Development of Computational Intelligence Systems for Parameter Estimation and Event Identification in Fast Breeder Reactors

By

Subhra Rani Patra

(Enrollment No: ENGG02200704007)

Indira Gandhi Center for Atomic Research

Kalpakkam-603102, Tamil Nadu, India

A thesis submitted to the

Board of Studies in Engineering Sciences

In partial fulfillment of requirements

For the degree of

DOCTOR OF PHILOSOPHY

of

HOMI BHABHA NATIONAL INSTITUTE

July, 2012

Development of Computational Intelligence Systems for Parameter Estimation and Event Identification in Fast Breeder Reactors

By

Subhra Rani Patra

(Enrollment No: ENGG02200704007)

Indira Gandhi Center for Atomic Research

Kalpakkam-603102, Tamil Nadu, India

A thesis submitted to the

Board of Studies in Engineering Sciences

In partial fulfillment of requirements

For the degree of

DOCTOR OF PHILOSOPHY

of

HOMI BHABHA NATIONAL INSTITUTE

Research Supervisor: Prof. M. Sai Baba

Associate Director, Resources Management Group

Indira Gandhi Centre for Atomic Research

Department of Atomic Energy, Kalpakkam, India

HOMI BHABHA NATIONAL INSTITUTE

Recommendations of the Viva Voce Board

As members of the Viva Voce Board, we certify that we have read the dissertation prepared by Ms. Subhra Rani Patra entitled "**Development of Computational Intelligence Systems for parameter estimation and event identification in Fast Breeder Reactors** " and recommend that it may be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date:

Chairman Prof. T. Jayakumar

Date:

Convener Prof. M. Sai Baba

Date:

External Examiner

Date:

Member1 Prof. G. Amarendra

Date:

Member2 Prof. K. Veluswamy

Dated:

Technology Advisor Shri S. A. V. Satya Murty

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertations to HBNI.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it may be accepted as fulfilling the dissertation requirement.

Date: Guide: Prof. M. Sai Baba

Place:

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in library to make available to borrowers under the rules of HBNI.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the competent authority of HBNI when in his or her judgment the proposed use of the material is in the interest of scholarship. In all other instances, permission must be obtained from the author.

Subhra Rani Patra

Kalpakkam
July, 2012

DECLARATION

I, hereby declare that the thesis entitled "**Development of Computational Intelligence Systems for Parameter Estimation and Event Identification in Fast Breeder Reactors**" submitted to Homi Bhabha National Institute (HBNI), Mumbai, India, for the award of Doctor of Philosophy in Engineering Science, is the record of work carried out by me during the period from July 2007 to July 2012 under the guidance of Dr. M Sai Baba, Associate Director, Resources Management Group" and Shri S.A.V. Satya Murty, Director, Electronics Instrumentation and Radiological Safety Group, Indira Gandhi Center for Atomic Research, Kalpakkam. The work is original and has not been submitted earlier as a whole or in part of degree/diploma at this or any other Institute/University of higher learning.

Subhra Rani Patra

Kalpakkam
July, 2012

Epigraph

"A journey of a thousand miles begins with a single step"…

~ Lao-tzu

DEDICATION

I dedicate this thesis

to

my family

ACKNOWLEDGEMENTS

First of all I would like to thank my research supervisor, Dr. M. Sai Baba, Associate Director, Resource Management Group and my technology advisor, Shri S.A.V. Satya Murty, Director, Electronics Instrumentation and Radiological Safety Group, IGCAR not only for their constant support, advice, encouragement and patience but also for allowing me to interact with people from various research fields.

I would like to express my sincere gratitude to Mrs. S. Rajeswari and Mr. R. Jehadeesan for their guidance in many aspects including literature survey, experimental work, preparation of publications, completion of thesis and detailed discussions regarding neural networks.

I am extremely thankful to my doctoral committee Chairman Prof. T. Jayakumar and committee members Prof. G. Amarendra and Prof. K. Veluswamy for their support and encouragement during my research work.

My sincere thanks go to Shri S. C. Chetal, Director, IGCAR and Dr. Baldev Raj, former Director, IGCAR for providing excellent environment to carry out research work.

I would like to thank Mrs. T. Jayanthi for her unceasing and invaluable support. I would like to thank Mr. M. Jayalal for his detailed feedback and comments on my work and his useful advice.

My sincere thanks are due to Mrs. H. Seetha for giving me the basic idea about Fast Breeder Reactor Systems and also for her guidance. My special thanks to Mr. A. Samba Sivarao for helping me learn Prolog language.

I am thankful to Ms. Rashmi Nawlakha for her moral support and going through my thesis and giving me valuable suggestions during my entire PhD work. My thanks are due to Dr. S. Srinivasan for his valuable feedback given to me during my research work.

My sincere thanks also go to Prasanth, Jasmine, Raju, Baghyalakshmi, Sandhyarani, Vijayalakshmi, Vinita, Haripriya, Priyanka, Molly, Saptarshi Ghosh, Pradeep, Krishnan, Manas, Sujith, Salabh and all members of computer division for their attention, support and patience during this period.

I am grateful for the financial support provided by research fellowship scheme from Indira Gandhi Centre for Atomic Research for the duration of the work.

Special thanks are due to the following people who helped me in various ways to make this work a reality: Madhusmita, Ruth Nithila, Kuheli, Rajini, Sudeepta, Maneesha, Pravati, Bishnu, Prasanna, Sulata Sahu, Girish Padhy, Alok Rout, Sanatana Maharana, Pradeep Samantroy, Bubathi, Arun and many more. I would like to thank all my colleagues in the enclave for their love, caring during my stay at Kalpakkam.

Finally I would like to thank my mother Smt. Bidyut Prabha Patra, elder brother Sri Ajit Kumar Patra, elder sisters Ms. Snigdha Rani Patra and Ms. Smita Rani Patra who have provided me the ideal environment to grow up in and from whom I have got the moral support to carry out the task and make my study carrier smooth. My thanks are due to my cousin Jully for her constant moral support. Thank you everyone.

Subhra Rani Patra

TABLE OF CONTENTS

CHAPTER 4
EVENT IDENTIFICATION IN FAST BREEDER REACTOR SUBSYSTEMS

CHAPTER 5
OPTIMIZATION OF FEED FORWARD NEURAL NETWORK USING GENETIC ALGORITHM

Abstract

This thesis summarizes a detailed study of the implementation of computational intelligence systems in nuclear reactors as intelligent monitors. It provides elaborated description of the implementation of computationally intelligent models for parameter estimation in Intermediate Heat Exchanger of Fast Breeder Test Reactor (FBTR) and Neutronics System of Prototype Fast Breeder Reactor (PFBR). The parameters predicted are: temperature of sodium in intermediate heat exchanger of FBTR and reactor power in PFBR. It briefly covers the event identification in Neutronics System and Primary Sodium Circuit System of Prototype Fast Breeder Reactor.

A novel neural network model is proposed to estimate the value of temperature parameters in Fast Breeder Reactor subsystems. Supervised back propagation algorithm is proposed, fine-tuned and from the results obtained, it is shown that this algorithm shows faster convergence compared to conventional theoretical models developed earlier for fast reactors. The significant parameters used for prediction of the temperature of sodium are primary inlet temperature, primary flow, secondary inlet temperature and secondary flow. The desired output parameters to be predicted are primary outlet temperature and secondary outlet temperature. Input data for intermediate heat exchanger has been generated from the Quadratic Upstream Interpolation for Convective Kinetics (QUICK) code. It took 10 minutes time to generate the data set using QUICK code. After fine tuning, the neural network is trained for 10^5 iterations using back propagation algorithm and the achieved error convergence is of the order of 10^{-4}. The network is also trained with radial basis function algorithm. A comparative study of back propagation algorithm and radial basis function algorithm is carried out to predict the

temperature parameters of intermediate heat exchanger of FBTR. The training results show that radial basis function algorithm is much faster compared to the standard back propagation algorithm as the network got converged to mean square error 8.9×10^{-6} *in* 10^4 *iterations itself. To train the network using radial basis function it took 3 minutes on a typical personal computer system (2.66 GHz Intel Core 2 Duo processor). Once trained, the network gives the test outputs in few milliseconds time.*

Reactor power estimation has also been carried out using back propagation algorithm using multilayer perceptron model. The input parameters to the neural networks are position of nine control and safety rods and the output parameter is reactor power. To predict the reactor power output for one sample the time taken is 23.34 seconds in the PFBR simulator manually. While, artificial neural network is able to obtain the output in few milliseconds giving accurate predictions. Back propagation algorithm along with its variants namely standard back propagation, back propagation with momentum in pattern mode, back propagation with momentum in batch mode, quick propagation and resilient back propagation algorithm are also applied for process modeling of neutronics system. Among those algorithms resilient algorithm has converged faster with less number of epochs and an error convergence of 4.9×10^{-7} *and produces the target output which is well agreement with the desired output.*

Neural network model has also been implemented for identification of events or unsought occurrence of plant conditions which affect the safe operation of plant. In nuclear reactor, thousands of alarms are generated within a fraction of seconds. So, the operator has to take immediate as well as appropriate action which can prevent occurrence of any abnormal plant conditions. Neural network is a tool that helps in *predicting plant behavior in a fast and reliable manner. The neutronics system and primary sodium circuit related events of PFBR have been monitored using neural network algorithms. The event related input data have been generated from in-house developed thermal hydraulics code and has been validated as per the event analysis reports of PFBR. To detect uncontrolled withdrawal of control and safety rod, it took 3.8 seconds and to detect primary sodium pump trip it took 0.43 seconds using simulator. Whereas, neural network predicts event in milliseconds range. Four different events are integrated in a single neural network and back propagation algorithm and its variants are implemented. The events are: uncontrolled withdrawal of control and safety rod and primary sodium circuit related events, primary sodium pump trip, and primary sodium pump seizure. The significant parameters for these four events are reactivity (ρ), linear power (Lin P), central subassembly outlet temperature (θ_{CSAM}), increase in the central subassembly temperature ($\Delta\theta_{CSA}$), increase in the mean core temperature ($\Delta\theta_{M}$), power to flow ratio (P/Q), pump speed (N_p) and these are used to represent as input nodes to the neural network. The network is trained using BIKAS (Bhabha Atomic Research Centre – Indian Institute of Technology Kanpur-Artificial Neural Networks – Simulator) with variants of back propagation algorithms. Among these algorithms, resilient back propagation algorithm shows least mean square error convergence of 4.29×10^{-4} providing better performance compared to other algorithms. From the obtained test results, it is observed that the neural network could identify the events successfully.*

A hybrid genetic algorithm based neural network model has been developed for sodium temperature parameter estimation of intermediate heat exchanger of FBTR and the results are compared with standard back propagation algorithm. From the results *obtained, it could be observed that genetic algorithm based neural network has faster convergence with less time of computation in comparison with standard back propagation algorithm.*

From the results obtained in this thesis, it is seen that artificial neural network can be used to predict the parameters and detect the anomalies of subsystems accurately and consistently, which will be an aid to operators handling non-linear complex systems in nuclear reactors.

List of Figures

List of Tables

List of Abbreviations

Abbreviations	Explanations
CI	Computational Intelligence
ANN	Artificial Neural Network
AI	Artificial Intelligence
BI	Biological Intelligence
NPP	Nuclear power plant
FBTR	Fast Breeder Test Reactor
PFBR	Prototype Fast Breeder Reactor
PHWR	Pressurized Heavy Water Reactor
SG	Steam Generator
IHX	Intermediate Heat Exchanger
SCRAM	Safety Control Rod Accelerated Movement
MOX	Mixed Oxide
OGDHR	Operation Grade Decay Heat Removal System
SGDHR	Safety Grade Decay Heat Removal
BNN	Biological Neural Network
MLP	Multi Layer Perceptron
MSE	Mean Square Error
CDPS	Central Data Processing System
SCS	Safety Critical System
SRS	Safety Related System
NSS	Non-Safety System
DYANAP	Dynamic Analysis-P
CSR	Control and Safety Rod
QUICK	Quadratic Upstream Interpolation for Convective Kinetics code
NHB	Nodal Heat Balance
RBF	Radial Basis Function

BPN	Back Propagation Network
SDS	Shutdown System
CSRDM	Control and Safety Rods Drive Mechanism
DSRDM	Diverse Safety Rods Drive Mechanism
MIMO	Multi-Input Multi-Output
DBE	Design Basis Event
PSP	Primary Sodium Pump
EDS	Event Detection System

Chapter 1

Introduction

This chapter gives an introduction to the study of computational intelligence and its applications in the operation and control of nuclear reactors. It gives a brief description about the Fast Breeder Test Reactor, Prototype Fast Breeder Reactor and the application of computational intelligence in the operation of Fast Breeder Reactors.

1.1. Introduction

Nuclear power plant is a very complex arrangement of machinery consisting of a large number of control and support systems exhibiting nonlinear behavior. Accurate assessment of the operational characteristics in a nuclear power plant is of paramount importance. The plant must be designed and operated within a quantitative risk-informed approach, integrating both deterministic and probabilistic techniques, where the outputs of both the approaches complement with each other. There should be a balance between deterministic approaches and probabilistic analyses, in order to achieve an integrated decision making process that serves in an optimal fashion to ensure nuclear reactor safety. Employing conventional physical model is often time consuming. Hence, it is preferable to go for empirical modeling for gaining insights into the overall processing behavior of the complex systems in comparison with the conventional physical models based on rigorous mathematical equations.

In real time, it is possible to implement intelligent systems in the form of artificial neural network, data mining, and expert system etc., for modeling the nuclear power plant. Artificial neural network model, being nonlinear, data driven and having black box approach, is a powerful tool for identification of relevant physical parameters. A less understood system with large input and output datasets can be modeled using artificial neural network without having much knowledge of its internals [1.1, 1.2]. The present study is primarily concerned with the development of artificial neural network models for parameter estimation and fault detection of the systems in a nuclear reactor.

The major contributions of the work carried out in the thesis is summarized and given below.

- The role of computational intelligence and applications of neural network in the operation of a nuclear reactor are presented.
- The generalization ability and the optimization methods of neural network have been studied.
- Advanced neural network training algorithms for parameter estimation and event identification have been studied and applied for some of the subsystems of Fast Breeder Reactor.
- Intermediate heat exchanger of Fast Breeder Test Reactor and neutronics system and primary sodium circuit of Prototype Fast Breeder Reactor are the subsystems studied.
- A unified empirical model has been developed for optimizing different aspects of artificial neural networks in parameter estimation and event identification. This model is implemented using various forms of hardware and software.
- It is seen from the results obtained, that artificial neural network could be used to estimate and identify the events and parametric values of the nuclear reactor accurately and consistently.

1.2. Overview of the Thesis

An overview of this thesis is presented in this section.

Chapter 1 begins with a brief introduction to nuclear power plant, its complexity, characteristics and the methods adopted to achieve its safe operation. A detailed explanation of the concept of computational intelligence and its hierarchy is presented. The applications of computational intelligence in nuclear power plant operation are also described. The Fast Breeder Test Reactor (FBTR) and Prototype Fast Breeder Reactor (PFBR) subsystems are described. The scope of the present study namely, parameter estimation and event identification in Fast Breeder Reactor have been depicted.

Chapter 2 briefly covers the basics of artificial neural network (ANN), its advantages, disadvantages etc and its comparison with biological neural network (BNN). Different architectures of artificial neural network are presented. Various learning procedures and activation functions used in artificial neural network are outlined. Various neural network algorithms are also described. A literature survey on the applications of artificial neural network in various nuclear reactor studies has been carried out and presented.

Chapter 3 starts with a brief description of intermediate heat exchanger of Fast Breeder Reactor. It deals with the estimation of temperature parameters in intermediate heat exchanger of FBTR using standard back propagation algorithm for steady state condition. Further, starting from steady state to transient state, the network is trained with standard back propagation algorithm and radial basis function algorithm and results obtained are compared. Reactor power estimation of neutronics system of PFBR has been carried out using standard back propagation algorithm, and compared with its variants.

Chapter 4 outlines the event identification of nuclear reactor subsystems using neural network algorithms. The events associated with neutronics system and primary sodium circuit systems of PFBR are described in detail. The neural network is trained with standard back propagation algorithm to identify PFBR related events. The different events considered are: uncontrolled withdrawal of control and safety rod, primary sodium pump trip, primary sodium pump seizure, primary pump rupture. An Event Detection System (EDS) has been developed for identifying these four events in PFBR subsystems at the earliest time of occurrence using back propagation algorithm and its variants and the results are compared.

Chapter 5 Genetic algorithm has been considered as one of the potential candidates for optimization of weight parameters of neural network. A hybrid genetic algorithm based neural network model has been developed for parameter estimation of intermediate heat exchanger of FBTR. The training of neural network has been accomplished using the weight optimization method of genetic algorithm based neural network to predict the temperature parameters of intermediate heat exchanger of FBTR.

Chapter 6 summarizes the work carried out in this thesis and the relevant conclusions drawn based on the study. The chapter ends with suggestions for future work in this field.

1.3. Computational Intelligence

Computational intelligence (CI) being a relatively new area, its identity and definition are still a subject of debate. According to one of the definition, it is an area of applied research involving the study of adaptive mechanisms to enable or facilitate intelligent behavior in complex and changing environments [1.3]. It can be described as the science of “Synthetic Psychology” or “Experimental Philosophy” or “Computational Epistemology”-epistemology being the study of knowledge [1.4]. A Computationally intelligent system deals with numerical (low-level) data, has pattern recognition component and it begins to exhibit computational adaptability, computational fault tolerance and speed approaching human-like turnaround, error rates that approximates human performance [1.6].

The main building blocks of computational intelligence are artificial neural network, fuzzy logic, evolutionary computing, genetic algorithm and artificial life. The hierarchy of computational intelligence is given in Fig. 1.1. Computational intelligence being the parent node, represents a broader discipline and is subdivided into child nodes such as granular computing, neuro-computing, evolutionary computing and artificial life [1.5]. These child nodes are again subdivided further more as shown in Fig. 1.1.

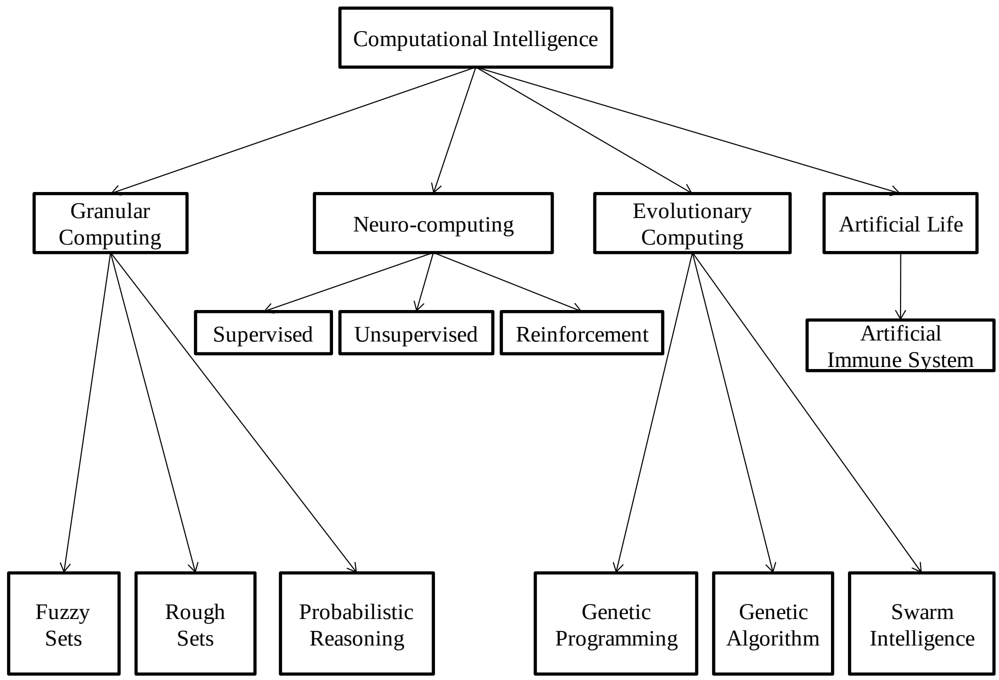


Fig. 1. 1 A fragment of computational intelligence family tree

Computational intelligence differs from the traditional artificial intelligence (AI) in few aspects, as it uses sub-symbolic knowledge i.e., numerical knowledge representation and processing, where as classical artificial intelligence uses symbolic approaches. Symbolic approach of artificial intelligence and distributed approach of computational intelligence are clearly distinguishable and can be explained as follows: in a symbolic representation, the knowledge can be described in symbols (e.g. a concept in a semantic net) in which each symbol has a particular meaning. Whereas, in distributed or sub symbolic representations, a meaning or specific part of the knowledge cannot be clearly located. The knowledge is represented in the whole state of the system. Neither humans nor any other animal thinks in symbols. Language is purely symbolic. But a person does not think in language. When he thinks of a concept which can be described by a symbol (a word of the language), he does not think of the symbol itself, but of all the associations that he has, when he hears the specific word. These associations can often also be described by symbols, but they are not symbols themselves. They are made of many recollections of past internal states of the system, which is produced by sensory inputs combined with the previous states of the system. In simpler terms it can be stated that, a person thinks in the combined recollections of many images, sounds, smells and other past sensory experiences. Hence the natural or sub-symbolic approach has a more promising future even though it is easier for us to build symbolic systems which are therefore still often the best choice.

Artificial intelligence techniques follow top-down approach i.e. the structure of a given problem (environment, domain context) is analyzed beforehand and the construction of the intelligent system is based upon this structure. Computational intelligence techniques follow bottom-up approach where order and structure emerge from an unstructured beginning rather than being imposed from above [1.7]. A comparison of artificial intelligence with computational intelligence, which comes under machine intelligence, is given in Fig. 1.2. Artificial intelligence is based on hard computing whereas computational intelligence is based on soft computing. Hard computing is oriented towards the analysis and design of physical process and systems.

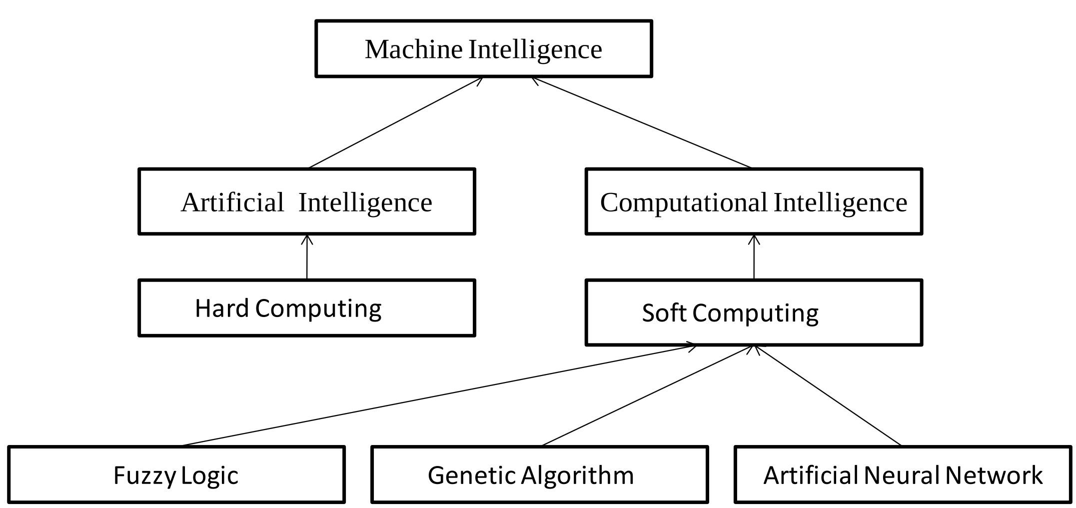


Fig. 1.2 Artificial Intelligence vs. Computational Intelligence

It has characteristics like precision, formality and categoricity. It is based on binary logic, crisp systems, numerical analysis, probability theory, differential equations, functional analysis, mathematical programming, approximation theory and crisp software. Soft computing is oriented towards the analysis and design of intelligent systems. It is based on fuzzy logic, artificial neural networks algorithms, genetic algorithms, chaos theory and a part of machine learning. While in hard computing, impression and uncertainty are undesirable properties, in soft computing the tolerance for imprecision and uncertainty is exploited to achieve an acceptable solution at a low cost, tractability and high machine intelligence quotient.

Artificial intelligence is defined as “the branch of science which deals with helping machines to find solutions to complex problems in a more human-like fashion”[1.8]. The discipline of human engineered systems exhibiting intelligence has been described in three levels of complexity by Bezdeck[1.9]. Level A stands for artificial or symbolic, Level B for biological or organic, Level C stands for computational or numeric systems. This is represented in Fig. 1.3.

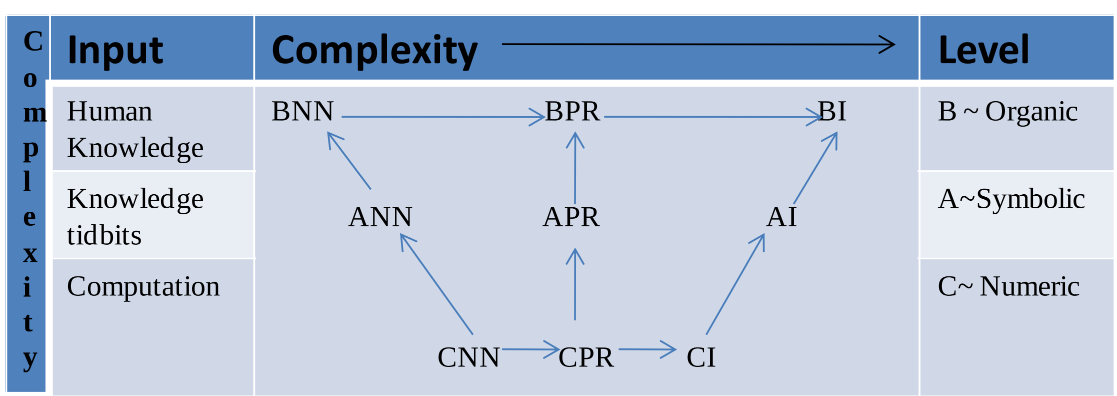


Fig. 1.3 Relationship among components of intelligent systems

According to Bezdek, computational intelligence would be a subset of artificial intelligence which in turn is a subset of biological intelligence [1.6]. Fogel expressed a different view that the central focus in traditional artificial intelligence research has been on emulating human behavior by extracting rules and knowledge from human experts [1.6]. The vast majority of artificial intelligence programs have nothing to do with learning. In contrast, computational intelligence techniques model natural processes or end products associated with intelligent behavior, either at the level of neuronal activity and function, human behavior or evolutionary learning in terms of the adaptive behavior. Hence, the objective of this research work is to exploit their characteristics for their application to the nuclear reactor systems, without being too specific about the definitions of artificial intelligence and computational intelligence.

1.4. Application of Computational Intelligence in Condition Monitoring of Nuclear Reactor

The next generation nuclear power plants (NPP) are being planned to have their licenses up to 60 years. The development of condition monitoring and diagnostic techniques is in constant transition and evolving towards its best to achieve plant operating objectives, meeting the future needs and regulatory requirements. Monitoring the plant as and when required is termed as condition monitoring. It helps in monitoring the plant characteristics using the changes and trends of monitored signals in order to predict the need for maintenance before occurrence of any anomaly, avoiding failures and minimizing the downtime. It is an optimal method in comparison with time based maintenance [1.9]. Computational intelligence is one of the promising options in solving many intricate problems associated with maintenance of the nuclear reactor. However, it is a challenging task due to the requirement of adhering to strict nuclear safety regulation.

Computational intelligence is implemented in diagnosing specific abnormal conditions, identifying non linear dynamics and transients, validating the signals, monitoring plant parameters etc. Some of the computational intelligence techniques implemented in nuclear reactors are mentioned below.

Noise Analysis: It is a computational intelligence technique that determines the state of the system, the monitoring of loose parts, acoustic leak monitoring and so on [1.10].

Online Monitoring and Verification of Sensor Calibration: Some of the online monitoring activities carried out for transient identification, alarm management and general equipment monitoring using various computational intelligence techniques are presented [1.9, 1.10]. These are multivariate state estimation techniques, auto-associative neural network, probabilistic neural network etc.

Nuclear Power Plant Efficiency Improvement: Several models are developed using computational intelligence techniques and are available to monitor the thermodynamic performance of a plant and to help in diagnosing the operating problems and the effects of changes in equipment and operating parameters, which could save power plants considerable amount of money annually [1.10].

Autonomous Anticipatory Control and Intelligent Agents: In the new generation of reactors (Gen IV), it is envisaged to introduce multi-intelligent agent system to aid semi-autonomous operation, thereby reducing the requirement of human resources. The multi-intelligent agent system consists of intelligent controllers that contain perception module performing fuzzy inference, information integration and response module performing operational assessment, planning etc [1.10-1.15].

The three stage nuclear power programme has been described in next section and FBTR and PFBR are briefly described in the following section.

