approach to build various neural net structures.

In this thesis, a new approach has been applied to build neural nets. Lab VIEW, powerful graphical programming software developed by National Instruments, which has, so far been successfully used for data acquisition and control, has been used here for building neural nets. Both supervised and the unsupervised neural nets have been successfully developed in this thesis using Lab VIEW, and it has been proved that Lab VIEW is a very powerful software tool for building neural nets. Neural nets are parallel processors. They have data flowing in parallel lines simultaneously. Lab VIEW has the unique ability to develop data flow diagrams that are highly parallel in structure. So Lab VIEW seems to be a very effective approach for building neural nets.

Table of Contents

Abstract		**iv**
List of Figures		**ix-xii**
CHAPTER 1	**Introduction**	**1-4**

1.1 Introduction

1.2 Objective of Study

1.3 Motivation

1.4 Features of neurons

1.5 Features of Lab VIEW

1.6 Dissection of Thesis

1.7 Conclusions

CHAPTER 2	**Literature Review**	**5-13**

2.1 Introduction
2.2 Neural Networks
2.3 Different types of neural network architectures
2.4 Lab VIEW
2.5 Some important features of Lab VIEW
2.6 Conclusion

CHAPTER 3 Neural Net Modeling 14-32

3.1 Introduction

3.2 Definition of neural network

3.3 Neural network construction and working principle

3.4 Characteristics of artificial neural networks

3.5 Electrical representation of neural network

3.6 Problem suited to neural networks

3.7 Problems not suited to neural networks

3.8 Structure of neural networks

3.8.1 Neuron

3.8.2 Neuron connection weights

3.8.3 Neuron layers

3.9 Main Features of neural networks

3.10 Mathematical model representation of neural network

3.11 Different types of activation functions

3.12 Classification of neural networks

3.13 Conclusion

CHAPTER 4 Introductions to Lab VIEW 32-50

4.1 Introduction

4.2 Virtual instrumentation

4.3 Virtual instruments vs. Traditional instruments

4.4 Lab VIEW

4.5 Features of Lab VIEW

4.6 Present draw backs of Lab VIEW

4.7 Main parts of Lab VIEW

4.7.1 Front panel window

4.7.2 Block diagram window

4.8 Controls, functions and tools palette

4.9 Structures

4.9.1 Different types of structures

4.10 Shift registers

4.11 Arrays

4.12 Clusters

4.13 Conclusion

CHAPTER 5 Fusion of neuron with Lab VIEW 50-78

5.1 Lab VIEW implementation of a neuron

5.1.1 Using Sigmoid function

5.1.2 Sigmoid function

5.1.3 Using Gaussian function

5.1.4 Gaussian function

5.2 Multi input-single output neuron response

5.2.1 Using Sigmoid function

5.2.2 Using Gaussian function

5.3 New weight of the neuron using different types of input

5.3.1 Sigmoid function

5.3.2 Tangential hyperbola

5.3.3 Linear input

5.4 Neuron response using different types of inputs

5.3.1 Sigmoid function

5.3.2 Tangential hyperbola

5.3.3 Linear input

5.5 Gaussian function implementation using radial basis input

5.6 Implementation of learning algorithms using lab view

5.6.1 Supervised learning algorithms

5.6.2 Back propagation algorithm

5.6.3 Lab VIEW implementation of back propagation algorithm

5.6.4 Lab view implementation of error function

5.7 Unsupervised learning algorithms

5.7.1 Unsupervised ANN

5.7.2 Unsupervised clustering algorithms

5.7.3 K-means clustering algorithm

5.7.4 SOM algorithm

5.7.8 Implementation of feed forward ANN using logic gates

Conclusions **79-80**

Bibliography **81-82**

Curriculum vitae **83**

__List of Figures__

CHAPTER 2

Figure 2.1: Structure of the feed forward network

Figure 2.2: Structure of the feedback network

Figure 2.3: Structure of lateral network

CHAPTER 3

Figure 3.1: Schematic diagram of a biological neuron

Figure 3.2: Electrical membrane model

Figure 3.3: Neural network layers

Figure 3.4: Mathematical model of an M-P neuron

Figure 3.5: response of sigmoid function

Figure 3.6: Tangential hyperbola response

Figure 3.7: Linear function response

Figure 3.8: Single – input neuron

Figure 3.9: Multiple- Input neuron

Figure 3.10: Single layer feed forward ANN

Figure 3.11: Multi layer feed forward ANN

Figure 3.12: Structure of self organizing map (SOM)

CHAPTER 4

Figure 4.1: Architecture of a virtual instrument

Figure 4.2: VI front panel

Figure 4.3: Controls and indicators

Figure 4.4: Boolean controls and indicators

Figure 4.5: VI block diagram

Fig 4.6 1. Nodes, 2. Indicator Terminals, 3.Wires, 4. Control Terminals

Figure 4.7: Icon and connector pane

Figure 4.8a: controls palette

Figure 4.8b: Functions palette

Figure 4.8c: Tools palette

Figure 4.9: For loop

Figure 4.10: Block diagram example to illustrate using “for” loop

Figure 4.11: While loop

CHAPTER 5

Figure 5.1: Block diagram implementation of a neuron

Figure 5.2: Front panel implementation of a neuron

Figure 5.3: BD implementation of sigmoid function

Figure 5.4: FF implementation of sigmoid function

Figure 5.5: BD implementation of a neuron using Gaussian function

Figure 5.6: FF implementation of a neuron using Gaussian function

Figure 5.7: BD implementation of Gaussian function

Figure 5.8: FF implementation of Gaussian function

Figure 5.9: BD implementation of a neuron using multi input-single output using sigmoid function

Figure 5.10: FF implementation of a neuron using multi input-single output using sigmoid function

Figure 5.11: BD implementation of a neuron using multi input-single output using Gaussian function

Figure 5.12: FF implementation of a neuron using multi input-single output using Gaussian function

Figure 5.13: BD implementation for finding new weight of the neuron using sigmoid function

Figure 5.14: FF implementation for finding new weight of the neuron using sigmoid function

Figure 5.15: BD implementation for finding new weight of the neuron using tangential hyperbola function

Figure 5.16: FF implementation for finding new weight of the neuron using tangential hyperbola function

Figure 5.17: BD implementation for finding new weight of the neuron using linear function

Figure 5.18: FF implementation for finding new weight of the neuron using linear function

Figure 5.19: BD implementation of neuron output response using sigmoid function

Figure 5.20: FF implementation of neuron output response using sigmoid function

Figure 5.21: BD implementation of neuron output response using tangential function

Figure 5.22: FF implementation of neuron output response using tangential function

Figure 5.23: BD implementation of neuron output response using Linear input

Figure 5.24: FF implementation of neuron output response using Linear input

Figure 5.25: BD of Gaussian function implementation using radial basis input

Figure 5.26: FF of Gaussian function implementation using radial basis input

Figure 5.27 Block diagram of learning with teacher

Figure 5.28: BD implementation of Back- Propagation algorithm

Figure 5.29: FF implementation of Back- propagation algorithm

Figure 5.30: BD implementation of k-means clustering algorithm

Figure 5.31: FF implementation of K-means clustering algorithm

Figure 5.32: BD implementation of SOM algorithm

Figure 5.33: BD implementation of Feed Forward ANN using logic gates

Figure 5.34: FF implementation of Feed Forward ANN using logic gates