1.5. Three stage Nuclear Power Programme

For a large country like India, energy security is an important and inevitable need from economic and strategic point of view. The energy demand is being met largely through utilization of coal and hydro resources. Taking into consideration the CO_2 emission and environmental impacts of using fossil fuels, nuclear resources have come a long way to meet the energy needs for many years [1.16]. The nuclear energy options are being pursued as a source of energy for long term energy security in a developing country like India. A strong indigenous research and development infrastructure including trained scientific and engineering manpower needs to be developed and deployed for utilizing all the nuclear energy sources optimally. The nuclear power programme is being implemented in three stages in India (Fig. 1.4), considering the limited uranium resources. Natural uranium fuelled Pressurized Heavy Water Reactors (PHWR) are being operated in the first stage. In the second stage, plutonium generated from PHWR and the depleted uranium are being utilized in the Fast Breeder Reactors and thorium is being utilized to generate U^{233}. This in turn will launch large scale thorium-uranium fuel cycle in the third stage [1.17, 1.18].

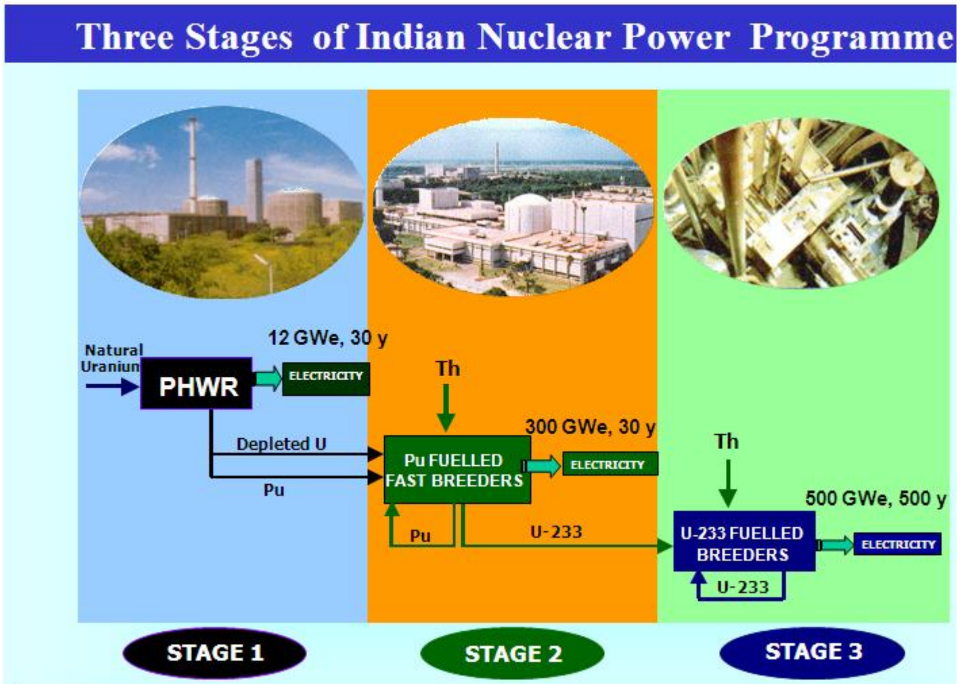


Fig. 1.4 Three Stages of Nuclear Power Programme

In India, Fast Breeder Test Reactor is in operation at Indira Gandhi Centre for Atomic Research, Kalpakkam. Prototype Fast Breeder Reactor is designed by IGCAR and is under construction at Kalpakkam [1.19]. Case studies conducted in this thesis are based upon the modeling of systems of these two reactors. A brief description of these reactors is given below.

1.5.1. Fast Breeder Test Reactor

Fast Breeder Test Reactor constitutes the second stage of India's three stage nuclear energy program. It is a 13.2 Megawatt electrical sodium cooled, loop type, mixed carbide fuelled reactor. The main purpose of FBTR is to acquire experience in the design, development and operation of the fast reactors. It serves as a test bed for irradiation of fuel materials and provides experience in large scale sodium handling and reactor operation. The reactor started operating with Mark I Core (70% PuC-30% UC) which is indigenously developed. It has two primary and secondary loops and a common steam water circuit with once through steam generator (SG) supplying super heated steam to the condensing turbine. There are two steam generators per loop and are located in the common casing. The heat transportation circuit has been divided in to two loops so that incase of non availability of one loop, the other loop is available for removing the decay heat from the core. Heat generated by the reactor is removed by these two primary sodium loops, and transferred to corresponding secondary sodium loops through intermediate heat exchangers. Flow sheet of FBTR is shown in Fig. 1.5 and the major parts of the reactor are briefly explained below.

1.5.1.1. Reactor Core

The core of the reactor constitutes fuel at the centre, surrounded by nickel reflectors, thoria blankets and steel reflectors. The core is vertical and freestanding, with

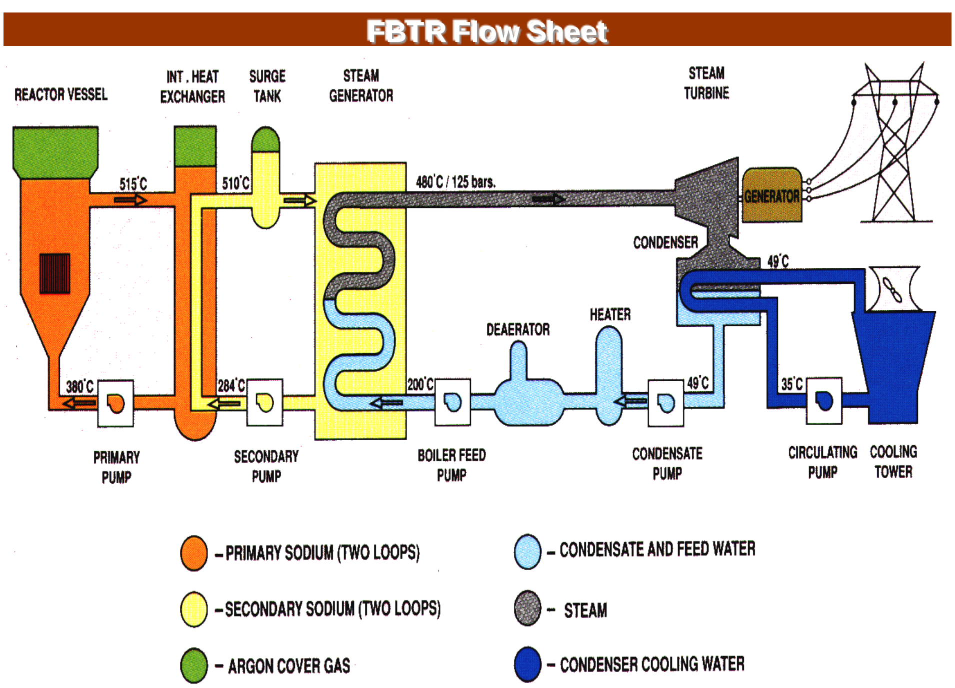


Fig. 1.5 Flow Sheet of Fast Breeder Test Reactor

the subassemblies supported at the bottom by the grid plate. The reactor vessel houses the core and serves as a conduit for the primary sodium coolant flow through the core. The sodium inlet pipe joins the reactor vessel at the bottom and two sodium outlet pipes radially branch out of the vessel above the core. The reactor is closed at the top by large and small rotatable plugs serving as top shields. Thermal shields are provided inside the reactor vessel to minimize the thermal stresses due to cold and hot shocks. A steel vessel with thermal insulation surrounds the reactor vessel.

1.5.1.2. Primary Sodium System

The major components of the primary system are reactor assembly, intermediate heat exchanger and two sodium pumps. The entire primary system is provided with a double envelope filled with nitrogen, designed to reduce the sodium level drop in the reactor in the event of any sodium leak. Primary sodium is pumped into the reactor by primary sodium pumps and flows by gravity to the intermediate heat exchangers (IHX) and back to the pump suction. The IHXs are vertical, counter-flow heat exchangers and transfer heat from the active primary sodium to the inactive secondary sodium. Primary sodium flows on the shell side and secondary sodium on the tube side. The shell is fixed and the tube bundles are removable. Secondary sodium is pumped into the IHXs by the secondary sodium pumps. After removing the heat from primary sodium, the secondary sodium enters the steam generator. The four sodium pumps are vertical, single stage centrifugal pumps with axial suction and radial discharge.

1.5.1.3. Secondary Sodium System

The main system consists of secondary sodium pumps, reheaters, surge tanks, steam generators and the connecting pipes. The steam generator modules are once-through, counter-flow type, with sodium entering the shell side from the top and water entering the tube side from the bottom. The steam generators are not insulated to facilitate removal of the decay heat by natural convection of air casing. The modules have a serpentine configuration, with evaporation and superheating occurring in a single pass. The surge tanks are cylindrical tanks interposed between the IHX and steam generator to dampen the pressure surges to the intermediate heat exchanger in the event of a sodium-water reaction in steam generator.

1.5.1.4. Steam Water Circuit

The steam water circuit forms the tertiary loop of the heat transport system. It consists of three subsystems viz; an on-line condensate polishing unit that meets the stringent feed water chemistry requirements of the once-through steam generators, feed water system and steam system. In once-through steam generator, sodium flows on the low pressure shell side whereas water/steam flows on the high pressure tube side. A cooling tower that serves as terminal heat sink is also present. It also consists of single cylinder, non reheating condensing turbine.

1.5.1.5. Instrumentation and Control

The reactor power and reactor shutdown are controlled by six control rods. For shutdown the control rods are inserted into the core by two methods, i.e., lowering of the rods, wherein all the rods are driven down by the respective drive mechanisms at a speed of 1 mm/s and Safety Control Rod Accelerated Movement (SCRAM), wherein the rods are dropped down by gravity in less than 400 milli-seconds. Lowering of rods is ordered by thermal hydraulics parameters of the plant, whereas SCRAM is ordered by neutronic parameters, core parameters and delayed neutron detection system.

Though FBTR is designed and developed based on the French reactor Rapsodie, more than 80% of the components were indigenously developed. The excellent performance of the components and structures during the past 26 years bears the testimony to the caliber of Indian industries [1.20-1.22].

1.5.2. Prototype Fast Breeder Reactor

Prototype Fast breeder Reactor (PFBR) is a 500 Megawatt electrical (MWe), sodium cooled, pool type, mixed oxide (MOX) fuelled reactor, with two secondary loops. The primary objective of PFBR is to demonstrate techno-economic viability of Fast Breeder Reactors on an industrial scale. PFBR being a commercial demonstration plant, mixed oxide fuel is selected on account of its proven capability for safe operation, high burnup, ease of fabrication and proven reprocessing. Pool type concept is adopted due to its inherently high thermal inertia of the large mass of sodium in the pool which eases the removal of decay heat. Two loop designs have been adopted from economic and safety point of view. The flow sheet of PFBR having all the components is shown in Fig. 1.6. The main components of PFBR are reactor core, reactor assembly, heat transport system and steam water system shown in the flow sheet given below. PFBR has been designed and constructed based on the experience gained from FBTR [1.17, 1.18].

1.5.2.1. Reactor Core

A homogeneous core concept with two fissile enrichment zones is adopted for power flattening. The active core where most of the nuclear heat is generated consists of 181 fuel subassemblies, 12 absorber rods viz; 9 control and safety rods and three diverse safety rods are arranged in two rings. Two independent and diverse shutdown systems are provided for ensuring safe shutdown of reactor even when one system is unavailable.

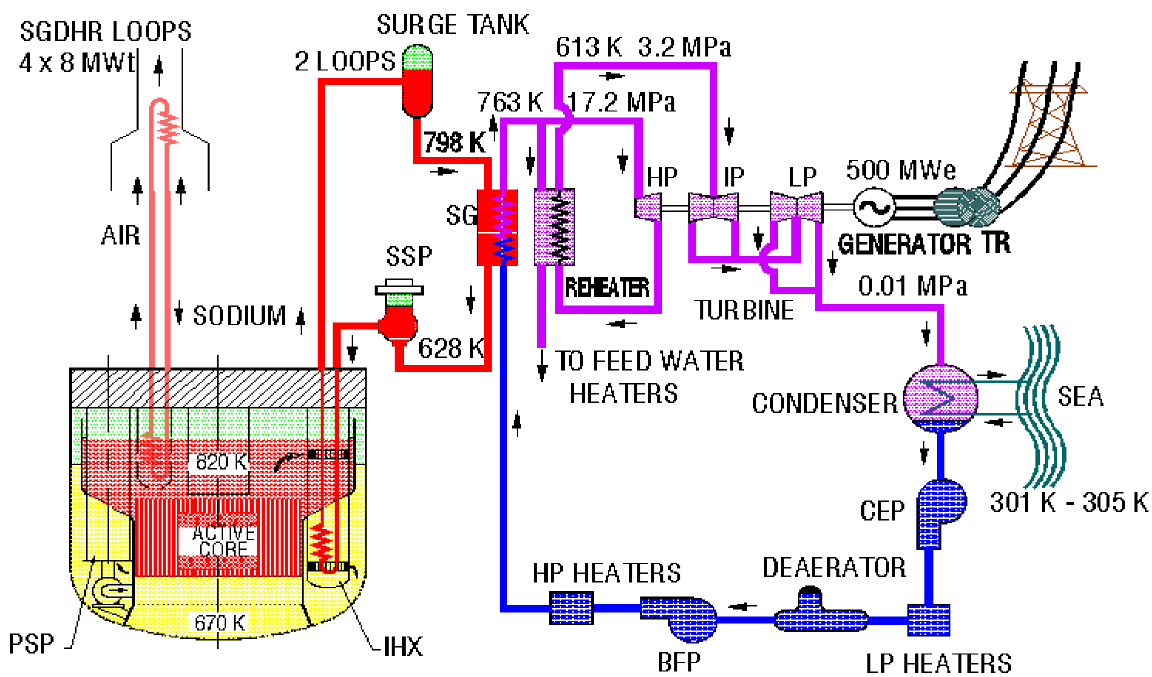


Fig. 1.6 Flow Sheet of Prototype Fast Breeder Reactor

1.5.2.2. Reactor Assembly

The entire primary sodium circuit consisting of core, primary pumps, intermediate heat exchanger and primary pipe connecting the pumps and grid plate, is contained in a single large vessel, main vessel. The main vessel is cooled by cold sodium to enhance its structural integrity. The main vessel is surrounded by safety vessel and the gap in-between is filled with nitrogen. An inner vessel separates the hot and cold pools of sodium. The main vessel is closed at its top by a top shield which includes roof slab, large and small rotatable plugs and control plug.

1.5.2.3. Heat Transport System

Liquid sodium is circulated by two primary sodium pumps through the core which in turn gets heated. The hot primary sodium is radioactive and it is not used directly to produce steam, but it transfers the heat to secondary sodium through intermediate heat exchangers. The non radioactive secondary sodium is circulated through two independent secondary loops, each having a sodium pump, two intermediate heat exchangers and four steam generators. The primary and secondary pumps are vertical, single stage and single suction centrifugal type. The steam generator is once through integrated type with straight tubes and an expansion in each tube. Decay heat removal under normal conditions is done by the Operation Grade Decay Heat Removal system (OGDHRS) of maximum 20 Megawatt capacity in the steam water system. In case of off-site power failure or non availability of steam water system, the decay heat is removed by passive Safety Grade Decay Heat Removal (SGDHR) circuit.

1.5.2.4. Steam Water System

The steam water system adopts a reheat and regenerative cycle using steam for reheating. High pressure superheated steam from the steam generators drives the turbo alternator. Three boiler feed pumps are provided to deliver the feed water to the steam generator.

1.6. Summary

In the present study, the artificial neural network algorithms are applied for parameter estimation and event identification in subsystems of Fast Breeder Reactors. The subsystems considered for application are intermediate heat exchangers of FBTR and neutronics system, primary sodium circuit of PFBR. Using various architectures and learning algorithms artificial neural networks are modeled for predicting the plant parameters and identifying the events. The details of case studies carried out in this research work are presented in Table 1. In the subsequent chapters the detailed description of case studies is given.

Table 1: Systems used for parameter estimation and event identification in Fast Breeder Reactor

<table>
<tr><th colspan="2">CASE STUDIES</th><th>INPUTS</th><th>OUTPUTS</th></tr>
<tr><td colspan="2">Intermediate heat exchanger (temperature parameter estimation of FBTR)</td><td>primary sodium inlet temperature , secondary sodium inlet temperature, primary sodium flow, secondary sodium flow</td><td>primary sodium outlet temperature , secondary outlet sodium inlet temperature in ^{0}C</td></tr>
<tr><td colspan="2">Neutronics system (reactor power estimation of PFBR)</td><td>nine control and safety rods positions (9 input parameters)</td><td>reactor power in Megawatt</td></tr>
<tr><td colspan="2">Neutronics system related event of PFBR</td><td>reactivity, linear Power, central subassembly temperature</td><td>control and safety rod withdrawal event</td></tr>
<tr><td rowspan="2">Primary sodium circuit related event of PFBR</td><td>Primary sodium pump trip</td><td>pump speed, power to flow ratio, central subassembly temperature, central subassembly temperature rise, mean core temperature rise</td><td>primary sodium pump trip event</td></tr>
<tr><td>Primary sodium pump seizure</td><td>pump speed, power to flow ratio, central subassembly temperature, central subassembly temperature rise, mean core temperature rise, reactivity</td><td>primary sodium pump seizure event</td></tr>
<tr><td rowspan="4">Event Detection System of PFBR</td><td>Control and safety rod withdrawal</td><td rowspan="4">reactivity, linear power, central subassembly outlet temperature, increase in central subassembly temperature rise, mean core temperature rise, power to flow ratio and Pump speed</td><td rowspan="4">control and safety rod withdrawal event

primary sodium pump trip event

primary sodium pump seizure event

primary pipe rupture event</td></tr>
<tr><td>Primary sodium pump trip</td></tr>
<tr><td>Primary sodium pump seizure</td></tr>
<tr><td>Primary pipe rupture</td></tr>
</table>

1.6. References

1.1 Saptoro A., An Integrated Approach to Artificial Neural Network based Process Modeling, *Thesis Chapter 1, Curtin University,* 2010 1-2 http://espace.library.curtin.edu.au/R?func=dbin-jump-full&local_base=gen01-era02&object_id=149373

1.2 Styer D. L., Vemuri V., Artificial Neural Network Controller for Multi-Link Structures, *Industrial Automation and Control, IEEE/IAS International Conference on (Cat. No.95TH8005),* 1995 37-42

1.3 Engelbrecht A.P., Computational Intelligence: An Introduction, *Jon Wiley & Sons*, 2007 1-6

1.4 Poole D., Mackworth A., Goebel R., Computational Intelligence A Logical Approach, *Oxford University Press,* 1998 1-6

1.5 Konar A., Computational Intelligence Principles: Techniques and Applications, *Springer International Edition* 2005

1.6 Craenen B.C., Eiben A.E., *Computational Intelligence, Encyclopedia of Life Support Icon Fuzzy Information Society Conference,* 2001 1-14 http://www.eolss.net/Sample-Chapters/C15/E6-44-02.pdf

1.7 Rudas I. J., Fodor J., *International Journal of Computers, Communications and Control*, 3 2008 132-138

1.8 Despande N., Artificial Intelligence, *Technical Publications Pune*, 2008 1-2

1.9 Marwala T., Vilakazi C. B., Computational Engineering, Finance, and Science, *Cornell university library*, 2007 http://arxiv.org/ftp/arxiv/papers/0705/0705.2604.pdf

1.10 Hines J.W., Uhrig R.E., *Progress in Nuclear Energy*, 46 2005 167-175

1.11 Ruan D., *EUSFLAT Conference,* 1 2003 799-802

1.12 Roverso D., *Progress in Nuclear Energy,* 46 2005 190-205

1.13 Vilakazi C.B., Marwala T., Mautla P., Moloto E., *WSEAS Transaction on Power Systems,* 1 2006 280-286

1.14 Uhrig R.E., Hines J.W., *Nuclear Engineering and Technology*, 37 2005 127-138

1.15 Uhrig R.E., Hines J.W., Integration of Artificial Intelligence Systems for Nuclear

Power Plant Surveillance and Diagnostics, *Short Course Notes*, 1998

1.16 Shaping the Third Stage of Indian Nuclear Power Programme, *Department of Atomic Energy, Government of India*, 1-22 http://www.dae.gov.in/publ/3rdstage.pdf

1.17 Raghupathy S., Singh O.P., Govindrajan S., Chetal S.C., Bhoje S. B., Design of 500 MWe Prototype Fast Breeder Reactor, *Indira Gandhi Center for Atomic Research, Department of Atomic Energy*, 1-9 http://www.dae.gov.in/ni/nimar04/design.pdf

1.18 Design of Prototype Fast Breeder Reactor, *Indira Gandhi Center for Atomic Research, Department of Atomic Energy, Kalpakkam* 2003 1-6 http://www.igcar.ernet.in/brochure/design.pdf

1.19 Velusamy K., Chellapandi P., Chetal S.C., Raj B., Overview of pool hydraulic design of Indian Prototype Fast Breeder Reactor, *Indira Gandhi Center for Atomic Research, Kalpakkam*

1.20 Suresh Kumar K.V., Kapoor R.P., Ramalingam P.V., Rajendran B., Srinivasan G., Kasiviswanathan K. V., Fast Breeder Test Reactor. 15 Years of Operating Experience, *Technical Meeting on Operational and Decommissioning Experience with Fast Reactors, IAEA-TM-25332, IAEA*, 2002 15-27

1.21 Srinivasan G., Suresh Kumar K. V., Rajendran B., Ramalingam P. V., *Nuclear Engineering and Design*, 236 2006 796-811

1.22 Suresh Kumar K.V., Srinivasan G., A brief Description of FBTR, *Indian Nuclear Society News*, 7 2010 5-11

Chapter 2

Artificial Neural Networks and their applications in the Operation and Maintenance of Nuclear Reactor

Artificial neural networks are examples of electronic models. This chapter gives a brief introduction to artificial neural networks, their advantages, disadvantages and comparison with their biological counterparts. This chapter also depicts their architectures, associated algorithms and applications of artificial neural networks in the maintenance and control of nuclear reactor design.

2.1. Principles of Artificial Neural Network

An artificial neural network (ANN) is a processing device, either an algorithm or an actual hardware that is inspired by the biological nervous system. It is well suited in real-time systems because of its parallel architecture leading to faster response and less computation time. A neural network tries to replicate the most basic function of brain. Brain is a highly complex, nonlinear and parallel computer having billions of nerve cells with trillions of interconnections. Hence mimicking a brain is not an easy task to do. The *cell body* in the neuron receives incoming impulses through *synapses* located on the *dendrites* (receiver) by means of chemical processes. If the number of incoming impulses exceed certain threshold value the neuron is *activated* and emits a signal though the *axon*.

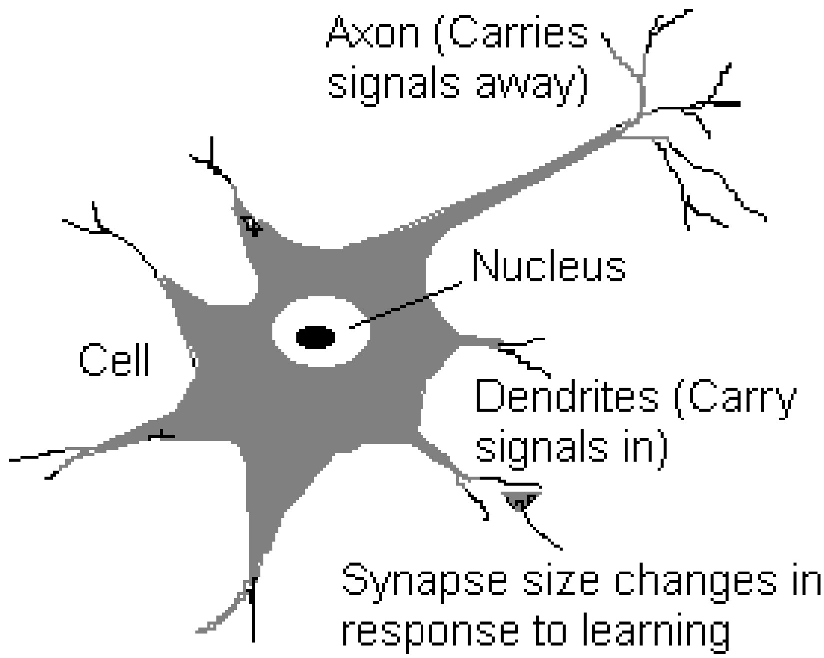


Fig. 2.1 Schematization of biological neuron

This signal might be sent to another synapse, and might activate other neurons. Fig. 2.1 represents a schematic of a biological neuron.

Artificial neural network consists of a large number of interconnected processing elements called neurons. However, the operation of an artificial neuron is far more simplified compared to the brain. An artificial neuron can be described as simple operator, performing multi-dimension input to one-dimension output mapping by adjusting the free parameters, called as the strength of connection units or the weight parameters. In artificial neural network, the knowledge lies in the interconnection weights between the neurons. Artificial neural networks learn by example, have the ability to derive meaning from complicated and imprecise data. A trained artificial neural network might be thought of as adept in some applications such as pattern recognition, data classification, optimization function, approximation, vector quantization etc. Fig. 2.2 shows promising areas where neural network has been implemented [2.1].

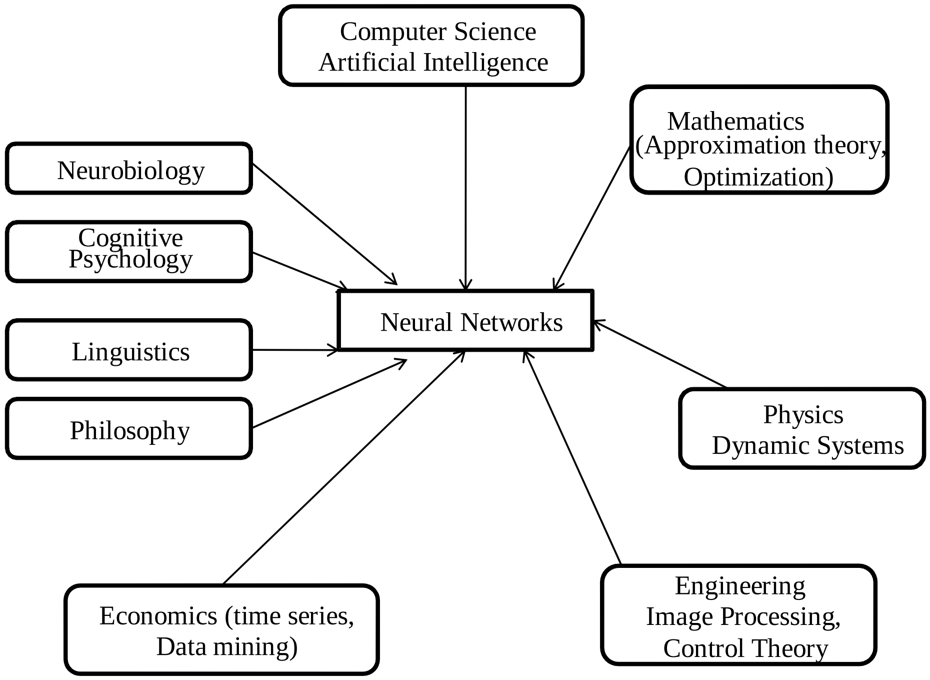


Fig. 2.2 Multi-disciplinary point of view of Neural Networks

The biological neural network and artificial neural network are compared in the following section.

2.2. Biological Neural Network versus Artificial Neural Network

Speed: Cycle time of execution of biological neural network is of nanoseconds while that of artificial neural network is of milliseconds. Hence the artificial neural networks are slow in processing information.

Processing: Artificial neural network performs instructions sequentially one after another, while human brain performs large number of parallel operations.

Size and complexity: Human brain has 10^{11} number of neurons and hence much more complex while artificial neural network has less number of neurons.

Storage capacity: Biological neural network stores the information in its interconnections or synapses. But in artificial neural network it is stored in some specific memory location. In artificial neural network sometimes new information overloads the older ones by destroying it. But in case of biological neural network new information can be added without destroying the older information.

Control mechanism: In artificial neural network there is a CPU which processes simpler interconnections and is free from chemical actions which take place in brain. Hence artificial neural network is much simpler than biological neural network [2.2].

The advantages of artificial neural network are summarized below.

2.3. Advantages of Artificial Neural Network

Artificial neural network is having capability of fast response and generalization from trained examples. This capability of artificial neural network enables to correctly classify patterns containing noise and incomplete or missing data. The non-algorithmic nature of artificial neural network simulation makes it possible to model complex systems where only data of system inputs and outputs are available. The artificial neural network characteristics are described as follows.

- Adaptive learning: Artificial neural network is empowered with the ability to learn how to do tasks based on the initial experience.
- Generalization ability: It possesses the capability to generalize, and make predictions for the new cases that are not trained.
- Self organization: It can self organize its structure based on the information given during learning.
- Fault tolerance: Some networks show the capability to learn even if there is partial destruction of neural network components.
- Parallelism: They are massively parallel and distributed in nature.

The objective of artificial neural network is to develop a computational device by modeling few characteristics of brain to perform various computational tasks. For neural network implementation, high-speed digital computers are required to make the simulation process feasible. A brief description of the basic models of artificial neural network is given below.

2.4. Basic models of Artificial Neural Network

The models can be specified on the basis of the three entities.

- Synaptic interconnections or architecture
- Training or learning rules
- Activation functions

2.4.1. Architectures of Artificial Neural Network

The arrangement of neurons in layers and the connection patterns formed within and between layers is called the network architecture. It may be of single layer or multilayer network. Input layer receives the input and helps in buffering the input signal. The output layer generates the output. Hidden layer has no direct contact with the external environment. It provides efficient output response. There exists two basic type of neural network architecture.

2.4.1.1. Feed Forward Network

In feed forward network the signals travel in one way from input to output. It forms an acyclic graph. There is no feedback (loops) i.e. 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 typical multi layer feed forward network is shown in Fig. 2.3.

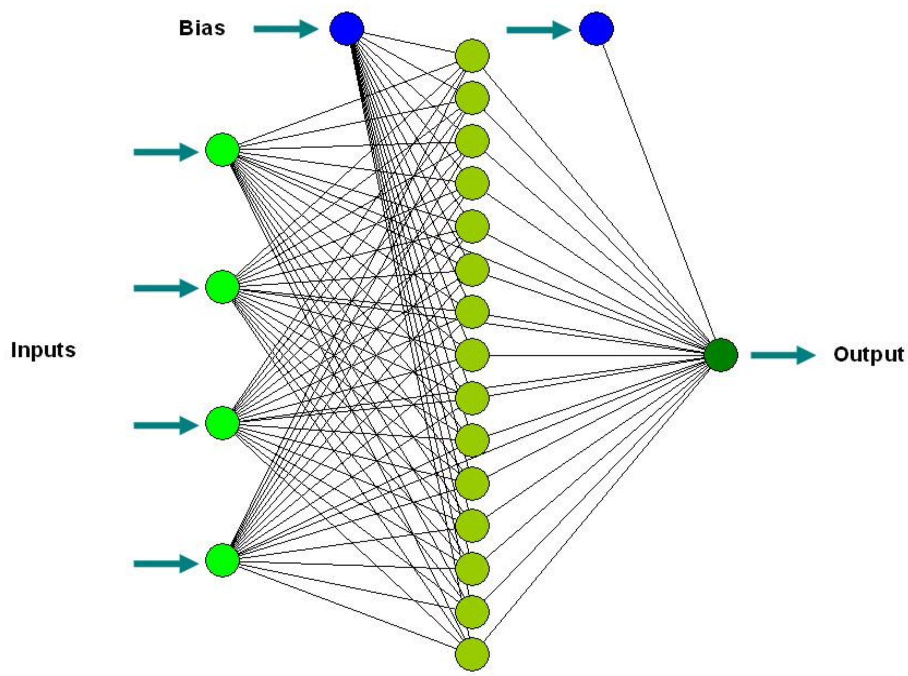


Fig. 2.3 A typical feed forward network

2.4.1.2. Feedback Network

The feedback neural networks have loops that feedback information to the hidden and input layers. They are developed to deal with the time varying or time-lagged patterns and are used for the problems where the dynamics of the considered process is complex and the measured data is noisy. Specific groups of the units get the feedback signals from the previous time steps and these units are called *context* unit. The network can be either fully or partially connected. In a fully connected network all the hidden units are connected recurrently, whereas in a partially connected network the recurrent connections are omitted partially. Examples of recurrent neural networks are Hopfield Networks, Regressive Networks, Jordan-Elman Networks, and Brain-State-In-A-Box (BSB) Networks [2.4]. A typical feedback network is presented in Fig. 2.4.

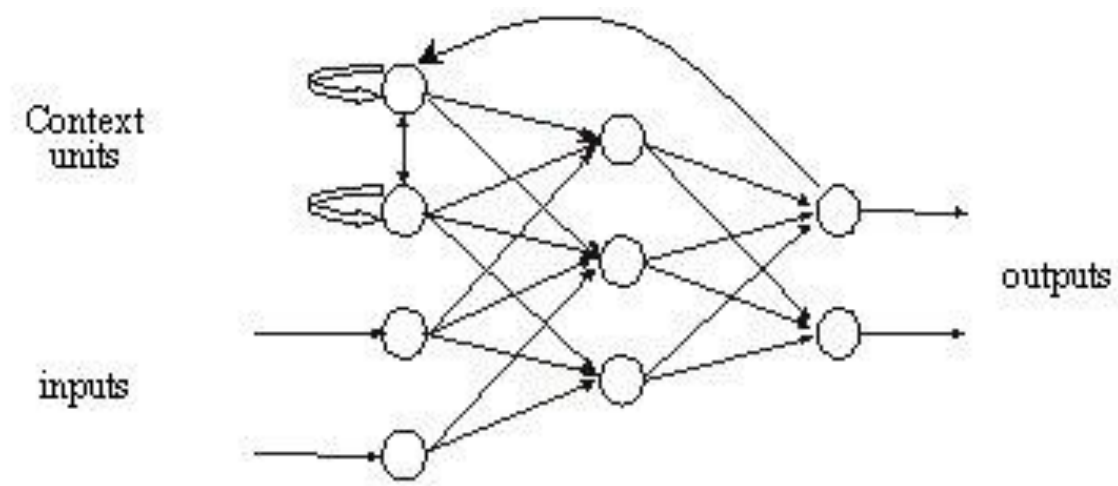


Fig. 2.4 A typical feedback network

2.4.2. Learning Algorithms

Learning is a process by means of which a neural network adapts itself to a stimulus by making proper weight adjustments which results desired response. Neural network has the inherent ability to adapt to a changing environment. In the process of adaptation they generate internal models of sampled environmental data. Learning algorithms define an architecture-dependent procedure to encode pattern information into weights to generate these internal models. It is an important characteristic of artificial neural network as in this process, the weights are adjusted and updated based on the training input and output parameters [2.5]. The selection of input parameters for neural network model development includes identification of the most relevant information about the desired output parameters. The various types of learning mechanisms presented are: Error Correction Learning, Hebbian Learning, Competitive Learning, and Boltzmann Learning. Error correction learning follows the simple delta rule which adjusts the weights using the error between the input and output. Hebbian learning is closest to biological learning. It describes that if both the pre-synaptic and post-synaptic neurons have got activated at the same time step, then the strength weights between them increased. Competitive learning follows the "winner takes all" rule where the inputs compete with each other and only the weights of winner are adjusted. Boltzmann learning is stochastic derived from statistical mechanics [2.6]. In a network, the probability of a state of neuron going to be changed is stochastic in nature. Fig. 2.5 illustrates the classification of learning algorithms.

Learning can also be classified into three categories as

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

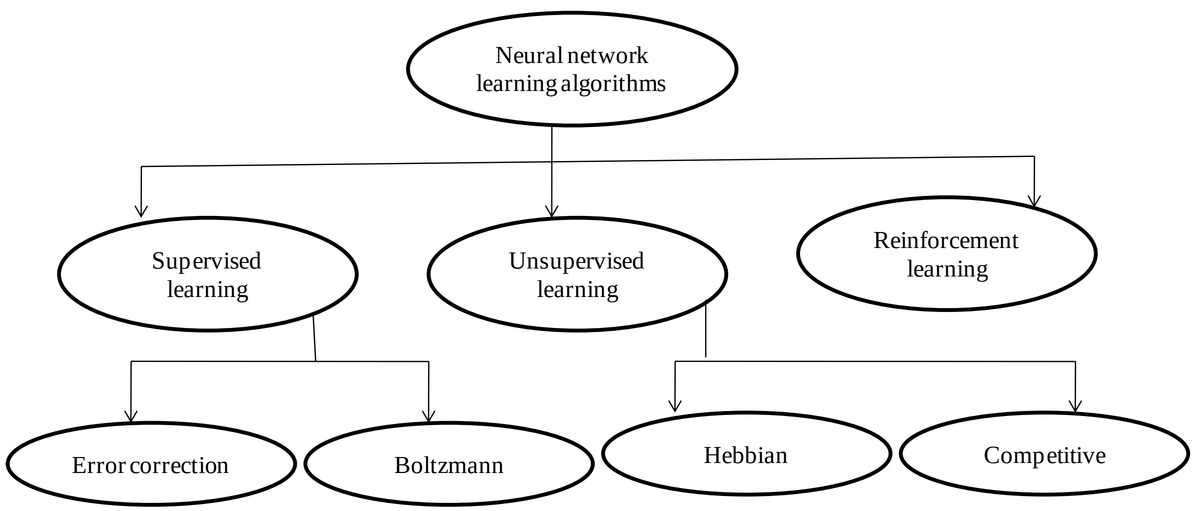


Fig. 2.5 Classification of learning algorithms

2.4.2.1. Supervised Learning

It is a type of learning performed with the help of a teacher. Each input vector requires a corresponding target output. During the training each input is presented to the network, which generates an actual output. This actual output is compared with a desired output and if there exits any difference, the difference or the error signal is used for the corresponding weight adjustment until the actual output matches the target output. Supervised learning is a process which adapts to reduce the output error for the current training pattern, with minimal disturbance to responses already learned [2.7].

2.4.2.2. Unsupervised Learning

It allows self-organization of its parameters to generate internal prototypes of sample vectors. The network here receives the input patterns and organizes these patterns to form clusters. When a new input pattern is applied, the network gives an output response indicating the class to which the input pattern belongs. Classes are derived from clusters by appropriate labeling. If for an input pattern, class cannot be found, then a new class is generated. This type of learning is often driven by a complex competitive-cooperative process where, individual neurons compete and co-operate with each other to update the weights based on inputs and out of that only winning neuron or clusters of neuron learn.

2.4.2.3. Reinforcement Learning

This type of learning is similar to supervised learning. But only critical information about the output is available instead of exact information. The network receives some feedback from its environment. But the feedback obtained here is evaluative not instructive. The external reinforcement signals are processed in the critical signal generator, and the obtained critical signals are sent to the artificial neural network for weight adjustments. It is also called as learning with critic.

2.4.3. Activation Function

Activation function or transfer function is applied to artificial neural network in order to achieve more efficient and exact output. The nonlinear activation function is used to make sure that a neuron's response is bounded. The actual response is conditioned or controlled by applying the activation function. For hidden units, sigmoid activation functions are usually preferable to threshold activation functions. Networks with threshold units are difficult to train because the error function is stepwise constant, hence the gradient either does not exist or is zero. Sigmoid units are easier to train than threshold units. With sigmoid units, a small change in the weights will usually produce a change in the outputs, which makes it possible to tell whether that change in the weights is good or bad. With threshold units, a small change in the weights will often produce no change in the outputs. Various types of activation functions are binary step function, bipolar step function, sigmoid function, ramp function etc. For the output units, activation function suited to the distribution of the target values should be properly chosen. Various types of transfer function graphs are shown in Fig. 2.6. The transfer functions are to be used for different targets given below.

- For binary targets, the logistic function is an excellent choice
- For continuous-valued targets with a bounded range, the logistic and tan-hyperbolic functions can be used.
- For continuous-valued targets with no known bounds, the identity or linear activation function can be used.

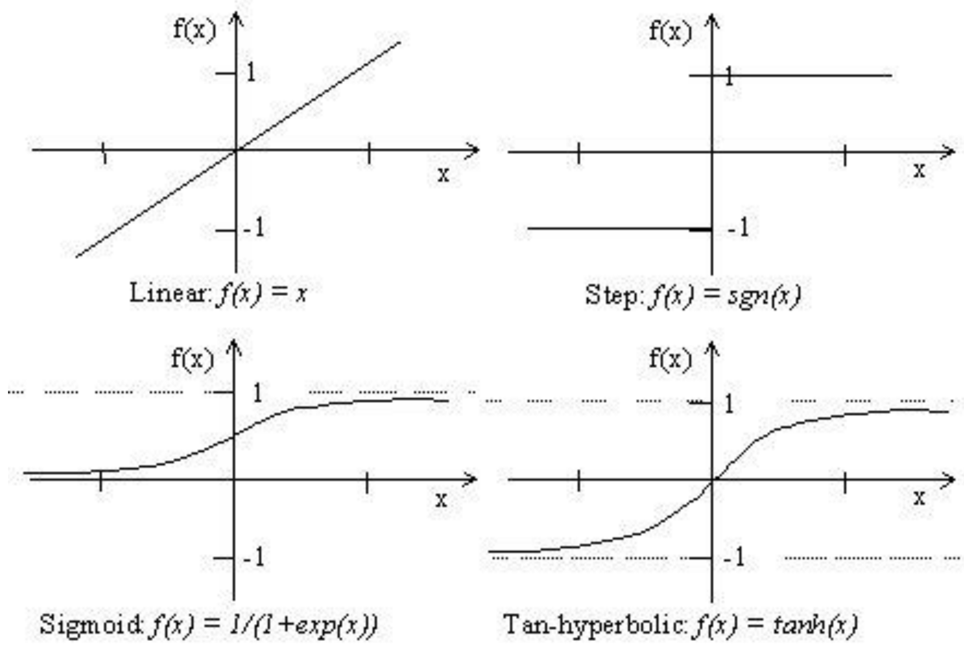


Fig. 2.6 Transfer functions

Artificial neural network models being nonlinear, data driven and having black box approach are very attractive choices for application in nuclear reactors. Artificial neural network plays a major role, whenever it is necessary to model complex or less understood processes with large input and output datasets, as well as to replace models that are too complicated to solve in real time [2.3]. The present study undertaken is primarily concerned with the development of neural network models for parameter estimation and fault detection in nuclear reactors.

2.5. Applications of Artificial Neural Network in Nuclear Reactors

Artificial neural network is one of the methods for aiding operation monitoring and diagnostics of a nuclear power plant. Artificial neural network is found to be one among the notably good non-linear statistical data modeling tool. Some of the applications of artificial neural network in nuclear power plant already implemented are listed below.

- Development of a fission gas release model to predict fission gas release in UO_2 fuel under reactivity initiated accidents conditions [2.8].
- Estimation of system parameters in Pressurized Water Reactor during transient conditions [2.9].
- Investigation of the deformation behavior of type 304L stainless steel during hot torsion in Fast Breeder Reactor [2.10].
- Development of a parallel multilayer neural network controller and simulation of load following operation for wide range of power regulation in Pressurized Water Reactor [2.11].
- Development of hybrid accident simulation methodology for prediction of critical parameters in Korean nuclear power plant [2.12].
- Numerical simulation of natural circulation Boiling Water Reactor to predict the thermal-hydraulic trends successfully [2.13].
- Prediction of critical heat flux parameter under low pressure for either natural circulation or forced circulation [2.14].
- Diagnosis of accidents based on Pressurized Heavy Water Reactor process parameters [2.15-2.17].
- Prediction of correlation between chemical composition, process variables and flow stress of austenitic stainless steels under hot compression [2.18].
- Prediction of the local power peaking factor in nuclear fuel reactor optimization in Boiling Water Reactor fuel lattice [2.19].

- Prediction of the release of volatile fission products from both Canada Deuterium Uranium (CANDU) Reactor and Light Water Reactor (LWR) under severe accident conditions [2.20].
- Neural network modeling for selecting the testing data of the self-unbalancing system to ensure sufficient perturbations covering proper dynamic and load conditions [2.21].
- Modeling of pressure drop coefficient for Cyclone Separators [2.22].
- Investigation of cavitations detection under given operating conditions to predict cavitation characteristics of any device without cavitation testing [2.23].
- Identification of vertical flow regime based on theoretical two phase flow simulation [2.24].
- Prediction of the influence of welding process on pitting corrosion behavior of the resistance spot welding joints of austenitic stainless steel [2.25].
- Detection of coolant boiling in the Hogner Onderwijs Reactor (HOR) [2.26].
- Modeling of the physical relationship among state variables to predict one particular variable among other ones in Belgian nuclear power plant [2.27].
- Fault detection and diagnosis in a realistic Heat Exchanger-Continuous Stirred Tank Reactor [2.28].
- Prediction of safety core parameters such as multiplication factor K_{eff} and fuel powers peaks $P_{\max}$ in Light Water Research Reactor [2.29].
- Problems of thermo-hydraulic prediction in advanced nuclear heat exchangers in evaluating, designing and optimizing the thermo-hydraulic performances [2.30].

- System identification, like predicting the concentration of reactor product and process monitoring and control [2.31].

In the present study, implementation of various neural network algorithms and hybrid genetic algorithm for subsystems of nuclear reactors subsystem has been taken up and described in the following chapters.

2.6. References

2.1 Sivanandam S.N., Deepa S.N., Principles of Soft Computing, *Wiley-India (P) Ltd.*, 2007 3-4

2.2 Yegnanarayana B., Artificial Neural Networks, *Prentice-Hall of India Private Ltd.*, 2008 19-21

2.3 Saptoro A., An Integrated Approach to Artificial Neural Network based Process Modeling, *Thesis Chapter 1*, *Curtin University*, 2010 1-2

2.4 Hrolenok B., Recurrent Neural Networks, *George Mason University*, 2009 http://www.cc.gatech.edu/grads/b/bhroleno/rnn_slides.pdf

2.5 Sivanandam S.N., Sumathi S., Deepa S.N., Introduction to Neural Networks using MATLAB 6.0, *Tata McGraw Hill Edu. Pvt. Ltd.*, 2006

2.6 Rajasekaran S., Vijayalakshmi Pai G. A., Neural Networks, Fuzzy Logic and Genetic Algorithms Synthesis and Applications, *PHI Learning Pvt. Ltd.*, 2008 19-20

2.7 Kumar S., Neural Networks A Class Room Approach, *Tata McGraw-Hill Edu. Pvt. Ltd.*, 2009

2.8 Koo Y., Oh J., Lee B., Tahk Y., Song K., *Journal of Nuclear Materials*, 405 2010 33-43

2.9 Akkurt H., Colak U., *Annals of Nuclear Energy*, 29 2002 2087-2103

2.10 Mandal S., Sivaprasad P.V., Venugopal S., Murty K.P.N., Raj B., *Materials Science and Engineering A*, 485 2008 571-580

2.11 Khajavi M.N., Menhaj M.B., Suratgar A.A., *Annals of Nuclear Energy*, 29 2002 751-760

2.12 Choi Y J., Kim H K., Back W P., Chang S H., *Information Sciences- Informatics and Computer Science: An International Journal*, 60 2004 207-224

2.13 Garg A., Sastry P.S., Pandey M., Dixit U.S., Gupta S.K., *Nuclear Engineering and Design*, 237 2007 230-239

2.14 Guanghui S., Morita K., Fukuda K., Pidduck M., Dounan J., Miettinen J., *Nuclear Engineering and Design*, 220 2003 17-35

2.15 Santosh T.V., Vinod G., Babar A.K., Kushwaha H.S. Raj V.V., *Reliability*

Engineering & System Safety, 83 2003 33-40

2.16 Santosh T. V., Vinod G., Saraf R.K., Ghosh, A.K., Kushwaha H.S., *Reliability Engineering & System Safety*, 92 2007 1468-1472

2.17 Santosh T.V., Kumar M., Thangamani I., Srivastava A., Dutta A., Verma V., Mukhopadhyay D., Ganju S., Chatterjee B., Rao V.V.S.S., Lele H.G., Ghosh A.K., *Nuclear Engineering and Design*, 241 2011 177-184

2.18 Mandal S., Sivaprasad P.V., Venugopal S., Murthy K.P.N., *Applied Soft Computing*, 9 2009 237-244

2.19 Montes J. L., Francois J. L., Ortiz J. J., M. C., Perusquia R., *Annals of Nuclear Energy*, 36 2009 121-130

2.20 Andrews W. S., Brent J. L., Cox D. S., *Journal of Nuclear Materials*, 270 1999 74-86

2.21 Lu S., Hogg B.W., *Electrical Power and Energy Systems*, 22 2000 67-78

2.22 Zhao B., Su Y., *Chemical Engineering Research Design*, 88 2010 606-612

2.23 Gupta P.K., Kumar P.A., Kaul A., Pandey G.K., Padmakumar G., Prakash V., Anandbabu C., *Proc. National Seminar on NDE*, 2006 245-254

2.24 Mi Y., Ishii M., Tsoukalas L.H., *Nuclear Engineering and Design*, 204 2001 87-100

2.25 Martin O., Tiedra P. D., Lopez M., *Corrosion Science*, 52 2010 2397-2402

2.26 Kozma R., Nabeshima K., *Annals in Nuclear Energy*, 22 1995 483-496

2.27 Renders J.M., Goosens A., Viron F. D., Vlaminck M. D., *Fuzzy Sets and Systems*, 74 1995 139-151

2.28 Dzwinel W., Pepyolyshev Y.N., Janiczk K., *Progress in Nuclear Energy*, 43 2003145-150

2.29 Mazrou H., Hamadouche M., *Progress in Nuclear Energy*, 44 2004 263-274

2.30 Ridluan A., Manic M., Tokuhiro A., *Nuclear Engineering Design*, 239 2009 308-319

2.31 Ahmad A., *Proceedings of Eleventh Symposium of Malaysia Chemical Engineers*, 1995 C3-1-C3-4

<u>Chapter 3</u>

Parameter Estimation of Fast Breeder Reactor Subsystem using Artificial Neural Network

---

This chapter gives brief introduction to the subsystems of sodium cooled fast reactor and the estimation of the parametric values using the artificial neural network. Primary and secondary sodium outlet temperatures of intermediate heat exchanger of FBTR have been predicted using the back propagation multi layer perceptron (MLP) algorithm. Sodium temperatures of intermediate heat exchanger has also been estimated by standard back propagation as well as radial basis network taking into consideration the reactor data, starting from steady state to the transient states. A comparison study has been carried out between the two algorithms. The reactor power of PFBR is also determined using the positions of control and safety rods implementing the standard back propagation algorithm and a comparison study is carried out applying variants of back propagation algorithms.

3.1. Process Modeling of Intermediate Heat Exchanger of FBTR for Steady State Condition using Artificial Neural Network

Neural network is a powerful tool for prediction of relevant physical parameters which are difficult to measure conventionally in nuclear reactor subsystems [3.1]. Prediction of temperature of sodium in intermediate heat exchanger of FBTR is very important for monitoring the state of the reactor. Hence accurate evaluation of sodium temperature is of major concern in case of both offline and online operation of the plant. Artificial neural network model has been developed to predict process parameters of intermediate heat exchanger subsystem of FBTR. Based on the theoretical computations carried out for primary and secondary outlet temperatures of intermediate heat exchanger, the input-output datasets are generated. Multi layer neural network is implemented with the standard back propagation algorithm to train the theoretically generated datasets. The advantages of artificial neural network over conventional method of temperature calculation include faster and accurate prediction. Artificial neural network being a data driven technique needs to know little about the process itself [3.2]. The feed forward neural network model learns about the reactor in steady state conditions and performs nonlinear regression analysis [3.3-3.7]. The main objective of regression analysis is to predict the value of continuous variables even if the variable interactions are not completely understood.

3.1.1 Intermediate Heat Exchanger

The intermediate heat exchanger transfers the heat from primary sodium which is radioactive to non radioactive secondary sodium in a reactor. It is housed in a fixed vessel with its double envelope. It is a vertical shell and tube counter flow unit featuring single pass on both shell and tube with primary sodium on shell side and secondary sodium on tube side. The tube bundle is placed in a fixed shell immersed in sodium pool. The intermediate heat exchanger is supported on the roof slab and is removable from the top. Heat passes from hot pool to cold pool through a stand pipe of the inner vessel. The sealing between the hot pool and cold pool is by mechanical seal. A manually operated sleeve type valve is provided on each intermediate heat exchanger for primary sodium

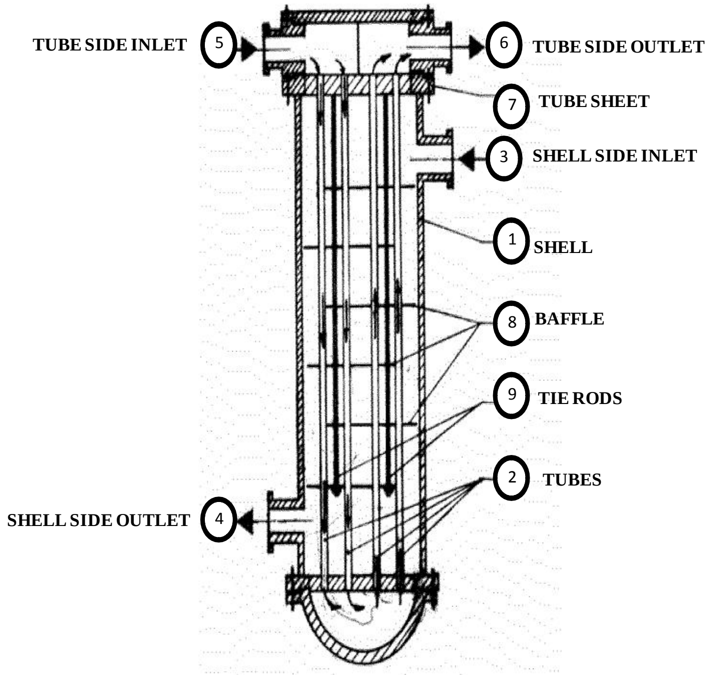


Fig. 3.1. Schematic of Intermediate Heat Exchanger of Fast Breeder Reactor

side isolation in the event of reactor operation with one secondary sodium circuit. In the reference fast reactor system there are two intermediate heat exchangers in primary circuit, one in each loop [3.8]. The schematic of intermediate heat exchanger is shown in Fig. 3.1.

When reactor is in operation, the control rods are taken out partially so that the neutrons generated can actively participate in the fission reaction, generating heat. The primary sodium will take heat from the radiated core and then will move in upward direction (because of low density property of sodium). Then it will reach up to intermediate heat exchanger and enters in radial manner to the shell side of intermediate heat exchanger at the top. The primary sodium flows vertically downwards and comes out radially at the bottom to the cold pool. The secondary sodium flows upwards inside the tubes. Heat is then transferred from primary sodium to secondary sodium which in turn goes to the steam generator to produce superheated steam. This steam rotates the turbine to generate electricity.

3.1.2 Architecture of Neural Network Model for Intermediate Heat Exchanger

Variable selection for process modeling of intermediate heat exchanger includes identification of the parameters having the most relevant information about the desired output parameters [3.9]. The input layer has five input nodes (including bias) which are primary inlet temperature, primary flow, secondary inlet temperature and secondary flow. The output layer has two output nodes which are primary outlet temperature and secondary outlet temperature. The neural architecture of the intermediate heat exchanger is represented in Fig. 3.2.

3.1.3 Training with Back Propagation Algorithm

The objective of neural network training is to adjust the parameters of the network based on a given set of input–output pairs. Neural network training is usually divided into two phases: off-line and on-line learning. For off-line training of a neural model, it is necessary to identify an error criterion that is used to determine when learning is complete. Once off-line learning phase is completed and a relatively accurate empirical model of the reactor core has been identified, further learning is accomplished online, capturing any dynamics not included in the training set used in the prior learning phase as well as for tracking any slow plant drifts [3.10].

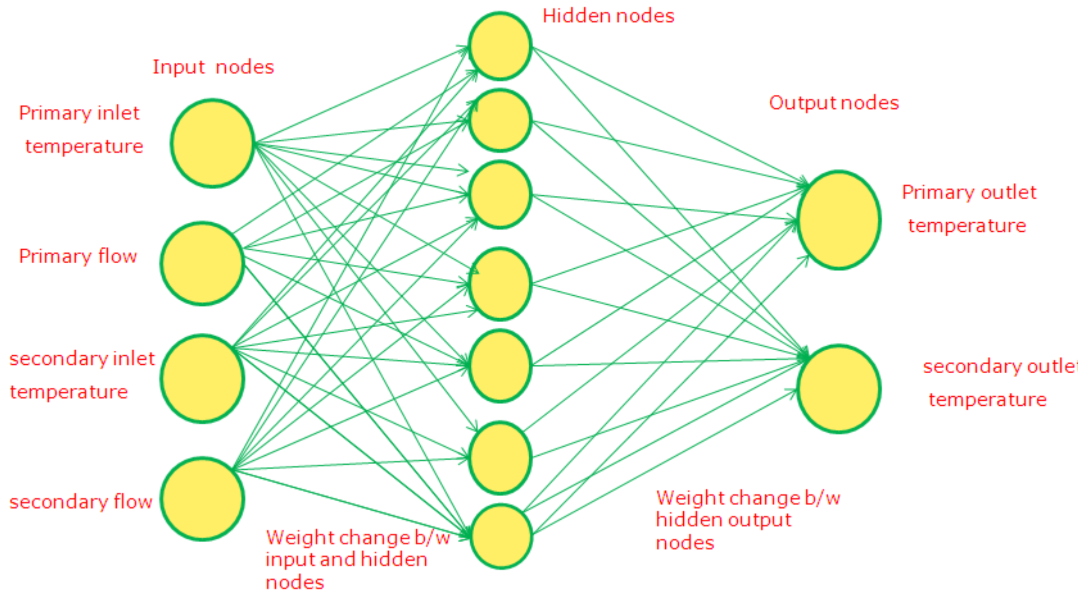


Fig. 3.2 Architecture of neural network model for Intermediate Heat Exchanger

Back propagation is the method for computing the gradient of the case-wise error function with respect to the weights for a feed-forward network, a straightforward but elegant application of the chain rule of elementary calculus. Standard back propagation can be used for both batch training (in which the weights are updated after processing the entire training set) and incremental training (in which the weights are updated after processing each case). For batch training, standard back propagation usually converges (eventually) to a local minimum, if one exists. For incremental training, standard back propagation does not converge to a stationary point of the error surface. To obtain convergence, the learning rate must be slowly reduced.

The back propagation algorithm uses the supervised learning which consists of two passes. One is forward pass, where the input values are propagated from input layer to output layer through hidden layer by applying non linear sigmoid function at each layer and finally matching the desired output with the actual output. This generates an error which in squared form is known as mean square error (MSE). The other is backward pass, where the estimated mean square error is back propagated to adjust the weight vectors. It minimizes a quadratic cost function by a gradient descent method. The steps involved in back propagation algorithm are presented below.

1. Present a training sample to the neural network.
2. Compare the output of network with the desired output. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a *scaling factor*, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Update the weights and repeat from step 3 on the neurons at the previous level.
6. Save the weights and outputs.

Since the input training data is of wide range, it has to be scaled down between zero to one which is also known as normalization. The normalization formula is given by equation 3.1.

$$d_{norm} = \frac{d - d_{\min}}{d_{\max} - d_{\min}} \qquad (3.1)$$

where d_{norm} is the normalized value of the input and d is the input parameter value and $d_{\max}$ is the maximum value, $d_{\min}$ is the minimum value of the respective parameters.

Back propagation algorithm is a steepest gradient descent algorithm. The multi layer perceptron is having a nonlinear transfer function known as sigmoid activation function. The activation function has been modified as follows in order to have better convergence and is presented by equation 3.2.

$$f = \left[\frac{2}{1+e^{-x}}\right] - 1 \qquad (3.2)$$

Subsequently fine tuning of the algorithm is required, in order to make mean square error as small as possible. To achieve this, different number of hidden nodes (nhn) and learning rate (lr) parameters are used and the network is trained so as to fix their values where mean square error is least. The formula for mean square error is given in equation 3.3

$$\text{Mean Square Error} = \frac{1}{TSN}\sum_{t=1}^{TSN}\sum_{k=1}^{NON} * \left(dout_{kt} - O_{kt}\right)^2 \qquad (3.3)$$

where TSN is the number of training samples and NON is the number of outputs nodes, $(dout_{kt})$ is the desired output and O_{kt} is the actual output. One more parameter known as momentum can also be used which filters out high-frequency changes in the weight values. By iterating the algorithm, the desired weight parameters can be found.

The weight parameters can be calculated using equations 3.4a and 3.4b.

$$W_{kj} = W_{kj} + \delta W_1 \qquad (3.4a)$$

$$W_{ji} = W_{ji} + \delta W_2 \qquad (3.4b)$$

where W_{kj} is the weight vector from hidden to output layer, W_{ji} is the weight vector from input to output layer and δW_1, δW_2 are the respective change in weight parameters. When the testing data is given to the algorithm, using the above weight parameters the neural network will be able to generate outputs almost matching with the desired outputs. This can be represented as ability to generalize.

The main advantage of neural network is that any unknown plant conditions, which the input sample set never learnt before can approximately be calculated. The artificial neural network with adaptive learning finds the weights and consequently the model is changed by updating the weights. The iterative process is continued until the convergence is found. So when any unseen data is given to the artificial neural network it generalizes the characteristics and produces an output corresponding to the input. The back propagation algorithm tries to minimize the mean square error value between the desired output and actual output. To make the algorithm converge faster, it is needed to use a learning rate coefficient (lr). The learning rate has to be selected carefully because if it is too small, the convergence is extremely slow and if it is too large, the artificial neural network may not converge at all. The learning rate and the number of hidden nodes are varied one by one such that after fine-tuning, result having minimum mean square error is obtained. The coding has been done in C language.

3.1.4 Results and Discussion

Out of 150 samples, a dataset of 140 are used for training. After training, learning rate is fine-tuned and found to be $lr = 0.015$ and number of hidden nodes to be $nhn = 12$ at 3×10^5 iterations, with mean square error value=1×10^{-4}. The dataset for training is presented in Table A of Annexure 1 and the output obtained using the test data are compared with the desired output and presented in Table B of Annexure 1. Fig. 3.3 shows the optimal hidden nodes and the learning rate parameter. To show the predicted outputs, the graph is plotted between the total number of training sample sets and the corresponding outputs. As observed from the graph of Fig. 3.4, the actual output approaches to the desired output.

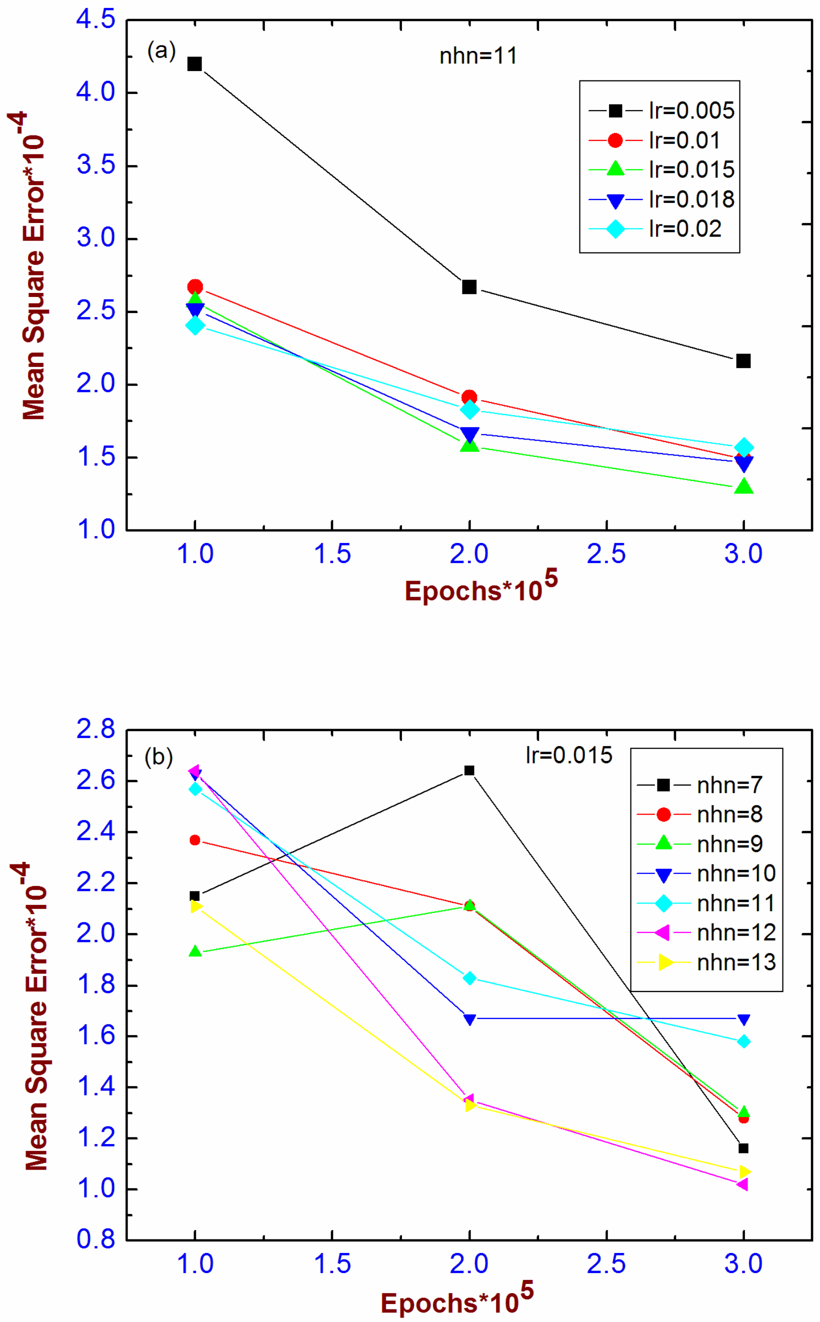
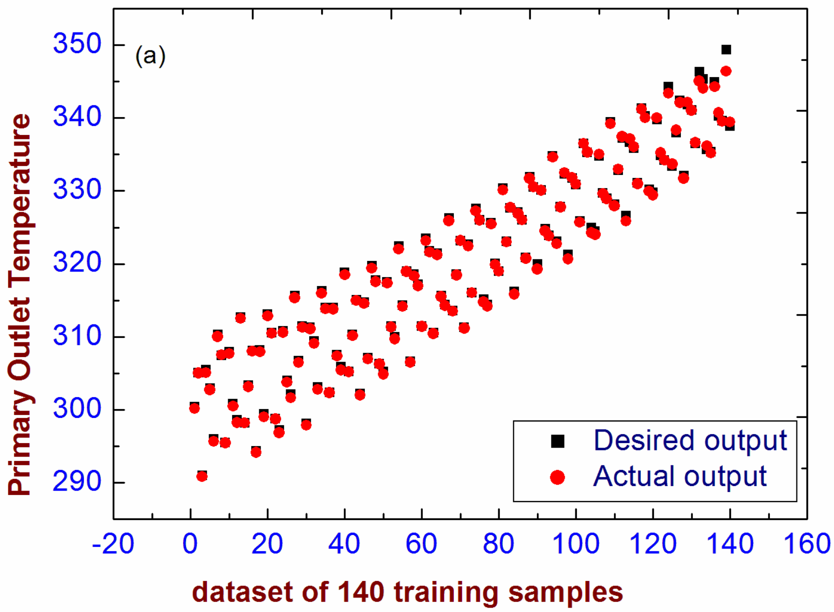


Fig. 3.3 Mean square error versus number of epochs for (a) learning rate (b) various hidden nodes for Intermediate Heat Exchanger



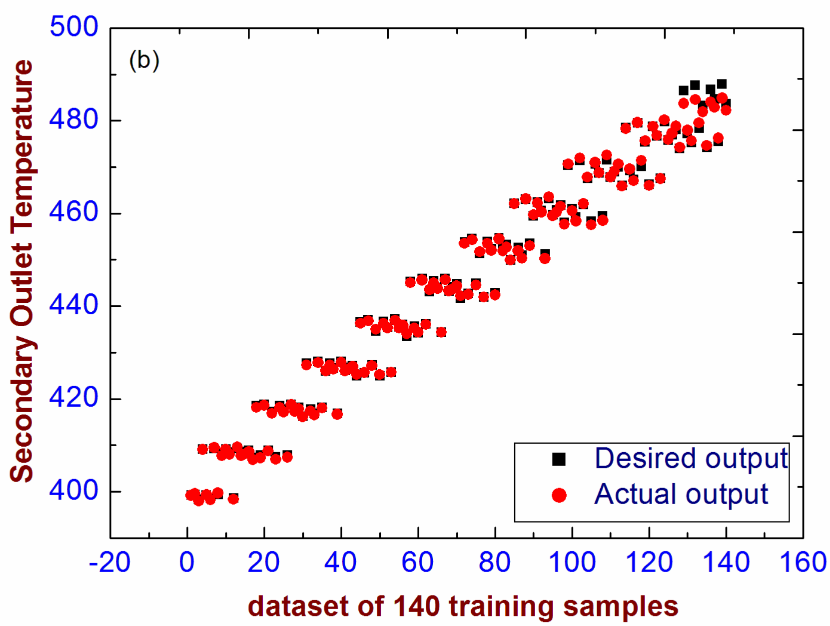


Fig. 3.4 Estimation of (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger for training sample index

Neural network models are data driven and therefore resist analytical or theoretical validation. These models are constructed by training using a data set, i.e., the model shifts from a random state to a “trained” state, and must be empirically validated. Validation of the back propagation algorithm is carried out with a test dataset which is within the range of the training sample data. Ten samples of testing data have been used for validation. From Fig. 3.5 it has been found that the trained neural network model is able to generalize and give a meaningful output for the untrained data. The graph shows the testing results of ten unknown samples from which it is observed that the desired and actual outputs are in well agreement with each other. The difference between the desired and actual output is set as ± 2 °C which is well within the permissible limit. The predicted outputs are compared with the actual outputs are compared in Fig. 3.6.

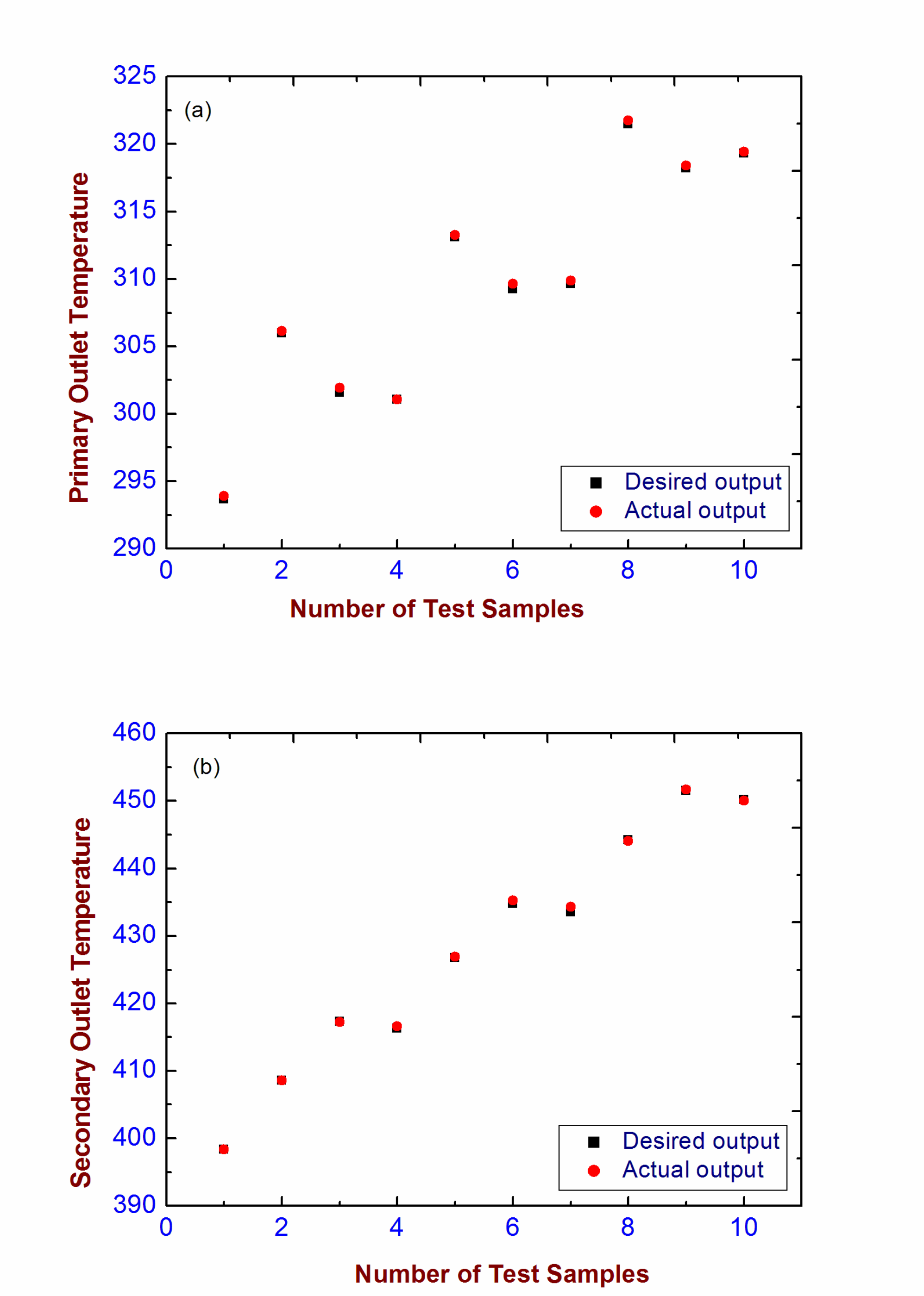


Fig. 3.5 Estimation of (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger for test sample index

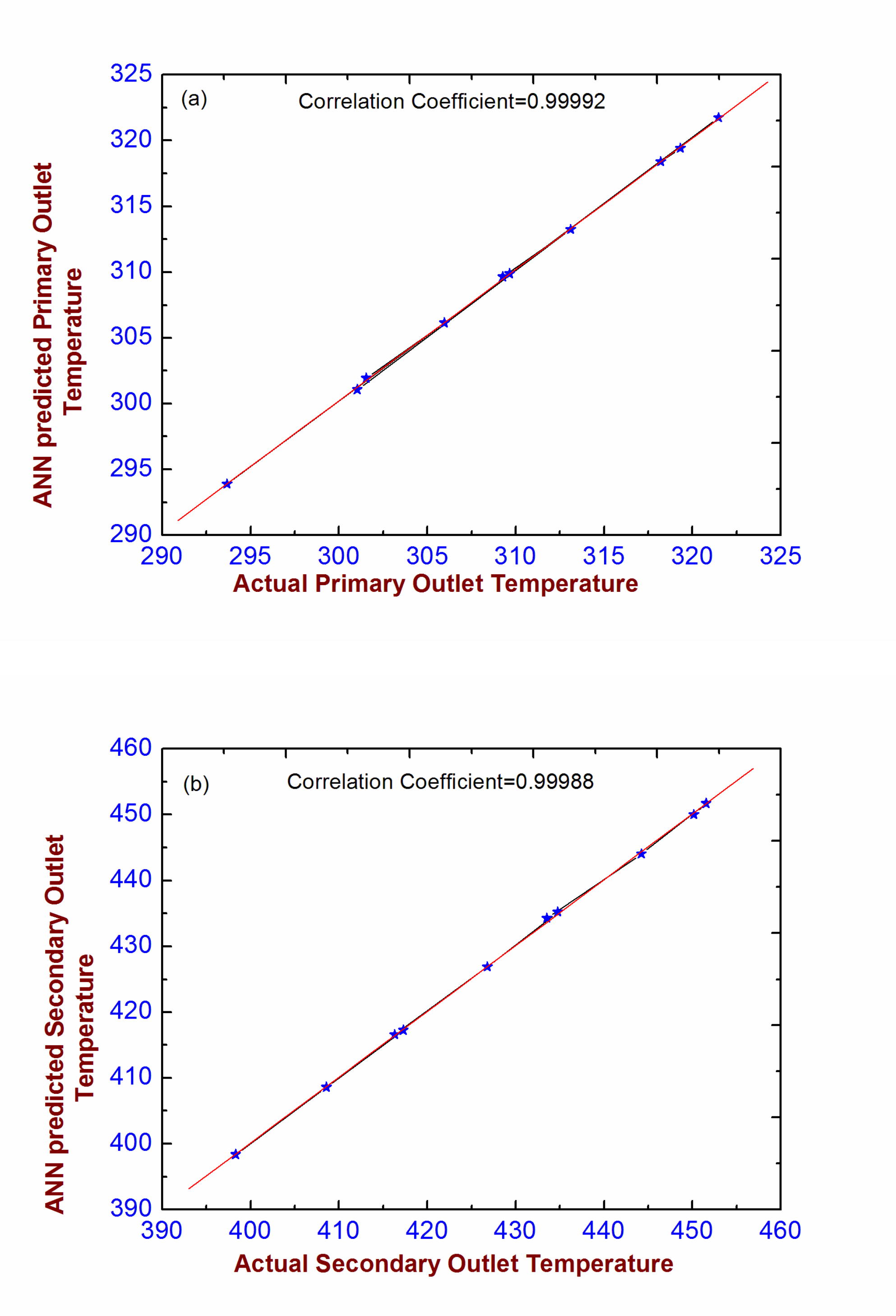


Fig. 3.6 Scatter plot between actual and desired output values for (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger

The higher value of correlation coefficient between the actual temperature of intermediate heat exchanger and temperature predicted by artificial neural network indicates network having good fit. From Fig. 3.6, it is shown that the correlation coefficient value is 0.99992 and 0.99988 for primary outlet temperature and secondary outlet temperature respectively. The time taken to obtain the training data set is 30 minutes using theoretical calculations in a personal computer (2.66 GHz Intel Core 2 Duo processor). Whereas, in neural network it is giving outputs in milliseconds range.

3.2. Neural Network Model for Intermediate Heat Exchanger using Real Time Data

Nuclear reactors are by nature nonlinear and their performances vary with time as a function of power level, fuel burnup, and control rod positions. Neural network model shows considerable reduction in computation time in comparison with conventional system and also shows good stability and performance for a wide range of operation [3.11]. The sole purpose of this model is to compliment the conventional model [3.12]. The model can be applied for prediction of sodium outlet temperature of intermediate heat exchanger as a function of sodium inlet temperatures and flow parameters [3.13].

By taking the real time data from the FBTR, an artificial neural network model for intermediate heat exchanger subsystem has been developed. The data has been taken from the online operation of the reactor from reactor startup to 18 Mega Watt thermal power from the Central Data Processing System (CDPS). A brief description about the central data processing system architecture is explained below.

3.2.1 Central Data Processing System Architecture

Central data processing system in FBTR consists of two real time embedded computer based system connected in fault tolerant configuration. Each system is classified as follows.

1) Safety Critical System (SCS)
2) Safety Related System(SRS)
3) Non-Safety System(NSS)

Fig. 3.7 shows the central data processing system architecture from where the real time data is saved. The data sent by each of the three systems has to be presented to the user as a single user interface. Hence, the purpose of the data server is to receive the data sent by safety critical system, safety related system, and non-safety system, store them and present to the operator using graphical user interface. The safety critical, safety related real time data has been taken from the data server in every half an hour interval of time from reactor startup to 18 Megawatt thermal power using graphical user interface software in control room of FBTR.

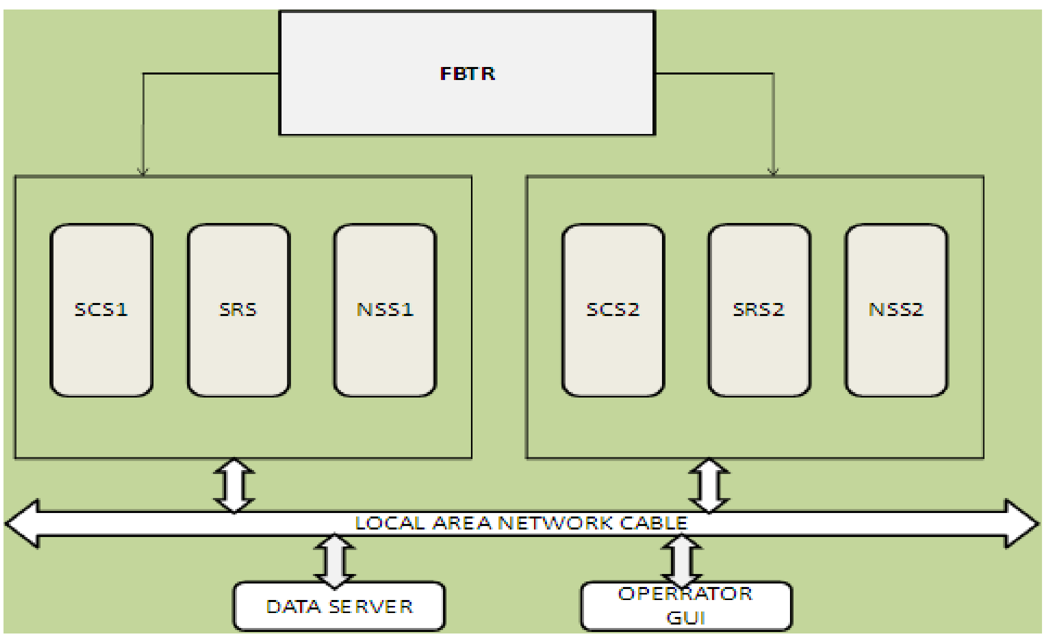


Fig3.7. Schematic of Central Data Processing System Architecture

3.2.2 Results and Discussion

451 training dataset and 14 test dataset have been used. Training and fine tuning of the parameters has been carried out for different hidden nodes and learning rates. From Fig. 3.8 it is seen that for number of hidden node 3 and learning rate 0.007 the mean square error is least.

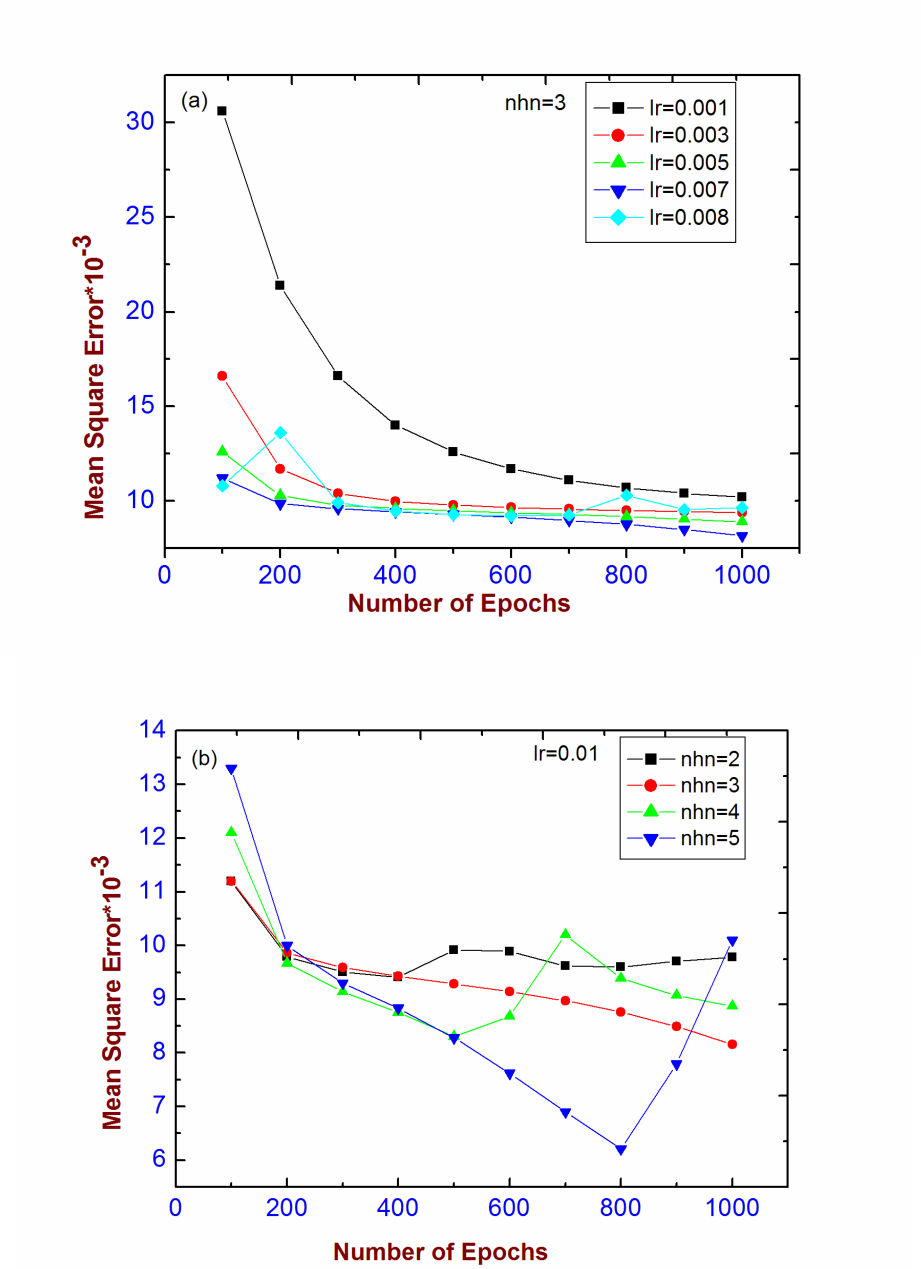


Fig. 3.8 Mean square error versus number of epochs for (a) learning rate (b) various hidden nodes for Intermediate Heat Exchanger

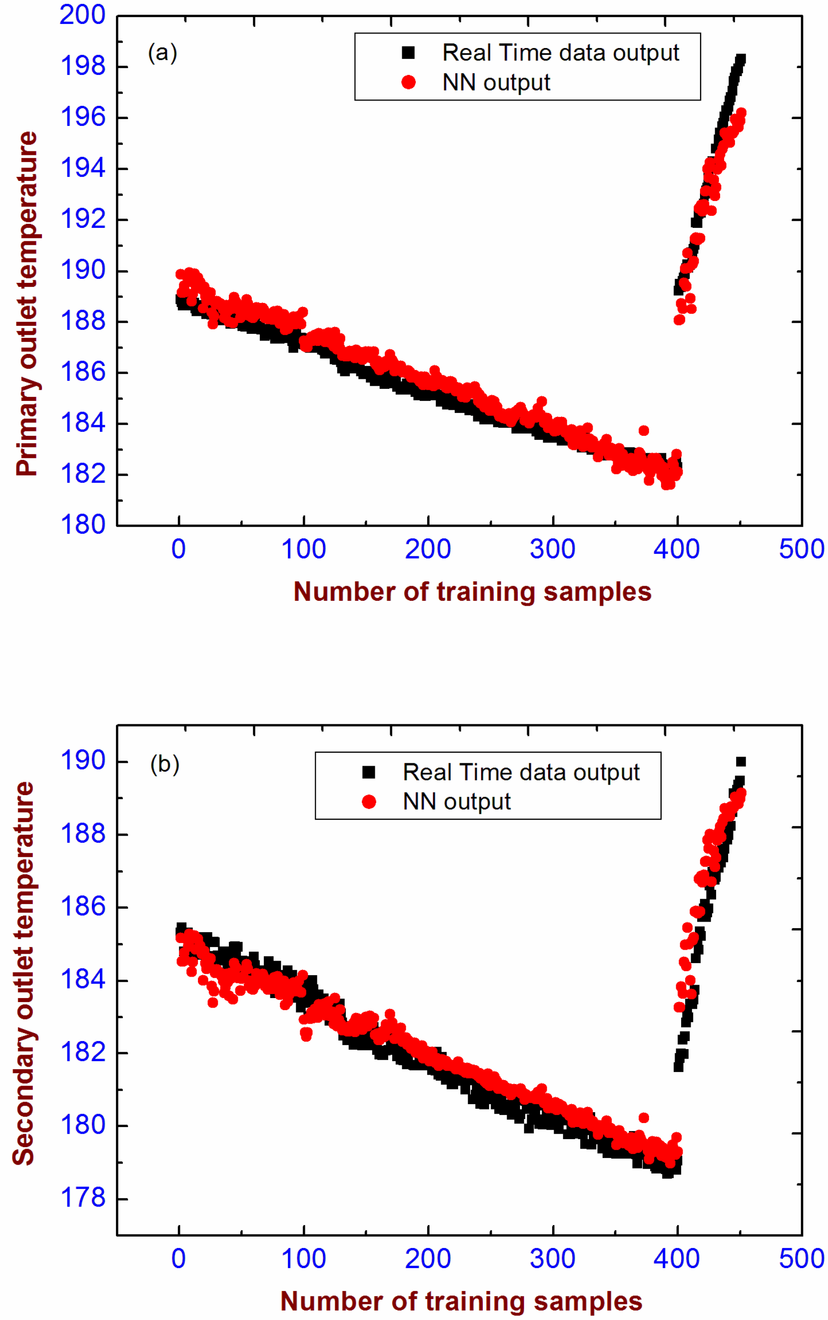


Fig. 3.9 Estimation of (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger for training sample index

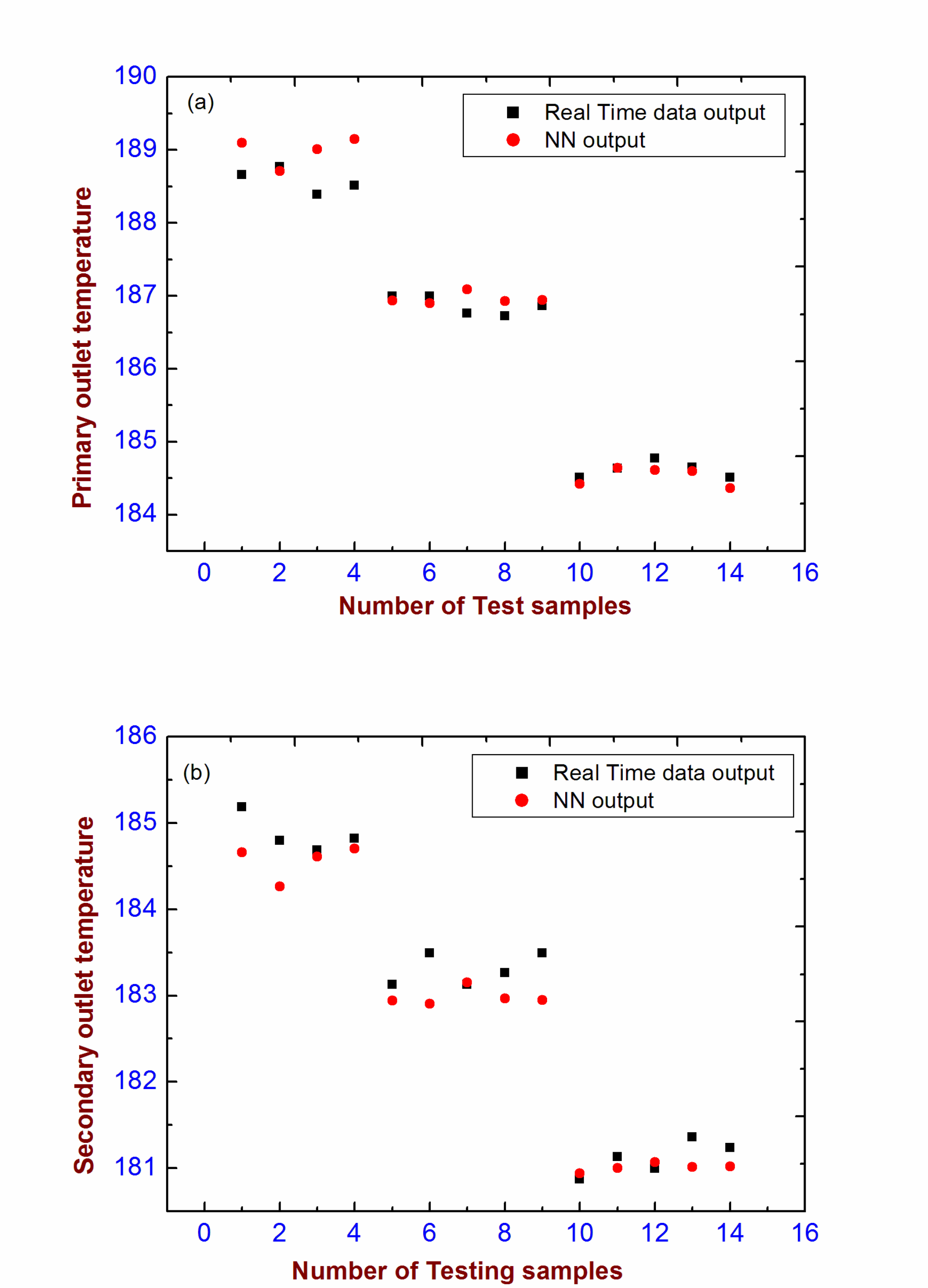


Fig. 3.10 Estimation of (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger for test sample index

After choosing the learning rate and hidden nodes the program is run for an additional number of epochs and it is found that at 6000 epochs, convergence of the algorithm is obtained. The graph shows the comparison between the real time data and the neural network data for training and testing. Fig. 3.9 shows that both the desired and actual output for training samples is overlapping with each other.

Fig. 3.10 depicts the ability to predict the outlet temperature using the testing samples. It shows the comparison between the real time output and the output obtained from the neural network. The difference limit between the desired and actual output is set as ± 2 °C and it tends to be in the permissible range. Fig.3.11shows that, the correlation coefficient value is 0.99393 and 0.99017 for primary outlet temperature and secondary outlet temperature respectively, which is very close to 1 indicating a good fit between desired and actual artificial neural network output.

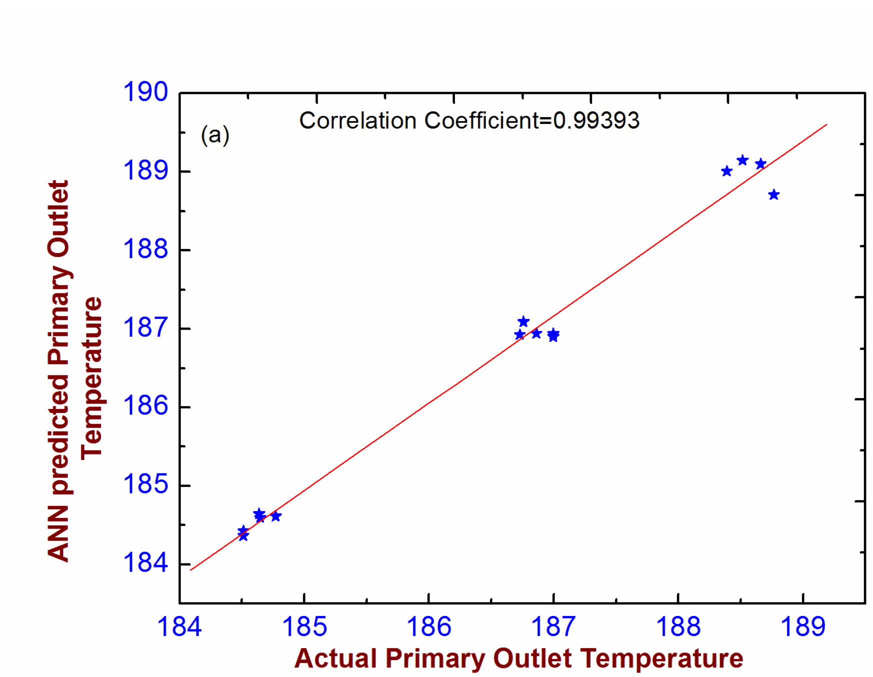


Fig. 3.11 (a) Scatter plot between actual and desired output values for primary outlet temperature of Intermediate Heat Exchanger

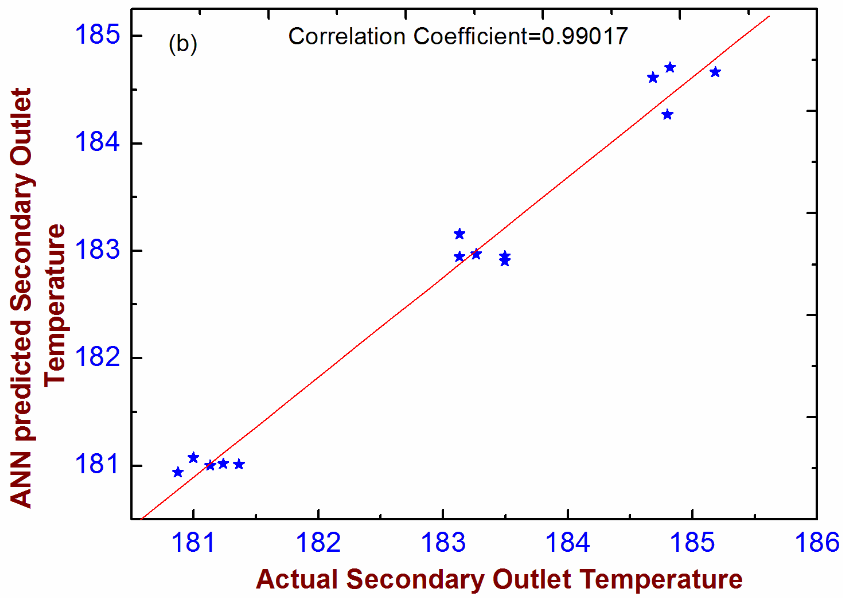


Fig. 3.11 (b) Scatter plot between actual and desired output values for secondary outlet temperature of Intermediate Heat Exchanger

3.3. Neural Network Modeling for Transient Conditions of Sodium-Sodium Heat Exchanger

A detailed non linear model is developed based on the physical parameter approach for a typical nuclear power plant. System behavior starting from steady state to transient is simulated. Simulation results are used as inputs to the nuclear reactor for parameter estimation. After training, the network is used to predict certain system parameters of the reactor for a number of different valued inputs. Artificial neural networks can be used as good estimator and as alternative to empirical models. The major drawback of the conventional systems is rather long computation time where as the artificial neural network developed in this project yields system behavior within a short computation time with acceptable accuracy [3.14-3.20].

3.3.1 Heat Exchanger Modeling using Back Propagation Algorithm and Radial Basis Function Algorithm

Artificial neural network model has been developed for sodium-sodium heat exchanger to study its behavior at transient conditions. Sodium-sodium heat exchanger is a very crucial component of the reactor which transfers heat from primary side to the secondary side of a fast reactor and the heat in turn is used for electricity generation. Using simulated data generated from the Quadratic Upstream Interpolation for Convective Kinetics (QUICK) code, a three layer artificial neural network is trained [3.21]. Artificial neural network can work efficiently where non-linear functions are used for evaluating the physical parameters at transient conditions with the help of powerful algorithms like back propagation algorithm [3.22-3.24]. The objective of this modeling is to evaluate primary and secondary sodium outlet temperatures for given mass flow rate in shell and tube side and respective inlet (primary and secondary) temperatures. The temperature prediction for severe unbalanced primary and secondary flow is predicted using Nodal Heat Balance (NHB) method [3.25]. Later it has been modified with the help of QUICK scheme and from the code using this scheme, the required input data has been generated [3.21]. The artificial neural network evaluates the outlet temperatures of heat exchanger during severely unbalanced flow conditions. Training and testing results show the successful modeling of plant dynamics of the reactor with improved accuracy. Artificial neural network performs better compared to the conventional method in estimating the outlet temperatures for normal and transient operating conditions. Further the model is trained with radial basis function neural network which gives comparatively better convergence than the multilayer back propagation network.

3.3.2 Quadratic Upstream Interpolation for Convective Kinetics (QUICK) Scheme

QUICK scheme is a higher order up winding numerical scheme which takes care of strong convective flow. It uses a three point upstream weighted quadratic interpolation for cell face values. The cell face values of fluxes are always calculated by quadratic interpolation between two upstream nodes and one downstream node. Since the scheme is based on a quadratic function, its accuracy in terms of the Taylor series truncation error is of third order on a uniform mesh [3.21]. It is an explicit scheme which uses discretization method which puts values in terms of nodes. The nodes themselves are treated as ordered or discrete values. With the help of QUICK scheme primary and secondary outlet and inlet temperatures have been evaluated for given mass flow rates of sodium-sodium heat exchanger in shell and tube side.

The energy balance diagram for a control volume is shown in Fig. 3.12.

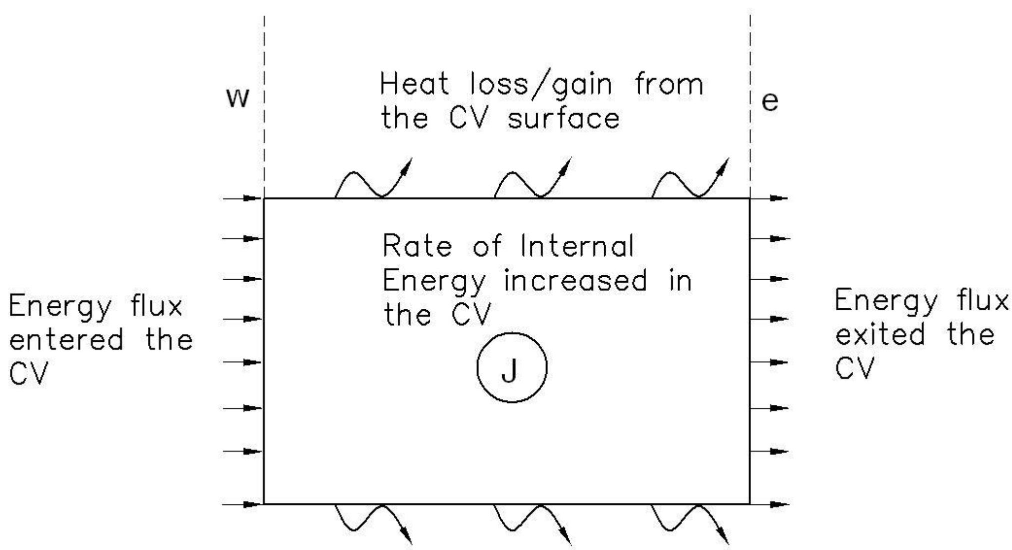


Fig. 3.12 Energy balance diagram for control volume

This is a 1-D model representing energy balance equations for the primary side and secondary side in equation 3.5 and 3.6.

Primary side:

$$\left(\Delta m_p\right)_{CV} \times \left(CP_p\right) \times \left(\frac{\partial T_p}{\partial t}\right)_J = \left(\dot{m} \times CP_p\right)\left(T_w - T_e\right) - U \cdot A\left(T_p - T_s\right) \qquad (3.5)$$

Secondary side:

$$\left(\Delta m_s\right)_{CV} \times \left(CP_s\right) \times \left(\frac{\partial T_s}{\partial t}\right)_J = \left(\dot{m} \times CP_s\right)\left(T_e - T_w\right) + U \cdot A\left(T_p - T_s\right) \qquad (3.6)$$

Where, $\left(\Delta m\right)_{CV}$ =Mass of a Control Volume (kg)

$\dot{m}$ =Mass flow rate (kg/s)

CP=Specific Heat (J/kg K)

U=Overall heat transfer coefficient (W/m^2 K)

A= Heat transfer area (m^2)

CP=Specific Heat (J/kg K)

T_P =Primary inlet temperature (^{0} C)

T_s = Secondary inlet temperature (^{0} C)

T_w =West face of control volume

T_e =East face of control volume

Using explicit scheme the above mentioned equations for next time step in terms of current time steps are given in equation (3.7a-3.7b).

$$\left(T_p\right)^{n+1} = \left(T_p\right)^{n} + A \times \left(T_w - T_e\right)^{n} - B\left(T_p - T_s\right)^{n} \qquad (3.7a)$$

$$\left(T_s\right)^{n+1} = \left(T_s\right)^{n} + C \times \left(T_e - T_w\right)^{n} + D\left(T_p - T_s\right)^{n} \qquad (3.7b)$$

where,

$$A=\left(\frac{\dot{m}_p \times CP_p}{(m_{CV}\times CP_p)+(m_{vessel}\times CP_{vessel})+0.5\times(m_{tube}\times CP_{tube})}\right)_J^n \times \Delta t \qquad (3.8a)$$

$$B=\left(\frac{U\cdot A}{(m_{CV}\times CP_p)+(m_{vessel}\times CP_{vessel})+0.5\times(m_{tube}\times CP_{tube})}\right)_J^n \times \Delta t \qquad (3.8b)$$

$$C=\left(\frac{\dot{m}_s \times CP_s}{(m_{CV}\times CP_s)+0.5\times(m_{tube}\times CP_{tube})}\right)_J^n \times \Delta t \qquad (3.8c)$$

$$D=\left(\frac{U\cdot A}{(m_{CV}\times CP_s)+0.5\times(m_{tube}\times CP_{tube})}\right)_J^n \times \Delta t \qquad (3.8d)$$

In equations (3.8a-3.8d), A,B,C,D are the coefficients of the equation taking into account of the thermal capacity of metal such as vessel and tube. Fig. 3.13 represents heat exchanger in terms of nodes. In this scheme the cell face temperatures T_w and T_e are discretized by fitting a quadratic polynomial between two upstream and one downstream node as given in equation 3.9.

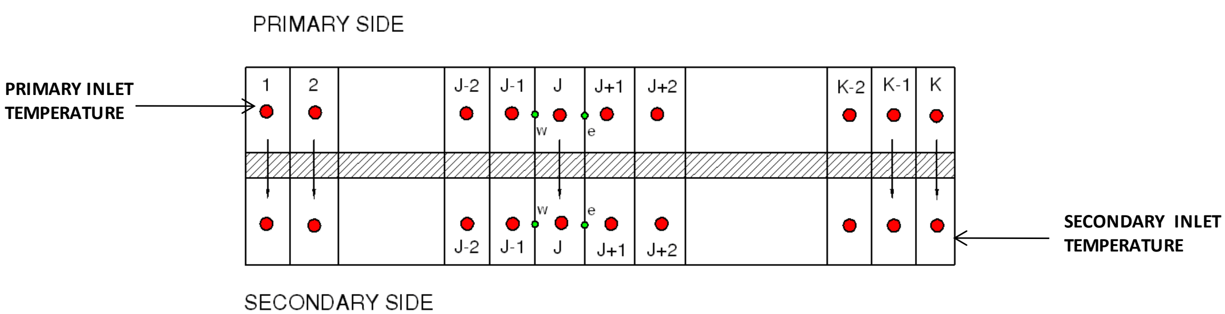


Fig. 3.13 Cross sectional view of sodium-sodium heat exchanger using the nodes

$$\left.\begin{aligned}
(T_p)_w &= \left(\frac{3}{8}\right)(T_p)_J + \left(\frac{3}{4}\right)(T_p)_{J-1} - \left(\frac{1}{8}\right)(T_p)_{J-2} \\
(T_p)_e &= \left(\frac{3}{4}\right)(T_p)_J + \left(\frac{3}{8}\right)(T_p)_{J+1} - \left(\frac{1}{8}\right)(T_p)_{J-1} \\
(T_s)_w &= \left(\frac{3}{4}\right)(T_s)_J + \left(\frac{3}{8}\right)(T_s)_{J-1} - \left(\frac{1}{8}\right)(T_s)_{J+1} \\
(T_s)_e &= \left(\frac{3}{8}\right)(T_s)_J + \left(\frac{3}{4}\right)(T_s)_{J+1} - \left(\frac{1}{8}\right)(T_s)_{J+2}
\end{aligned}\right\}_{J=3,K-2} \qquad (3.9)$$

Similarly for boundary nodes also, the temperatures can be determined. Subsequently substitution of the values of T_w and T_e from equation 3.9 in equation 3.7 yields,

$$\left.\begin{aligned}
(T_p)_J^{n+1} &= (1 - A - B) \times (T_p)_J^n + A \times (T_p)_{J-1}^n + B \times (T_s)_J^n \\
(T_s)_J^{n+1} &= (1 - C - D) \times (T_s)_J^n + C \times (T_s)_{J+1}^n + D \times (T_p)_J^n
\end{aligned}\right\}_{J=2,n-1} \qquad (3.10)$$

where, $(1 - A - B) > 0$ and $(1 - C - D) > 0$ is the criteria to make the solution stable.

Finally the outlet temperature of primary and secondary sodium can be estimated using linear interpolation and is given in 3.11a and 3.11b.

$$(T_p)_{out} = 1.5(T_p)_K - 0.5(T_p)_{K-1} \qquad (3.11a)$$

$$(T_s)_{out} = 1.5(T_s)_1 - 0.5(T_s)_2 \qquad (3.11b)$$

The required temperature and flow of sodium-sodium heat exchanger have been calculated by running the QUICK code which is used for training the artificial neural network model. The network is modeled using back propagation algorithm and then compared with radial basis function algorithm. As the input data is spread over a wide range, it has to be scaled down between zero to one which is also known as normalization. The normalization formula is given by equation 3.12.

$$d_{norm} = \frac{d}{d_{\max}} \qquad (3.12)$$

where, d_{norm} the normalized value of the input and d is the input parameter value and $d_{\max}$ is the maximum value of the respective parameters. The radial basis function algorithm is described below.

3.3.3 Radial Basis Function (RBF) Algorithm

Radial basis function (RBF) approach is designed as a curve fitting problem in high-dimensional space that provides the best curve fit for the applied dataset during training. A general radial basis function network consists of three layers, each of the layers having different utilities. The input layer consists of nodes where the datasets are applied (during training and testing). The second layer is the hidden layer of the network and unlike multilayer perceprton back propagation network radial basis function network contains a single hidden layer. The hidden layer applies a nonlinear transformation to the applied data to bring it to a hidden space of higher dimensionality. The set of functions that the hidden layer provides, constitute an arbitrary 'basis' for the input pattern when they are expanded into hidden space. These functions are called 'radial basis functions'. The third layer is the output layer of the network and it gives a linear transformation to the hidden space contents that forms the response of the network to the applied dataset.

In radial basis function the error vector generated is utilized to modify the network weights. The linear weights of the output units tend to evolve on a different 'time scale' compared to the nonlinear activation function of the hidden layer. Thus weights associated with the output layer adjust themselves rapidly than the weights of hidden layers due to a linear optimization strategy. The locations of centers are chosen randomly from the training dataset. The hidden layer basis functions are generally

Gaussian function whose standard deviation is fixed according to the spread at the centers.

The different radial basis functions used are:

Gaussian Functions:

$$\phi(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right) \qquad \text{width parameter } \sigma > 0$$

Multi-Quadric Functions:

$$\phi(r) = \left(r^2 + \sigma^2\right)^{1/2} \qquad \text{parameter } \sigma > 0$$

Generalized Multi-Quadric Functions:

$$\phi(r) = \left(r^2 + \sigma^2\right)^{\beta} \qquad \text{parameters } \sigma > 0,\ 1 > \beta > 0$$

Thin Plate Spline Function:

$$\phi(r) = r^2 \ln(r)$$

where, $\varphi(r)$ = basis function

σ =width parameter

r =center

β =multi-quadric function with values ...-3/2,-1/2,1/2,3/2....

The radial basis function is similar to the Gaussian density function which is defined by a 'centre' position and a 'width' parameter. It is a real valued function whose value depends only on the distance from its receptive field center μ to input x. The Gaussian function gives the highest output when the incoming variables are closest to the centre position and decreases monotonically as the distance from the centre increases. The width of the radial basis function unit controls the rate of decrease; for example, a small width gives a rapidly decreasing function and a large value gives a slowly decreasing function. When the distance between x and μ,(denoted as $\|x-\mu\|$), is smaller than the respective field width σ, the function has an appreciable value. Once the normalized center and training data are available, the width of the basis functions is found by the formula represented in equation 3.13.

$$\sigma = \frac{d_{\max}}{\sqrt{2 * nhn}} \qquad (3.13)$$

where σ = fixed width of all basis functions in the net

$d_{\max}$ = maximum distance between centre

nhn = number of hidden nodes

The non linear transfer function used here is Gaussian basis function whose output is inversely proportional to the distance from the center of the neuron and that is stated in equation 3.14.

$$Z_j = \exp\left[-\frac{\| x - \mu_j \|^2}{2\sigma_j^{\,2}} \right] \qquad (3.14)$$

where, x is the input vector and μ_j and σ_j are the center and width of the basis function and Z_j is corresponding activation function. In multilayer perceptron, the network is trained with single global supervised algorithm, whereas radial basis function is trained one layer at a time with first layer unsupervised.

The output expression for the network is computed by the formula given in equation 3.15 with $y = [y_1, y_2, \ldots\ldots, y_k]$ and y_i as the output of the ith neuron given by

$$y_i = \sum_{i=1}^{nhn} w_{ji} * Z_j + b \qquad \text{for i=1,2,...,k} \quad (3.15)$$

where,

nhn = number of hidden layer nodes (radial basis function)

Z_j = output value of node in hidden layer for the i_{th} incoming pattern

w_{ji} = weight between j_{th} radial basis function unit and i_{th} output node

b = biasing term

Radial basis function neural networks are classified as universal approximators, as this type of network structure, using basis functions for mapping the network neurons can approximate virtually any function of interest to any desired degree of accuracy, provided sufficient number of neurons are represented in the hidden layers of the network.

When the testing data is given to the algorithm, using the above weight parameters, the neural network will be able to generate outputs almost matching with the desired outputs. This can be represented as ability to generalize. The coding has been performed in C. Out of 77 sets of data generated from the QUICK code for prediction of temperature of IHX, the network has been trained with 70 sets. A set of seven distinct test cases have been chosen within the range of inputs already trained, for testing the network. The time taken to generate the data set using QUICK code is 10 minutes.

3.3.4 Results and Discussion

Different trials have been carried out using back propagation algorithm in the training phase, to get the optimal values for different number of hidden nodes and learning rates. Fig. 3.14 shows mean square error parameter with respect to the number of iterations for various learning rates and number of hidden nodes. At first the value of learning rate is varied keeping number of hidden nodes constant. Then the number of hidden nodes is varied keeping learning rate constant. The optimal learning rate and number of hidden node are $nhn = 4$ and learning rate $lr = 0.01$ showing least mean square error of 9.5×10^{-5} for 10^5 iterations.

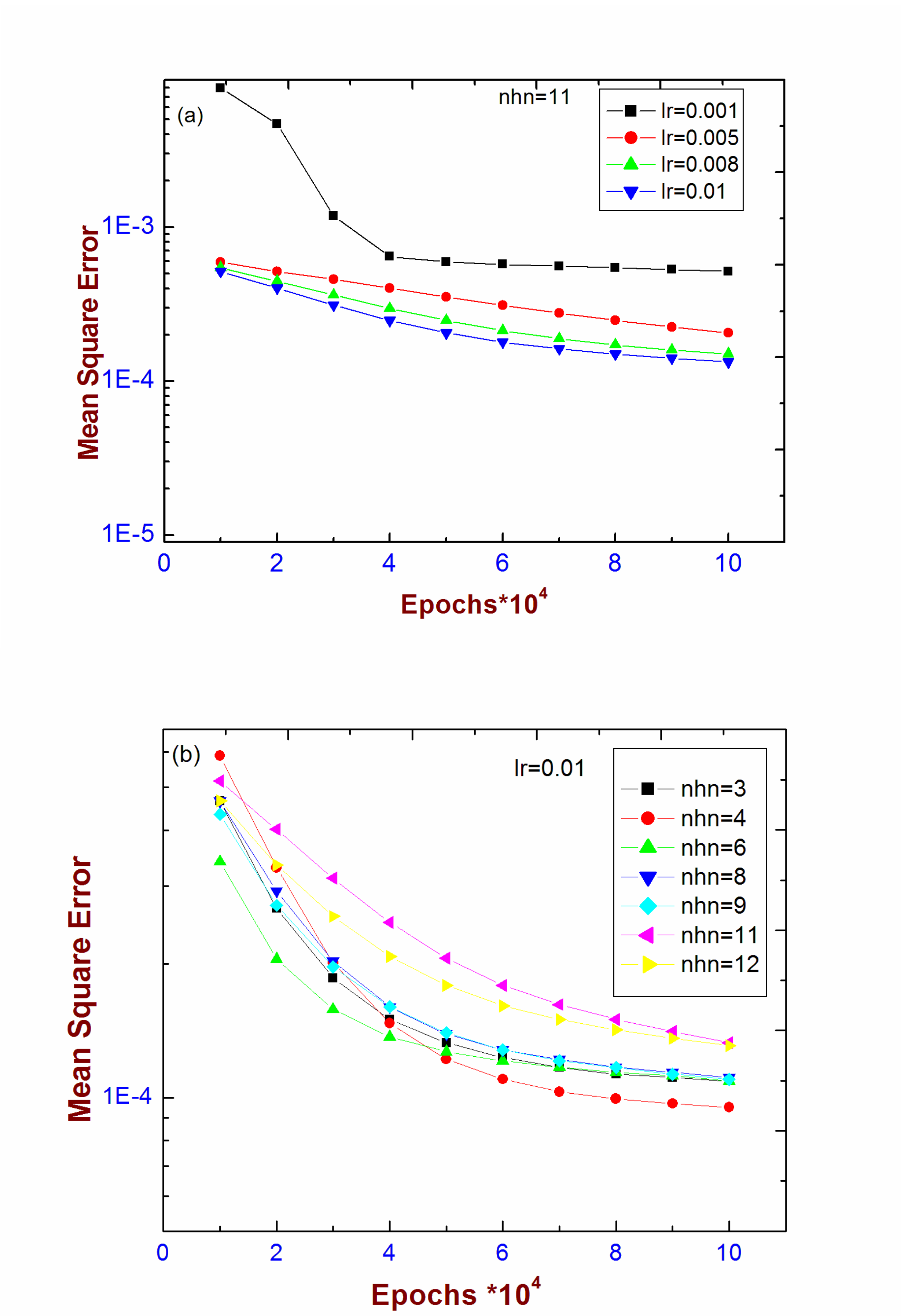
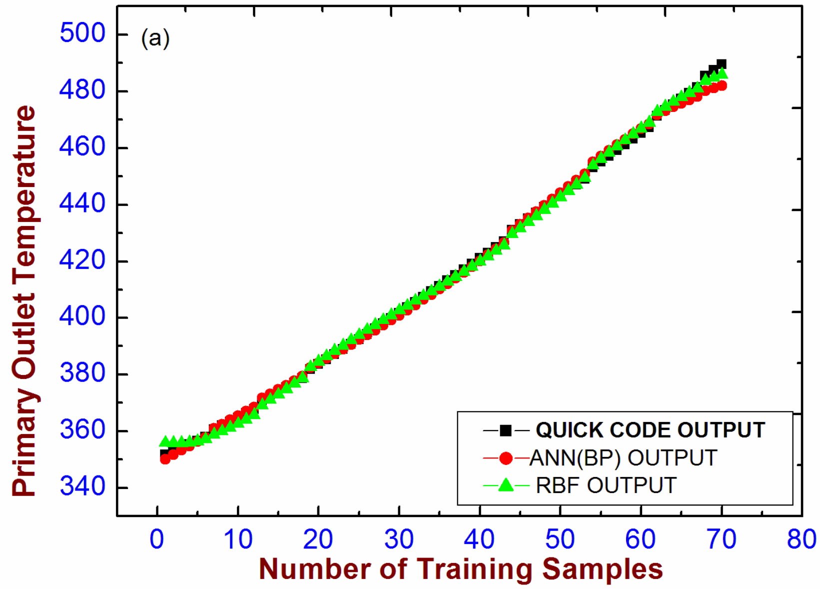


Fig. 3.14 Mean square error with respect to number of epochs for (a) various learning rates (b) number of hidden nodes for Intermediate Heat Exchanger



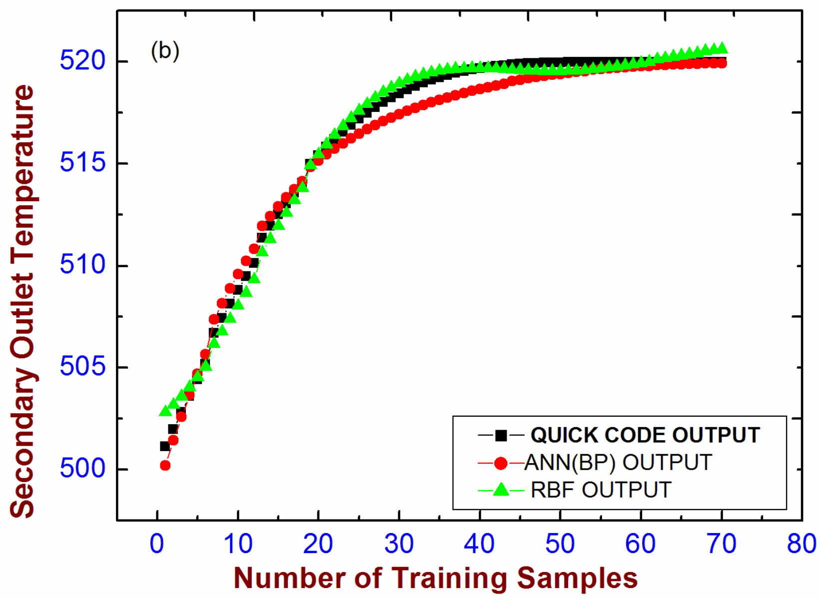
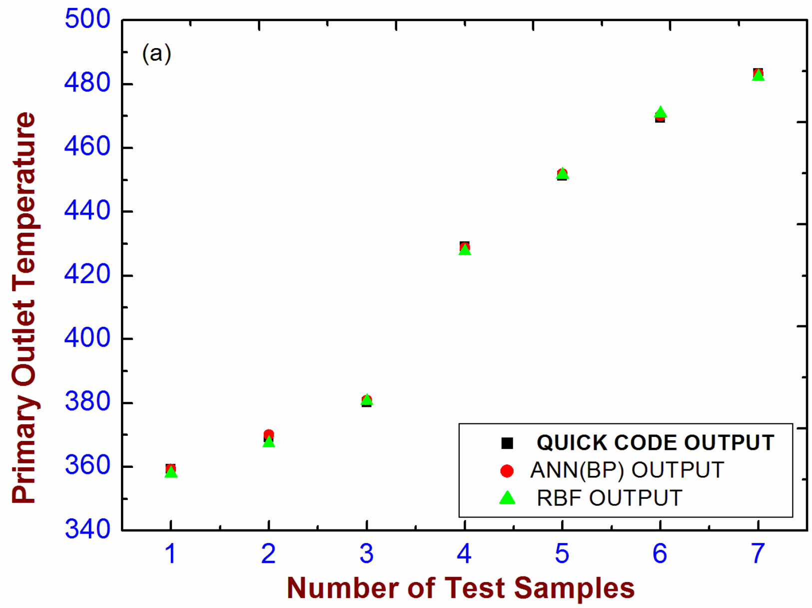


Fig. 3.15. Estimation of (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger for training sample index

After fine tuning, the network is further trained for 10^6 iterations using back propagation algorithm. Finally the network is again trained using the radial basis function algorithm. The number of centers used for radial basis function network is four. Both back propagation algorithm and radial basis function algorithm are employed. Variation in the graph plotted between the total number of training sample sets of primary and secondary outlet temperatures for desired and actual outputs are presented in Fig. 3.15. It can easily be seen that the actual outputs achieved from back propagation algorithm and radial basis function algorithm agree with the desired output.

The training results show that radial basis function algorithm is much faster compared to the standard back propagation algorithm as the network got converged to 8.9×10^{-6} in 10^4 iterations itself. In Fig. 3.16, the graph for testing samples using the back propagation algorithm and radial basis function algorithm is plotted. The testing samples are not present in the training set but these are taken from the range of training samples. This is the validation phase. The graph shows that the testing samples for desired and actual output of primary and secondary outlet temperatures overlap with each other and their difference (set to be ± 2 °C) is also not very large. From the results given above it can be observed that, compared to the conventional methods used for the prediction of intermediate heat exchanger temperature, the standard back propagation algorithm shows better results and compared to standard back propagation algorithm, radial basis function algorithm shows better convergence.



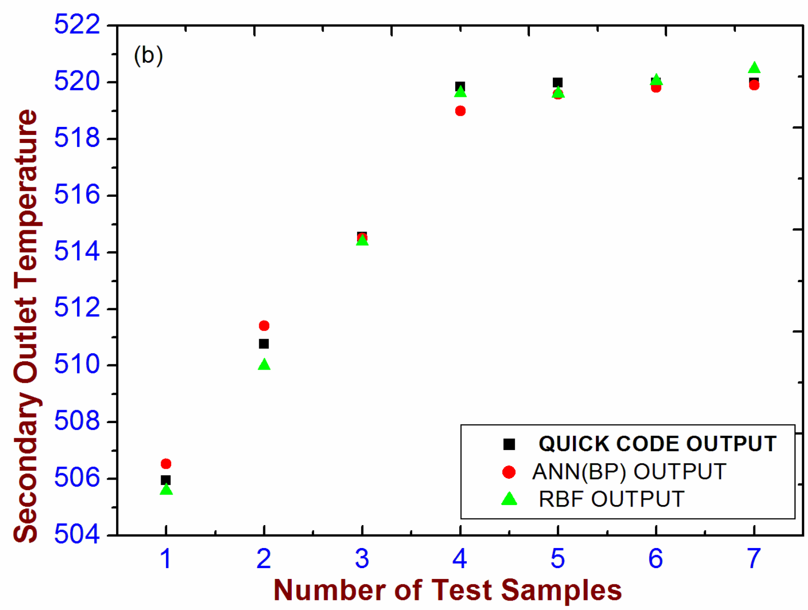


Fig. 3.16 Estimation of (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger for test sample index

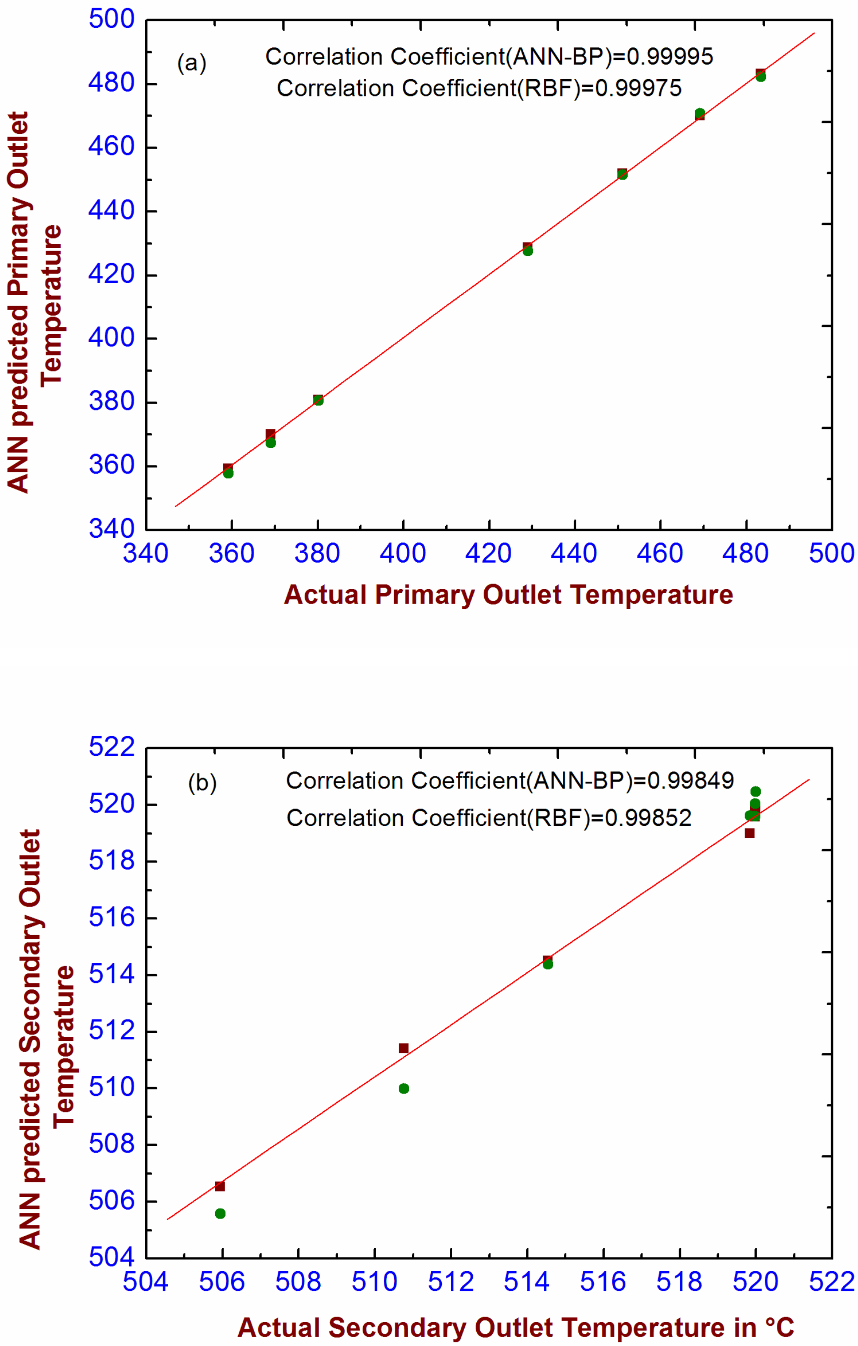


Fig. 3.17 Scatter plot between desired and predicted output values for QUICK code, back propagation and radial basis function for (a) primary outlet temperature (b) secondary outlet temperature of Intermediate Heat Exchanger

Fig. 3.17 depicts scatter plots between desired and actual outputs for testing samples. The data treatment yielded the correlation coefficient of 0.99995 (for standard back propagation network), 0.99975 (for radial basis function network) for primary outlet temperature and 0.99849 (for standard back propagation network) and 0.99852 (for radial basis function network) for secondary outlet temperature. The correlation approaching to unity indicates that the desired and the actual data lie on a 1:1 straight line when plotted against each another. The standard back propagation has certain drawbacks as it can get trapped in local minima instead of reaching the global minima [3.26]. This is overcome by radial basis function algorithm which uses Gaussian function as transfer function. The time taken to train the network using radial basis function is three minutes on a typical personal computer system (2.66 GHz Intel Core 2 Duo processor). Once trained, the network gives the output in few milliseconds with expected accuracy, which is set as the difference between expected output and desired output to be $\pm 2\,^{0}$C.

3.4. Development of Artificial Neural Network Model for Prediction of Reactor Power Using Neutronics Subsystem of PFBR

The ability to model the dynamic systems has become a fundamental aspect for the safe and economically competitive operation of modern nuclear power plants. System design and validation, fault detection and control are among the tasks in modern engineering which rely on the ability of identifying and modeling dynamic systems. The knowledge of the state of a system during each instant of its operation is a desirable feature of all nuclear power plants. This requires quick calculation of the physical parameters which are difficult to achieve in a conventional, complex code dynamics model, because it consumes more time. The main objective of this study is to utilize the ability of neural network to create data-driven representations of the underlying dynamics of reactor subsystem with less reliance on accurate mathematical or physical modeling. The subsystem taken for study is neutronics subsystem of PFBR [3.27-3.33].

3.4.1 Neutronics Model

Neutronics model is an important subsystem of PFBR simulator which simulates the neutron flux and power generation by the reactor core. This model was originally developed based on the one dimensional plant dynamic analysis code called DYANA-P [3.34].

It uses point kinetics equation to calculate the reactor power above criticality. The objectives of this model being:

- It should generate signals corresponding to reactor power, reactivity and period.
- It should give necessary trip parameters to shutdown system's simulation model and should operate in all stages of reactor such as startup and power operation.
- In order to train the operator to handle possible incidents and malfunctions, the neutronics model should be able to simulate the following transient conditions
    - Uncontrolled withdrawal of one control rod
    - Reactor power raising / lowering
    - Neutronic transients arising due to change of system temperature caused due to other events
    - Exercising facility for control and safety rods

The basic input to the model is the reactivity change, which is calculated, based on the position of the control and safety rods. The neutronics model should generate the signals corresponding to power, reactivity and reactor period in all stages of reactor

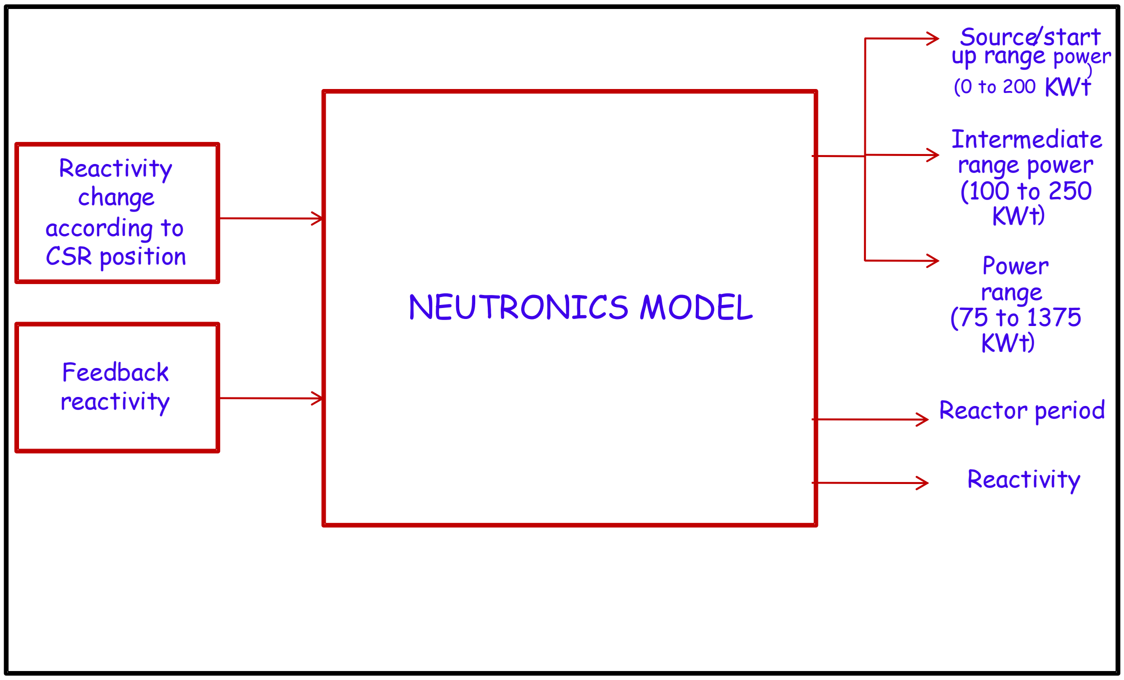


Fig. 3.18 Schematic of Neutronics Model

operation such as shutdown, start up and power operation. Fig. 3.18 represents the neutronics model of PFBR.

3.4.2 Neural Network Architecture for Neutronics System

Multilayer perceptron with feed forward neural network model is developed using back propagation algorithm to predict process parameters of neutronics subsystem. The multilayer perceptron is having three layers, one input layer, one hidden layer and output layer. The neural network model is having ten input nodes (including bias) denotes nine control and safety rod positions and the output layer has only one output node and that is reactor power. Artificial neural network with adaptive learning finds the weights and consequently the model is changed by updating the weights. Back propagation algorithm starts with a discrete jump, where the step size is adjusted by the number of hidden nodes and learning rate parameter. The iterative process is continued until the convergence is achieved.

3.4.3 Results and Discussion

100 sample sets of data has been taken out of which 95 samples are used for training and rest 5 are used for testing.

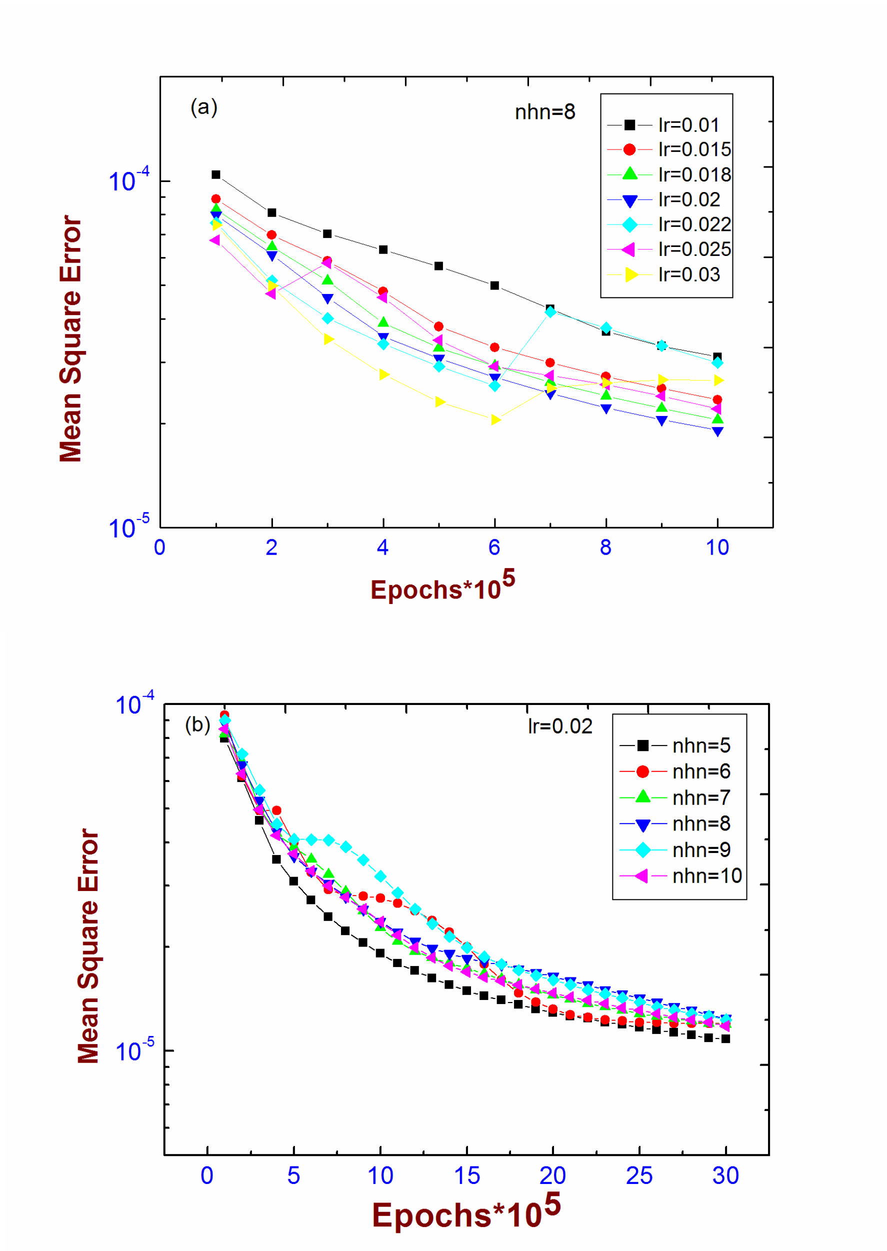


Fig. 3.19. Mean square error with respect to number of epochs for (a) various learning rates (b) number of hidden nodes for Neutronics System

The data used for training and testing is obtained from the simulator using DYANA-P code [3.34]. The dataset for training is presented in Table-C of Annexure 2 and the output obtained using the test data are compared with the desired output and are presented in Table D of Annexure 2. The training time is 3 hours and 20 minutes for 2×10^{7} iterations on a computational system (2.66 Gigahertz Intel Core 2 Duo PC). Different trials have been carried out to get the optimal parameters (the process of fine-tuning) in the training process of neural network and results are summarized in Fig. 3.19. It shows that by varying the number of hidden layers, learning rate and the number of iterations, it is possible to reduce the mean square error (difference between desired and actual output).

After training the artificial neural network with learning rate= 0.02 and number of hidden nodes= 5 with 2×10^{7} iterations, the mean square error value= 4.5×10^{-6} has been obtained. The parameters have been adopted as optimal. Fine tuning of the neural network is represented in Fig. 3.19.

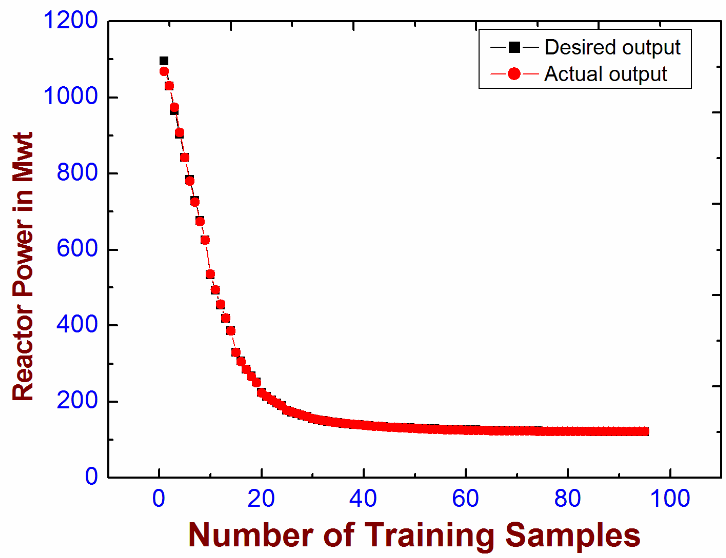


Fig. 3.20 Graph plotted between reactor power in Megawatt and the number of training sample index for desired and actual outputs of Neutronics System

Fig. 3.20 shows the graph plotted between the total number of training sample set and the corresponding outputs. It is seen from the graph that the actual output agrees with the desired output. Validation of the back propagation algorithm can be proved by testing a set of data which is within the range of training sample data. Here five sets of testing data have been used and it has been found that the trained neural network model is able to generalize and give a meaningful output for a given data within the range. Fig. 3.21 represents the graph showing the test results for desired and actual reactor power and their difference.

The artificial neural network model is tested with random inputs within the trained range. It has been found that the results produced are matching with expected outputs and the difference (± 2 Megawatt) between the desired and actual output is well within the permissible limit.

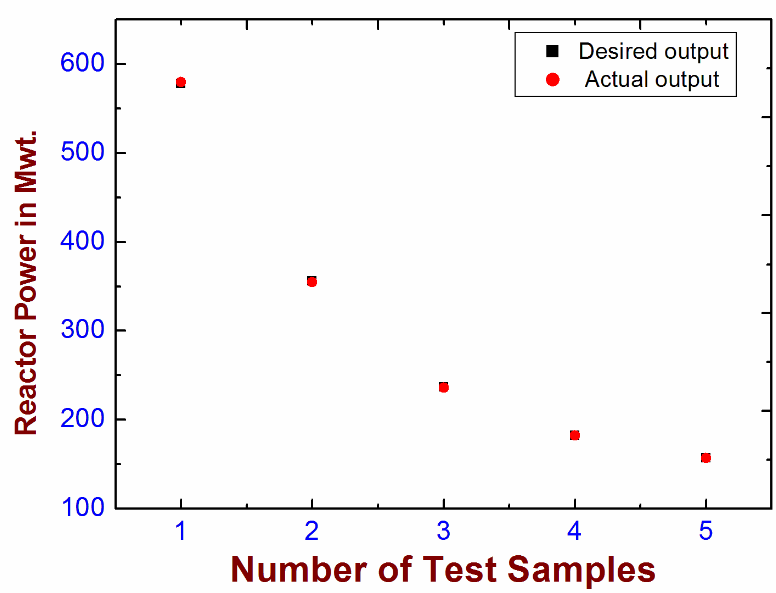


Fig. 3.21 Graph plotted between reactor power in Megawatt and the number of test sample index for desired and actual outputs of Neutronics System

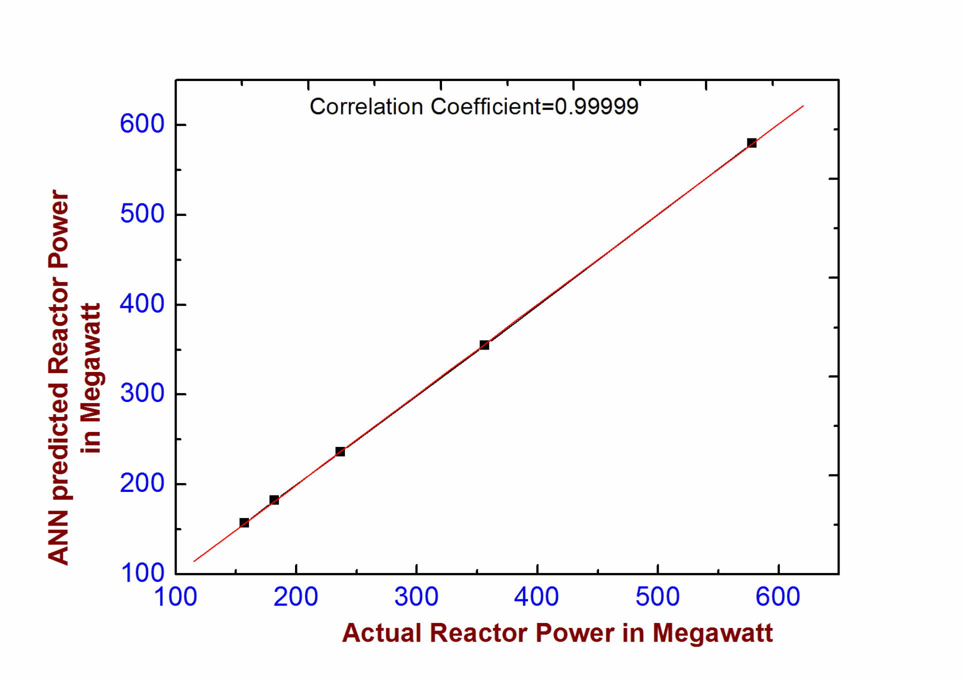


Fig. 3.22 Scatter plot between actual and back propagation neural network predicted values for reactor power in Megawatt of Neutronics System

The output data for one sample is obtained in 23.34 seconds from the simulator. Whereas, artificial neural network is able to obtain the output in few milliseconds giving accurate predictions. Fig. 3.22 gives the scatter plot between the actual and desired reactor power outputs. From the fit the correlation coefficient is found to be 0.9999 and which indicates a good fit between the desired and actual output.

3.5 Artificial Neural Network Model for Prediction of Reactor Power of Neutronics Subsystem of PFBR Using Variants of Back Propagation Algorithms

In this study, an artificial neural network model has been developed and trained with various back propagation learning algorithms to estimate neutronics power and model the plant dynamics. The best performing algorithm in terms of faster convergence has been identified among the variants of back propagation network (BPN). Out of 893 sets of data obtained from the neutronics mathematical model, 883 datasets are selected for training the network. Ten samples which are not part of the training set are used for prediction. The learning rate factor used in weight optimization formula is standardized based on the experience gained from our earlier artificial neural network simulation work and set as 0.7. The optimal number of hidden nodes is found to be 8 after carrying out trials with various hidden nodes starting from 5 to 10. After optimizing the key parameters, the network is further trained with different variants of back propagation algorithms to find the optimal results.

BIKAS (Bhabha Atomic Research Centre – Indian Institute of Technology Kanpur-Artificial Neural Networks – simulator) has been used for carrying out the training and testing. It is a general purpose neural network simulator written in JAVA. BIKAS has been used for training the network with various back propagation learning algorithms with different weight optimization schemes. The datasets are normalized using the simulator in order to scale down the entire range of data into 0.1-0.9 before training. Different weight optimization algorithms namely standard back propagation, back propagation with momentum in pattern mode, back propagation with momentum in batch mode, quick propagation and resilient back propagation have been applied and results are analyzed. A brief explanation of each of the algorithms is given below [3.35].

3.5.1 Back Propagation Algorithm and its Variants

3.5.1.1 Standard Back Propagation (BP) Algorithm with Pattern Mode

In standard back propagation algorithm, the inputs are applied to the input layer of the network. The random weights are then applied to the connection links between input layer and hidden layer neurons. The weights are in turn multiplied with the inputs and the summed up result is then applied with an activation function to calculate the output for hidden layer. In standard back propagation algorithm with pattern mode, the weights are updated after each input pattern is applied [3.36].

3.5.1.2 Back Propagation Algorithm with Momentum and Pattern Mode

In case of back propagation algorithm with pattern mode and momentum, the momentum factor is used in order to improve the local minima problem. This method takes the error estimate from the result in presenting just the current pattern. It introduces noise into the learning process and it is known that an accurate calculation of the error gradient is possible only when all training patterns have been presented. The momentum term also avoids the oscillations of the error curve. The momentum value used here is 0.8.

3.5.1.3 Back Propagation Algorithm with Momentum and Batch Mode

Back propagation algorithm with momentum and batch mode learning, explains that the weight update is done after the entire training set is applied to the input layer. It takes the total training error over all the patterns into account. The momentum speeds up convergence while training a feed-forward neural network.

3.5.1.4 Quick Propagation Algorithm

The quick propagation algorithm requires the computation of the second order derivatives of the error function. Everything proceeds as in standard back-propagation, but for each weight, a copy of the value of $\partial E / \partial w(t-1)$, the error derivative computed during the previous training epoch, along with the difference between the current and previous values of this weight is saved. The $\partial E / \partial w$ value for the current training epoch is also available at weight-update time. It assumes the error to be locally quadratic and attempts to jump in one step from the current position directly in to the minimum of the parabola. The weight update formula is represented in equation 3.16.

$$\Delta w(t) = \frac{s(t)}{s(t-1) - s(t)} \Delta w(t-1) \qquad (3.16)$$

where $s(t)$ and $s(t-1)$ are the current and previous values of the error gradient vector $\partial E / \partial w$, $\Delta w(t)$ is the weight change and $\Delta w(t-1)$ is weight change in previous step [3.37]. This new weight update value is only a crude approximation to the optimum value for the weight, but when applied iteratively, effective results can be obtained.

3.5.1.5 Resilient Back Propagation Algorithm

Resilient back propagation is known to provide faster local adaptation of weights and biases without sacrificing the accuracy of the network designed. This accelerated back propagation procedure achieves improved performance by adapting weights and biases on the basis of the polarities of the partial derivatives of $\partial E / \partial w_{ij}$ only, not considering the magnitude of $\partial E / \partial w_{ij}$. One of the distinguishing features of resilient back propagation is that here all the weights, whether in hidden or output layer, are similarly affected after each iteration. This is in sharp contrast to most of the other variants of back propagation, where weights in output layer are more sensitive to $\partial E / \partial w_{ij}$ than the weights in hidden layer. A uniform updating of all weights and biases after each iteration in resilient back propagation, ensures potential acceleration in training philosophy [3.38].

The basic principle of resilient back propagation is to eliminate the harmful influence of the size of the partial derivative on the weight step. As a consequence, only the sign of the derivative is considered to indicate the direction of weight step. A step size Δ_{ij} i.e., the update amount of weight W_{ij}, is adapted for each weight individually. The main difference of resilient back propagation compared to other techniques is that the step sizes are independent of the absolute value of the partial derivatives. The weight formula is shown is equation 3.17 and 3.18 [3.39].

$$\Delta_{ij}(t)=\begin{cases}\eta+*\Delta_{ij}(t-1) & if\ \frac{\partial E}{\partial w_{ij}}(t)*\frac{\partial E}{\partial w_{ij}}(t-1)>0\\ \eta-*\Delta_{ij}(t-1) & if\ \frac{\partial E}{\partial w_{ij}}(t)*\frac{\partial E}{\partial w_{ij}}(t-1)<0\\ \Delta_{ij}(t-1) & otherwise\end{cases}\qquad(3.17)$$

Where Δ_{ij} represents the new update value that solely determines the weight-update. $\partial E/\partial w_{ij}(t)$, $\partial E/\partial w_{ij}(t-1)$ are partial derivatives of error for current and previous steps. The adaption rule is as follows. Every time a partial derivative of the corresponding weight W_{ij} changes its sign, which indicates the last update is too big and the algorithm has jumped over a local minimum, the update value Δ_{ij} is decreased by the factor η^{-}. If the derivative retains its sign, the update value is slightly increased by factor η^{+}, in order to accelerate convergence in the shallow region. Once the update value for each weight is adapted, the weight update can be represented as follows: if the derivative is positive (increasing error), the weight is decreased by its update value, if the derivative value is negative, the update value is added.

$$\Delta w_{ij}(t)=\begin{cases}-\Delta_{ij}(t) & if\ \frac{\partial E}{\partial w_{ij}}(t)>0\\ +\Delta_{ij}(t) & if\ \frac{\partial E}{\partial w_{ij}}(t)<0\\ 0 & else\end{cases}\qquad(3.18)$$

where $\Delta w_{ij}(t)$ determines the change in weight parameter. The value of two learning rate factors η^{+} and η^{-} are 1.2 and 0.5 respectively. The value of Δ_0 is 0.07, $\Delta_{\max}$ is 50 and $\Delta_{\min}$ is 0.001.

3.5.2 Results and Discussion

Back propagation learning algorithm and its variants are applied to model the neutronics system using artificial neural network approach. Table 3.1 below shows the mean square error values for different algorithms with respect to number of epochs.

Table3.1. Mean Square Errors for various algorithms with respect to number of epochs

Algorithms	Epochs	Mean Square Error
Back Propagation (Pattern mode)	10000	2.3×10^{-6}
Back Propagation (Pattern mode, Momentum)	10000	2.1×10^{-6}
Back Propagation (Batch mode with Momentum)	10000	7.8×10^{-4}
Quick Propagation	10000	8.3×10^{-4}
Resilient Back Propagation	1800	4.9×10^{-7}

From the results shown in the above table, it can be seen that the resilient back propagation algorithm is showing faster convergence and yields satisfactory results with less number of iterations. Fig. 3.23 shows the training results of resilient back propagation algorithm for 99 training samples. The graph explains the overlapping of the conventional and artificial neural network results for neutronics power with minimal difference between each other.

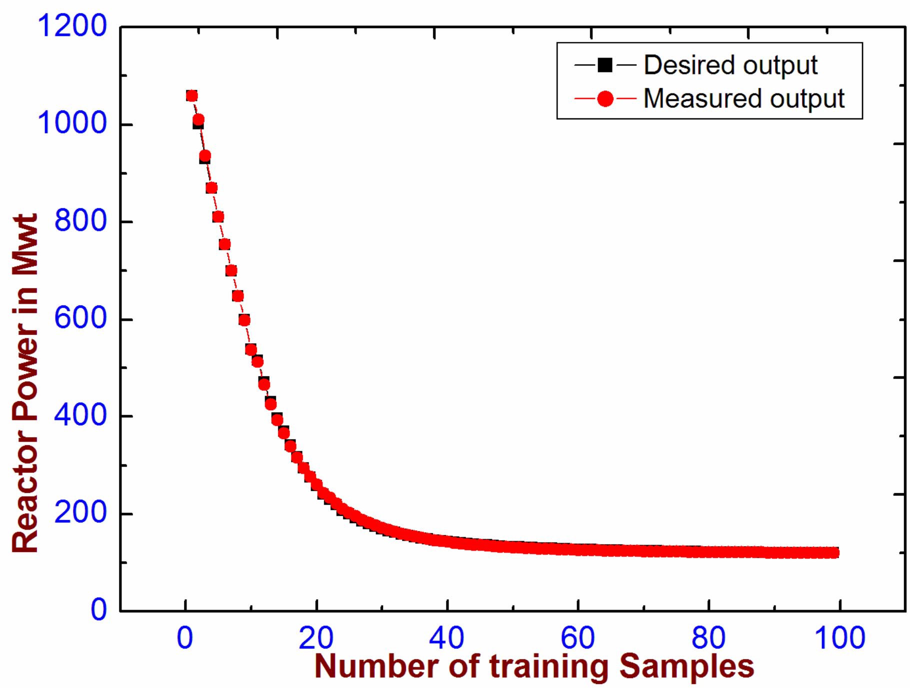


Fig. 3.23 Graph plotted between reactor power in Megawatt and the number of training sample index for desired and actual outputs of Neutronics System

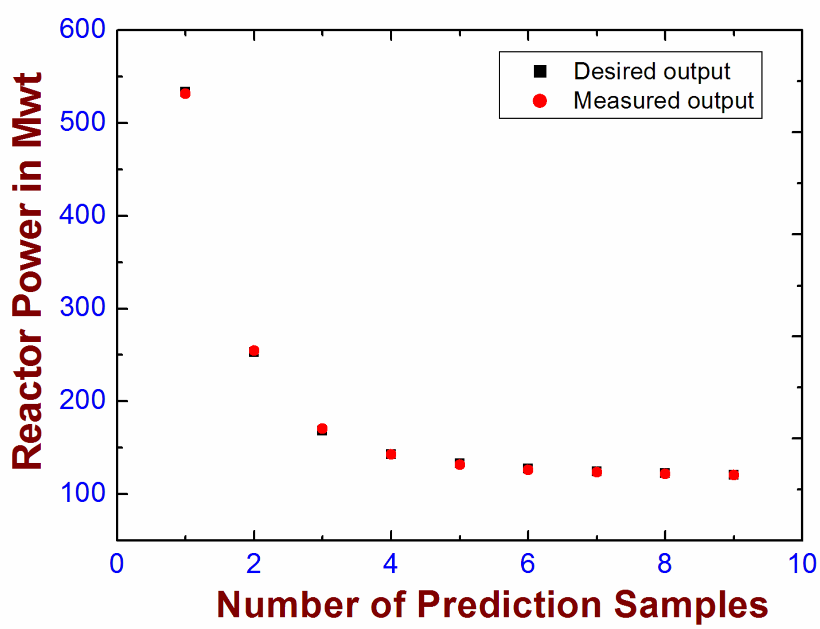


Fig. 3.24 Graph plotted between reactor power in Megawatt and the number of test sample index for desired and actual outputs of Neutronics System

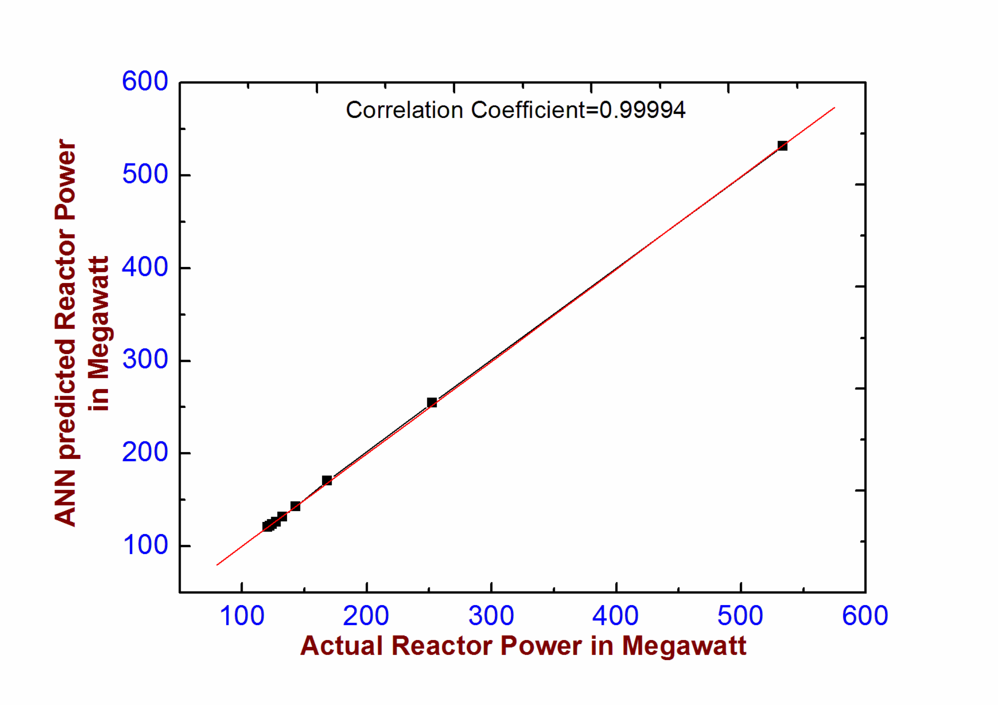


Fig. 3.25 Scatter plot between desired and actual predicted values for reactor power in Megawatt of Neutronics System

During validation phase after testing, ten samples, not used in training the data, are used for prediction. The prediction results shown in Fig. 3.24 are in excellent agreement with the results obtained from conventional model. Fig. 3.25 shows the scatter plot between desired and actual output. The correlation coefficient is found to be 0.99994 confirming the fit to be good and predicted values are in good agreement.

3.6. Conclusion

Neural network models have fast and accurate prediction capability for plant conditions starting from steady state to transient state covering a wide range of data sets and can be used for plant dynamic analysis. It also helps in studying the behavioral model of the subsystems at steady state and transient states.

- The first two cases are implemented using standard back propagation algorithms for temperature parameter estimation of intermediate heat exchanger.

- In the third case, the network is first trained and tested using back propagation algorithm and is again trained with radial basis function algorithm. The results of radial basis function algorithm are compared with standard back propagation algorithm. It is seen that the radial basis function algorithm is better as compared to standard back propagation network having faster convergence. From the results of the work carried out, it is found that the neural network model is able to generalize and produce satisfactory results for a wide range of data with faster response than the conventional methods.
- In the fourth case, standard back propagation algorithm is implemented for process modeling of neutronics system.
- In the fifth case, the back propagation algorithm along with variants is applied for process modeling of neutronics system. Among those algorithms, resilient back propagation algorithm converged faster with less number of epochs.

3.7. References

3.1 Vasudevan M., Rao B.P.C, Venkatraman B., Jayakumar T., Baldev Raj, *Journal of Materials Processing Technology,* 169 2005 396- 400

3.2 Attieh I.K., Gribok A.V., Hines J.W., *Nuclear Engineering and Design*, 207 2001 207-221

3.3 Mehrdad B., Mohammad Ghofrani B., Lucas C., *Annals of Nuclear Energy*, 29 2002 1225- 1240

3.4 Malik N., *National Conference on Unearthing Technological Developments and their Transfer for Serving Masses GLA ITM*, Mathura, India, 2005

3.5 Choi Y J., Kim H K., Back W P., Chang S H., *Information Sciences- Informatics and Computer Science: An International Journal,* 60 2004 207-224

3.6 Vallejo I., Barrio F., *Progress in Nuclear Energy*, 29 1995 371-373

3.7 Athinarayanan S., Jehadeesan R., *Seminar on Automation in Process Engineering and Manufacture (SA PREM 97), IE (I)*, IGCAR, kalpakkam, 1997

3.8 *Internal report on Primary Sodium Main Circuit System Manual*, Report No. FBTR/FRG/32000-DN-S-RS-06

3.9 James Stich T., Spoerre Julie K., Velasco T., *Journal of Industrial Technology,* 16 2000

3.10 Khalafi H., Terman M.S., *Progress in Nuclear Energy,* 51 2009 135-140

3.11 Khajavi M.N., Menhaj M.B., Suratgar A.A., *Annals of Nuclear Energy,* 29 2002 751-760

3.12 Nabeshima K., Suzudo T., Seker S., Ayaz E., Barutcu B., Turkcan E., Ohno T., Kudo K., *Progress in Nuclear Energy* , 43 2003 397-404

3.13 Akkurt H., Colak U., *Annals of Nuclear Energy,* 29 2002 2087-2103

3.14 Zhao B., Su Y., *Chemical Engineering Research and Design,* 88 2009 606-613

3.15 Lu S., Hogg B.W., *Electrical Power and Energy Systems*, 22 2000 67-78

3.16 Gupta P.K., Kumar P.A., Kaul A., Pandey G.K., Padmakumar G., Prakash V., Anandbabu C., , *Proc. National Seminar on NDE*, 2006

3.17 Ridluan A., Manic M., Tokuhiro A., EBaLM-THP- *Nuclear Engineering and Design*, 239 2009 308-319

3.18 Moon S. K., Chang S. H., *Nuclear Engineering and Design,* 150 1994 151–161

3.19 Seker S., Turkcan E., Aayz E., Barutcu B., *Annals of Nuclear Technology*, 30 2003 1777-1791

3.20 Koo Y., Oh J., Lee B., Tahk Y., Song K., *Journal of Nuclear Materials,* 404 2010 33-43

3.21 *Internal Report on Primary Sodium Main Circuit System Manual*, Report No. FBTR/FRG/32000-DN-S-RS-06

3.22 Hessel G., Schmitt W., Van Der Vorst K., Weiss F.P., *Progress in Nuclear Energy,* 34 1999 173-183

3.23 Hwang, B.C., *IEEE World Congress on Computational Intelligence*, 4 1994 2580-2584

3.24 Faria E.F., Pereira C., *Annals of Nuclear Technology*, 30 2003 603-613

3.25 *Internal Report on Implementation of QUICK Scheme for Modeling of Na-Na Shell & Tube Heat Exchanger,* Report No. FRTG/ETHS/99502/DN/3002.

3.26 Burse K., Manoria M., Kirar V. P. S., *Communications in Computer and Information Science,* 147 2011 67-73

3.27 Boroushaki M., Ghofrani M.B., Lucas C., *Annals of Nuclear Energy,* 29 2002 1225-1240

3.28 Nandi B.K, Moparthi A., Uppaluri R., Purkait M.K., *Chemical Engineering Research and Design,* 88 2010 881-892

3.29 Montes J.L., Francois J L., Ortiz J.J., Campo C., Perusquia R., *Annals of Nuclear Energy,* 36 2009 121-130

3.30 Mazrou H., Hamadouche M., *Progress in Nuclear Technology,* 44 2004 263-275

3.31 Mandal S., Sivaprasad P.V., Venugopal S., Murthy K.P.N., *Applied Soft Computing,* 9 2009 237-244

3.32 Smets, H.M.G., Bogaerts, W.F.L, *Fuzzy Sets and Systems,* 74 1995 153-162

3.33 Zainuddin Z., Mahat N., Hassan Y.A., *World Academy of Science, Engineering and Technology,* 1 2005 79-82

Chapter 4

Event Identification in Fast Breeder Reactor Subsystems

This chapter provides detailed description of some of the events associated with the Prototype Fast Breeder Reactor (PFBR) subsystems and identification of those events using the artificial neural network algorithms. The events related to neutronics system: uncontrolled withdrawal of control and safety rod and primary sodium circuit: primary sodium pump trips, primary sodium pump seizure, have been identified using the standard back propagation algorithm. An Event Detection System is proposed for integrating and identifying four events using the variants of back propagation algorithms. The best suited algorithm has been chosen from among them depending on the early convergence criteria.

4.1. Neural Network Based Event Identification (Uncontrolled Withdrawal of Control and Safety Rod) for Neutronics System in PFBR

The analytical methods used for the precise modeling of behavior of nuclear reactor are very complex and sometimes difficult to achieve. Thus, practical approach for event identification of dynamic systems like nuclear reactor is to use an empirical model with a nonlinear structure. Among empirical models, neural networks have been widely considered. The method of representing dynamic systems by vector differential equations is currently well established in system theory and applied to a fairly large class of systems. For example, in a multi-input, multi-output (MIMO) system of order n with $u(k)$ representing the inputs, $x(k)$ the state variables, and $y(k)$, the output of the system is given by the differential equations 4.1 and 4.2.

$$x(k+1) = \Phi[x(k), u(k)] \tag{4.1}$$

$$y(k) = \Psi[x(k)] \tag{4.2}$$

When the functions Φ and ψ are unknown, the problem of identification of the unknown system arises. The use of information gathered to complete, part or all of an unknown system model (either physical or empirical) is known as system (model) identification. The identification objective is to construct a suitable identification model which when subjected to the same input $u_P(k)$ as the plant produces an output $\hat{y}_P(k)$ which approximates $y_P(k)$. Fig. 4.1 below shows the method of identification of a multi-input, multi-output system. The choice of the identification model (i.e., its parameterization) and adjusting the model parameters based on the identification error $e_i(k)$ constitute two principal parts of the identification problem. The ability of neural networks to approximate large classes of nonlinear functions sufficiently and accurately, makes them the prime candidates for use in dynamic models representation of nonlinear plants [4.1].

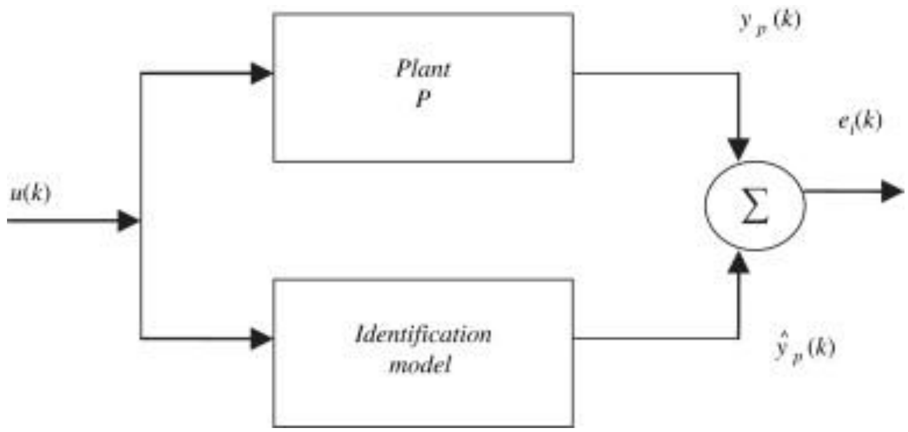


Fig. 4.1 System identification method of multi input, multi output system

Nuclear reactors are highly complex systems that are manned by the operators. As the safety of the plant depends upon the response to various events that affect the plant, identification of events at the earliest is a very important task. Maximum care has to be taken to keep the likelihood of potential risks to a very low value. However, in the event of an unlikely abnormal occurrence of plant condition, the operator has to take necessary actions relatively faster, which involve complex judgments, making trade-offs between partly incompatible demands, and requires expertise to take proper decision. It is well known that timely and correct decisions in these situations could either prevent the undesired incident from developing into a severe accident or palliate the undesired consequences of an accident. The operators sometimes may get confused by seeing multiple alarms, and may lead to delay in reacting [4.2]. Moreover, in such situations, poor decisions may be taken because of the short time available for sorting out the relevant information and lack of expert knowledge. To tackle this, artificial neural network model has been implemented which helps the operators to take diagnostic and corrective actions. Early detection will help in minimizing or even mitigating the negative consequences of such transients. Transient detection can be classified as a pattern recognition problem. When a transient occurs starting from steady state operation, the instrument's readings develop a time dependent pattern. These patterns are unique with respect to the type of accident, severity of accident, and initial conditions. For example, the control and safety rod withdrawal event will differ from primary sodium pump trip event. Therefore, by properly selecting the variables used by the pattern recognition system, the relevant features will be extracted from the measurements.

The objective of this study is to develop an artificial neural network model which can detect the neutronics system related events of PFBR simulator and help operators to take possible control actions in order to keep the reactor in safe operating condition [4.3, 4.4]. PFBR operator training simulator is a full-scope, replica type simulator being developed at Computer Division with the objective of providing comprehensive training to operators in the PFBR plant operations. The scope of simulator covers the entire plant. Being a replica simulator, it completely replicates the reference plant configuration, control system, operator interface and information systems. The data used for event identification is taken from the PFBR simulator. Using artificial neural network, the characteristics of neutronics subsystem have been modeled to identify the occurrence of uncontrolled withdrawal of control and safety rod event.

In a nuclear reactor, neutronic power generation is controlled by the insertion or withdrawal of control and safety rod. In normal operation, the control and safety rods are inserted to introduce negative reactivity (hence reduce the neutronic power) and it is withdrawn to add positive reactivity (hence increase the neutronic power). These operations are manually controlled by operators. The event associated with neutronics system, uncontrolled withdrawal of a control and safety rod, is identified using the artificial neural network model. Most effective process parameters related to this event are properly selected and a range of data is generated for the process parameters using the thermal hydraulics code [4.5]. The data generated is validated and documented in the event analysis report of PFBR [4.6].

PFBR has been provided with two independent and diverse reactor shutdown systems namely shutdown system-1 (SDS 1) and shutdown system-2 (SDS 2), each system consisting of two different types of absorber rod mechanism. They are classified as CSRDM (Control and Safety Rods Drive Mechanism) and DSRDM (Diverse Safety Rods Drive Mechanism). The shutdown systems are capable of ordering safety action through the activation of two sets of mechanisms and absorber rods of different design. The first sets of rods are called control and safety rods and second sets are called diverse safety rods (DSR). There are nine control and safety rods arranged in two rings. The speed of insertion and withdrawal of control and safety rods is fixed as 2 mm/s. During Safety Control Rods Accelerated Movement (SCRAM), the lead holding the control and safety rods and diverse safety rods are de-energized and they drop under gravity. SCRAM is an important safety action which results in the safe shutdown of the reactor. The drop time is less than one second [4.7]. The event associated with the uncontrolled withdrawal of control and safety rods (CSR) is explained below.

4.1.1 Uncontrolled Withdrawal of One Control and Safety Rod

The safety of PFBR is evaluated by the response of the plant to various events which affect the plant and they are known as design basis events (DBE). These design basis events are classified into 4 categories namely category 1 (all planned conditions), category 2 ($> 10^{-2}$/reactor year (ry)), category 3 ($\leq 10^{-2}$/ry but $> 10^{-4}$/ry) and category 4 events (≤ 10^{-4}/ry but > 10^{-6}/ry). A typical event considered for the study is: uncontrolled withdrawal of one control and safety rod, which may happen due to some faults in the drive mechanism or shutdown system logic. Under this event, the positive reactivity is added continuously to the system which in turn will result in SCRAM i.e., dropping of rods to shutdown the reactor. It is a category 2 event (based on frequency of occurrence). The reactor in full power operation has been considered for identification of the event. At the beginning of the reactor start up, all the control rods are at a maximum inserted position. During the startup, diverse safety rods are fully withdrawn and the control and safety rods are then gradually withdrawn to make the reactor critical. Then by further withdrawing the control and safety rods, the reactor is brought to full power. To compensate the burn up loss of the fuel, the control and safety rods are lifted up gradually so as to maintain the full power production in the core. The uncontrolled withdrawal of control and safety rod event has been simulated by considering that one control and safety rod moves upward from its initial location at a speed of 2 mm/s. The reactivity insertion rate during this transient has been calculated based on the speed of movement and the reactivity worth data of control and safety rod corresponding to its position inside the core at that instant.

Because of the insertion of external reactivity, reactor power increases and as a result the temperature of the coolant increases. When the reactivity crosses the trip threshold of +10 percent mili at 3.47 seconds, SCRAM is initiated. Apart from reactivity (ρ) the other effective SCRAM parameters available during this event are: high linear power (Lin P) and high central subassembly outlet temperature (θ_{CSAM}). Among these parameters ρ and θ_{CSAM} are the first SCRAM parameters that trigger reactor SCRAM independently by shutdown system 1 and shutdown system 2 respectively. Because of large uncertainty associated with measurement of ρ, next effective parameter Lin P is considered for shutdown System 1. Fig. 4.2 gives graphs showing the evolution of SCRAM parameters involved in the uncontrolled withdrawal of control and safety rod event.

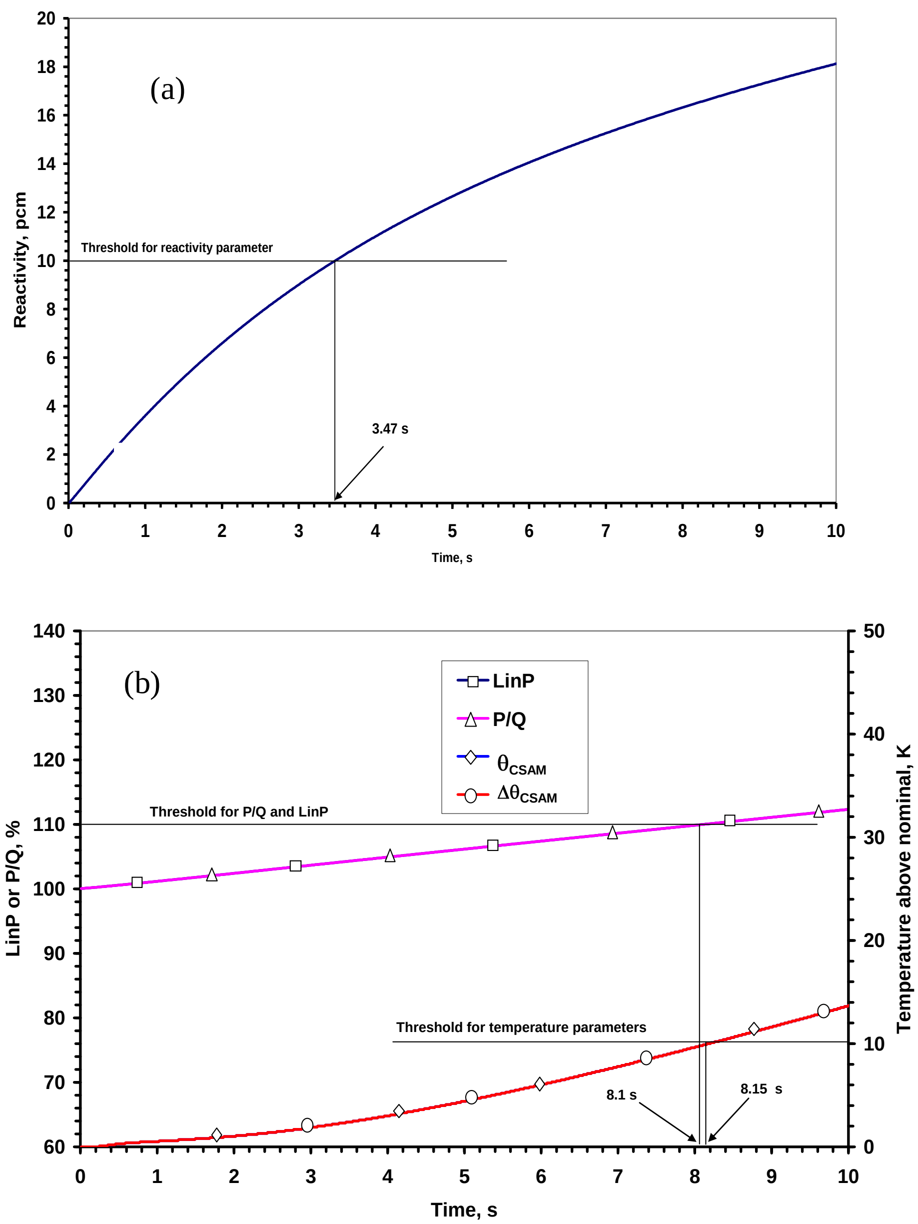


Fig. 4.2 Evolution of process values of (a) reactivity, (b) temperature and power related SCRAM parameter during one control and safety rod withdrawal event

4.1.2 Artificial Neural Network Model for Control and Safety Rod Withdrawal Event

The neural network architecture consisting of three layer model has been chosen to detect the control and safety rod withdrawal event of neutronics System. The input layer consists of three inputs: ρ, Lin P and θ_{CSA} The hidden layer contains the optimal number of hidden nodes, which is decided after several trials and fine tuning. The output layer consists of one output i.e., the control and safety rod withdrawal event. Each neuron is connected to few of its neighbors with varying coefficients of connectivity that represents the strength of these connections. Learning is achieved by adjusting these strengths to cause overall network to output the appropriate results. The neural network architecture of control and safety rod withdrawal event is depicted in Fig.4.3.

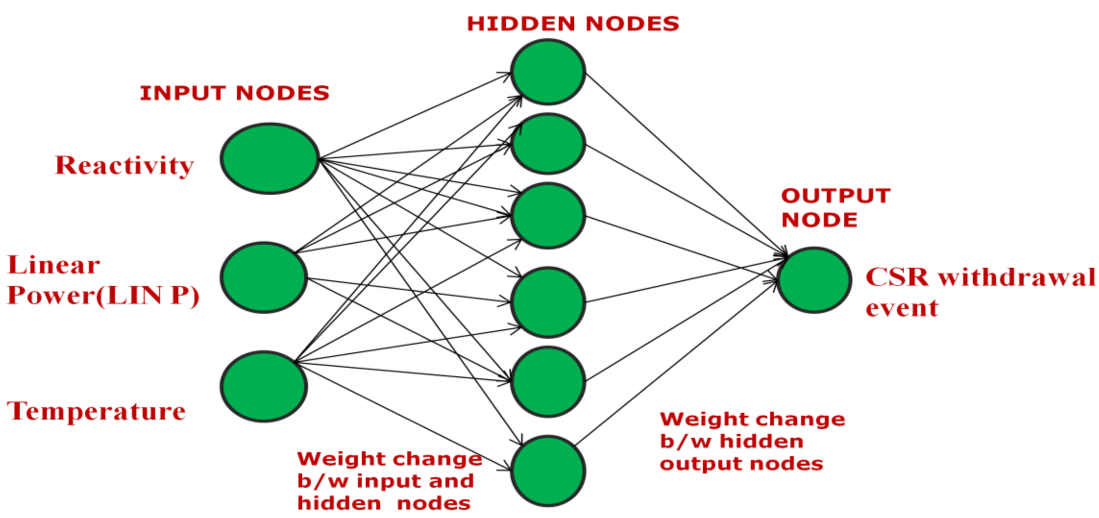


Fig. 4.3 Artificial neural network architecture of control and safety rod withdrawal event

4.1.3 Training and Fine Tuning of the Neural Network

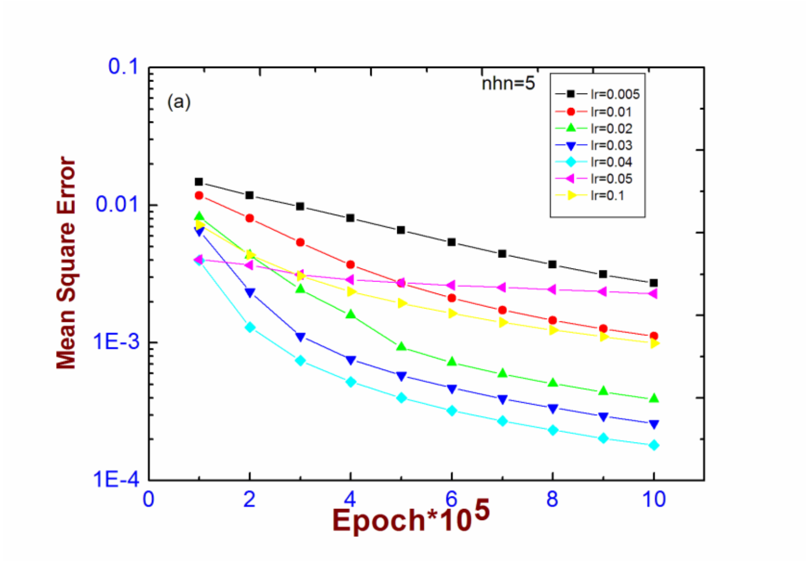
The standard multi layer neural network with back propagation algorithm has been used for the relevant event identification, uncontrolled withdrawal of one control and safety rod. The occurrence of event is subscribed as value 1 and the absence of event is considered as value 0.

The dataset used in training the network is divided into training samples and testing samples in an approximately 90:10 ratio. The dataset is normalized between zero to one and then fed to the neural network with random weights assigned to them. The input signals move from input layer to output layer through hidden layer. The number of hidden nodes in a hidden layer is varied in accordance with the problem. The inputs after getting multiplied with random weights are summed up. The summed up value is fed to a non linear sigmoid activation function. The result is then compared with the desired output and if it disagrees with the desired output the difference in desired and actual output is back propagated and put in the weight change step. The weight is then updated by adding the change in weight to the previous weight. Learning rate parameter is applied to the weight change factor in order to get faster convergence. Although multilayer neural network falls prey to local minima problem, it is a reasonable and well known approximation, that if proper parameters are chosen and the network is trained with a large dataset covering the entire range of operation, the neural network model will approximate the inter-relations that the conventional model obeys. For a continuous regression function, a single layer is sufficient enough for approximation [4.8].

The dataset for training is presented in Table E of Annexure 3 and the output obtained using the test data are compared with the desired output and are presented in

Annexure 3-Table F. Fifty-three samples are used for training and five distinct test samples are used for validation. Both the training and testing samples cover the range of data from steady state to transient. The network is first trained and the weight parameters obtained from the training process are used in the testing phase with unknown set of test samples.

4.1.4 Results and Discussion



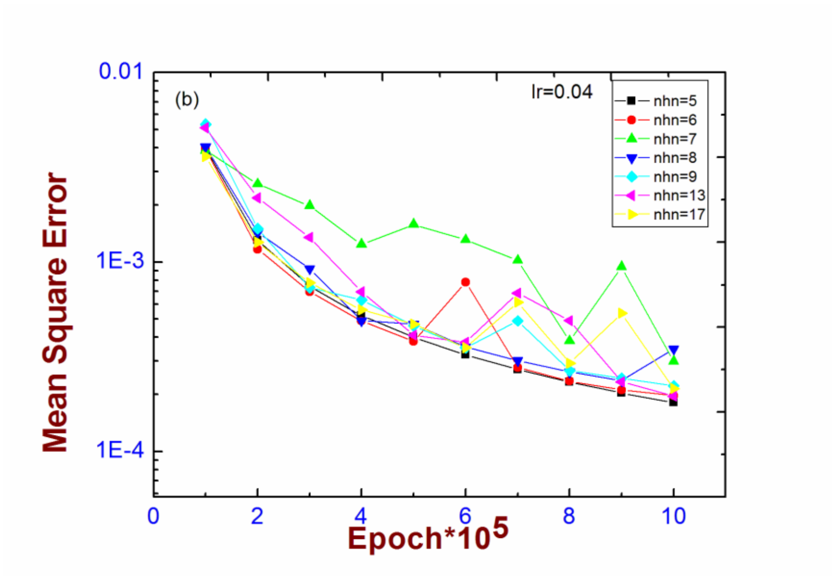


Fig4.4. Graph plotted between mean square errors with respect to number of epochs for (a) various learning rates and (b) hidden nodes for control and safety rod withdrawal event

Different trial methods have been carried out for fine tuning the hidden nodes and learning rates to get the least mean square error. Fig. 4.4 shows at number of hidden node 5 and learning rate 0.04, the mean square error is 1.8×10^{-4} which is the least. The algorithm is then trained for 10^5 iterations and the time taken for this is, 4 minutes 38 seconds in a 2.66 Gigahertz Intel Core 2 Duo PC. It is seen from Fig. 4.5 that at 10^5 iterations the desired output matches with the actual output.

The neural network is then validated using the test samples and it shows that the desired and actual output overlap with each other while there is a slight difference which is well within the acceptable limit. The value of output equal to one indicates the occurrence of the event, whereas the value zero indicates the non occurrence of the event.

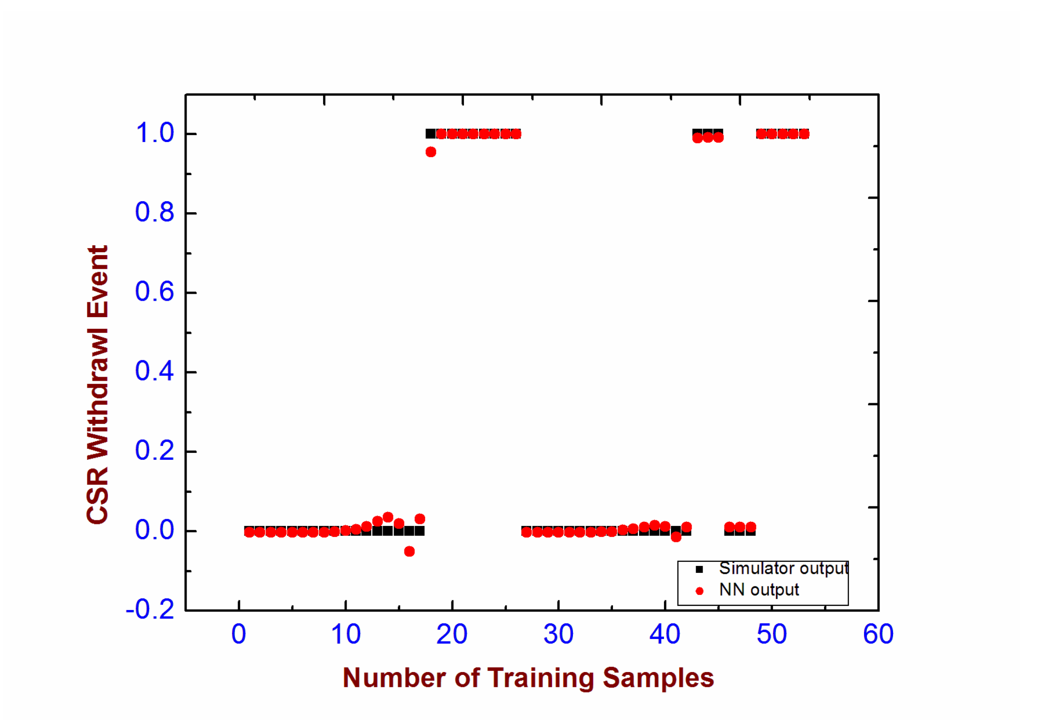


Fig. 4.5. Neural Network results for training sample index of control and safety rod withdrawal event for desired and actual output

Fig. 4.6 shows the results for test samples and the difference between desired and actual output. The difference limit between the desired and actual output is set as ± 0.2 .

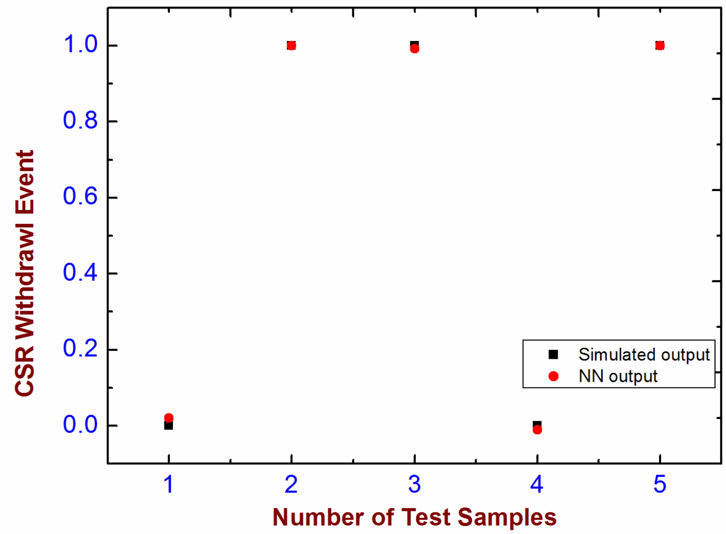


Fig. 4.6 Neural network results for test sample index of control and safety rod withdrawal event for desired and actual output

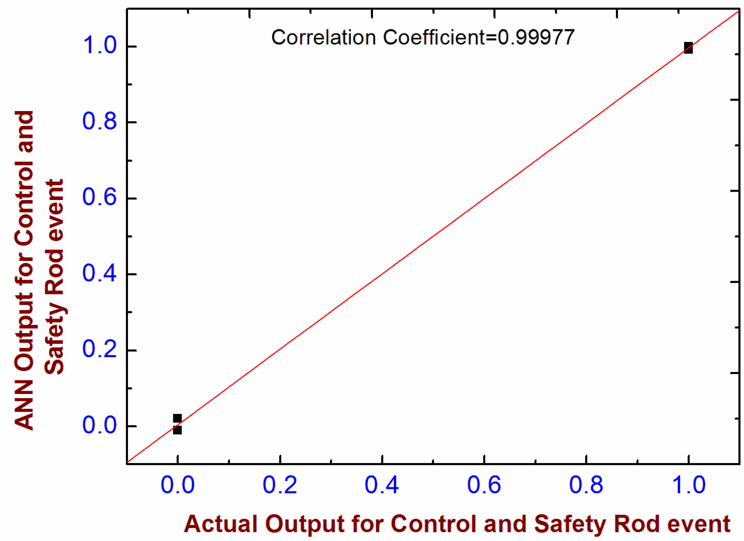


Fig. 4.7 Scatter plot between the results of obtained from the actual and desired output for control and safety rod withdrawal event

Fig. 4.6 shows that uncontrolled withdrawal of control and safety rod event can be detected by neural network model. It shows that output obtained from neural network is agreeing with the outputs obtained from the PFBR simulator. The difference between the desired and actual outputs is well within the permissible range. Fig. 4.7 shows the plot between the desired and actual artificial neural network output. The correlation coefficient is 0.9997, indicating a good fit between the desired and actual output. The event detected using artificial neural network takes less computation time as compared to the event detected by manually operating the PFBR simulator. The time taken to get the first SCRAM using simulator is 3.8 seconds. But artificial neural network model, once trained, takes few milliseconds to get the outputs after validation.

4.2. Event Identification in Primary Sodium Circuit of PFBR using Artificial Neural Network

An artificial neural network model for event identification of primary sodium system of PFBR has been developed. In reactor under normal operating condition, the primary sodium pump takes the sodium from the cold pool towards the core. Due to some mechanical and electrical problems the primary sodium pump may trip and primary sodium pump seizure may occur which can be identified manually after analyzing the process parameters. Current study involves implementation of two artificial neural network models to identify the occurrence of these two events. The effective parameters considered for these two events are SCRAM parameters. The training data for modeling the neural network is prepared using the thermo hydraulics simulation code of PFBR simulator [4.5].

The heat transport system of PFBR consists of primary sodium circuit, secondary sodium circuit and steam water system. The subsystem considered here is primary sodium circuit of PFBR. The primary sodium circuit is contained inside the main vessel of the reactor and consists of two primary sodium pumps and four intermediate heat exchangers. The main objectives of primary sodium circuit are:

- To transfer the nuclear heat generated in the core to the secondary sodium circuit through intermediate heat exchanger and maintain the safe operating temperature of core subassemblies and main vessel.
- To ensure safe operating temperature of the core subassembly and main Vessel by transferring the decay heat from the core subassembly to secondary sodium circuit.

4.2.1 Description of Primary Sodium Pump (PSP)

The primary sodium pump is a single top suction, single stage vertical shaft centrifugal pump placed in cold sodium pool [4.9]. Each pump is located between two intermediate heat exchangers around the core. The pumps operate in parallel across the core. The nominal speed of the pump is 590 rpm (revolutions per minute). The operating temperature of the pump is 670 ^{0}K and the fluid inside the pump is liquid sodium. The schematic of primary sodium pump and its operation is shown in Fig. 4.8 [4.10].

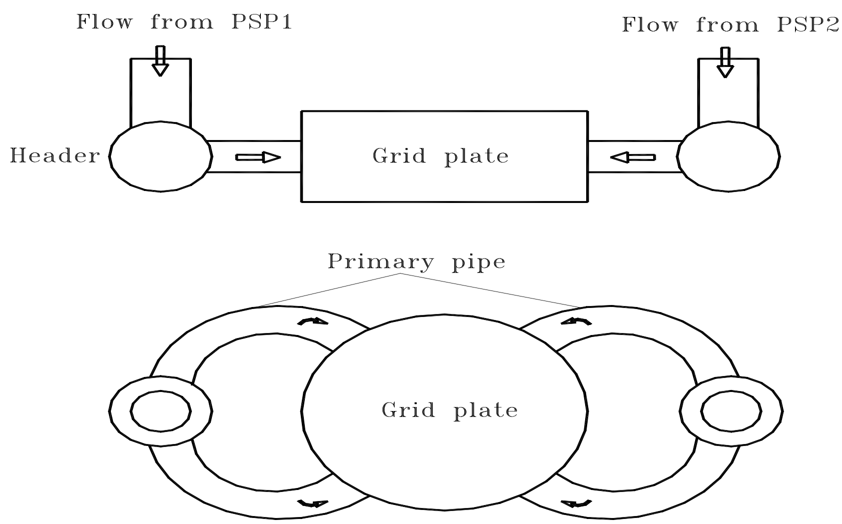


Fig. 4.8 Schematic of primary sodium pump

4.2.2 Events Associated with Primary Sodium Pump

There are several events associated with primary sodium pump. Two important events among those, primary sodium pump trip and primary sodium pump seizure have been considered for modeling. Primary sodium pump trip is a category 2 event. It may happen due to mechanical problems such as misalignment in flywheel, bearings and seismic events etc. and electrical problems like fault in power supply and failure in pony motor etc. Primary sodium pump seizure is a category 3 event. This may occur by some mechanical faults like insufficient hydrostatic bearing clearance, seismic events, foreign particle etc.

4.2.2.1 One Primary Sodium Pump (PSP) Trip event

When one primary sodium pump trip occurs, the speed of the tripped primary sodium pump flow reduces gradually against inertia to 50% in 2.6 seconds and to 0% in 9.4 seconds. Due to parallel operation of two primary sodium pumps, the operating primary sodium pump flow increases to 126% in order to balance the core flow. The total core flow reduces to 61% in 10 seconds. Hence the power to flow ratio (P/Q) increases and the central subassembly outlet temperature (θ_{CSA}) increases which leads to increase in the central subassembly temperature ($\Delta\theta_{CSA}$) and the mean core temperature ($\Delta\theta_{M}$). The temperature of the hot pool sodium increases, resulting in the increase of the cold pool sodium temperature or reactor inlet temperature, leading to SCRAM. Among the SCRAM parameters, five effective SCRAM parameters viz. N_P (Pump speed), P/Q, $\theta_{CSA}, \Delta\theta_{CSA}$ and $\Delta\theta_{M}$ are used for prediction of this event. Among these, the parameters that independently trigger reactor SCRAM by shutdown systems are, N_P and $\Delta\theta_{CSAM}$ respectively. The reactor is designed to ensure that the design safety limits are not exceeded during the normal operational range.

4.2.2.2 One Primary Sodium Pump (PSP) Seizure Event

When a primary sodium pump seizure occurs there is a ramp reduction of the speed of one pump to zero in one second. The second pump is considered to be continuing to operate at full speed. The operating primary sodium pump flow increases to 125% and the core flow reduces to 37% in 1.7seconds. Decrease in flow at such a fast rate, results in the rapid increase of the sodium temperature. Among the SCRAM parameters, six effective parameters viz. N_P, P/Q, θ_{CSAM}, ρ (reactivity), $\Delta\theta_{CSAM}$ and θ_{M} parameters are used for prediction of the event.

4.2.3 Analysis of Design Basis Events

Various SCRAM parameters and the time at which they cross the threshold values are shown in Fig 4.9. It can be observed from the graph that for primary sodium pump trip,

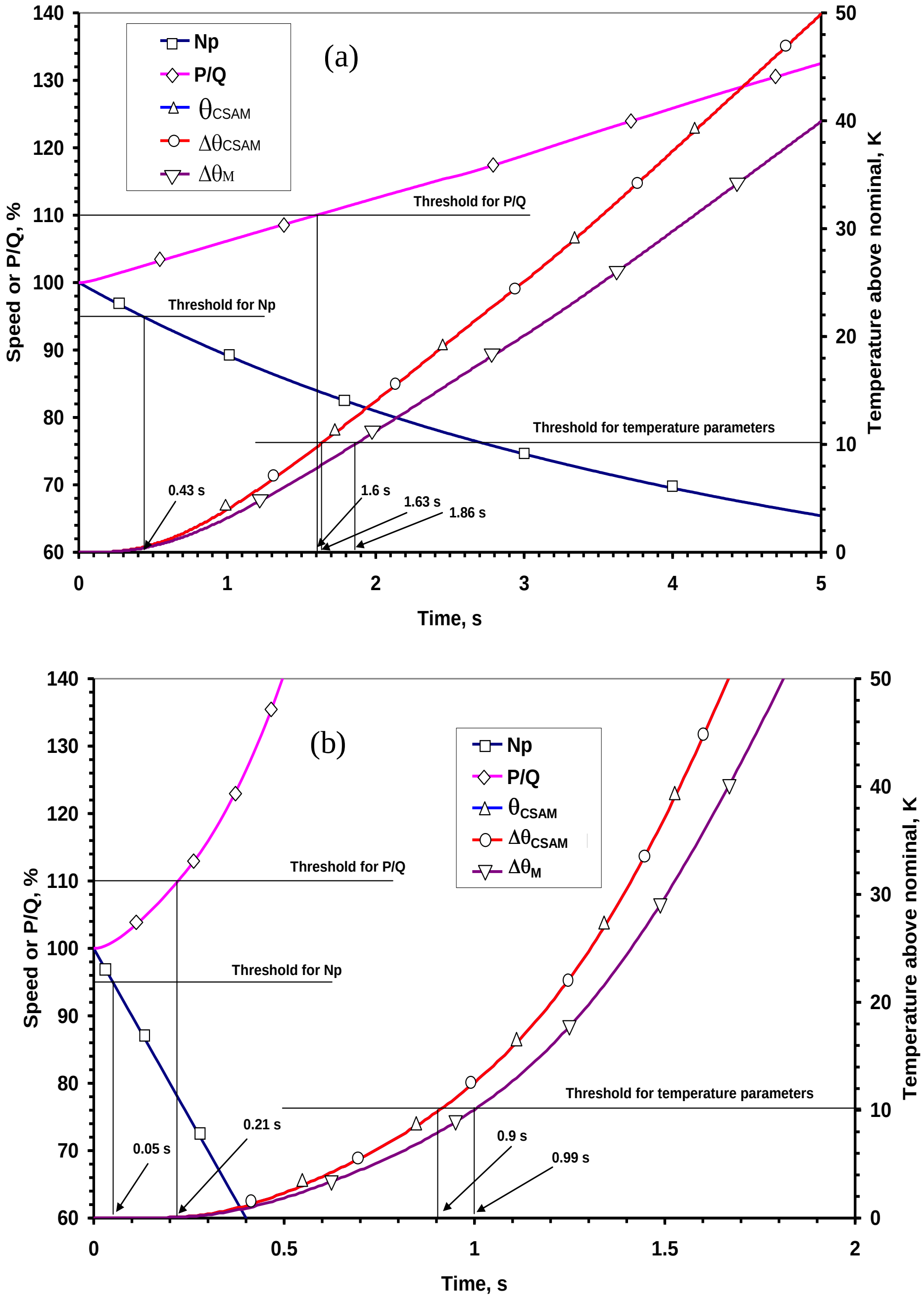


Fig. 4.9 Evolutions of process values of SCRAM parameters during one (a) primary sodium pump trip and (b) primary sodium pump seizure event

the N_p parameter crosses its threshold of 95% at 0.45 seconds and for primary sodium pump seizure, it crosses the threshold at 0.05 seconds. Similarly the P/Q parameter crosses its threshold of 1.1 at 1.6 seconds for primary sodium pump trip and at 0.21 seconds for primary sodium pump seizure. Threshold values for SCRAM parameters are indicated in Fig. 4.9.

4.2.4 Training and Fine Tuning of the Neural Network

A multilayer neural network using back propagation algorithm has been used for training. All the relevant parameters causing the occurrence of the events have been taken as input to the neural network. The event to be identified is presented by the output node of the neural network. The occurrence of event is set as 1 and the absence of event is set as 0. The neural network is having five input nodes and one output node. The input parameters for primary sodium pump trip are N_P, P/Q, θ_{CSA} and $\Delta\theta_{CSA}$ and $\Delta\theta_M$. Likewise, the inputs for primary sodium pump seizure event are six, N_P, P/Q, θ_{CSAM}, ρ (reactivity), $\Delta\theta_{CSAM}$ and θ_M.The transient data is used for training and testing are generated using thermo-hydraulics (DYANA-P)code [4.6].

For the study of primary sodium pump trip, 49 samples are used for training. The dataset for training is presented in Table G of Annexure 4 and the output obtained using the test data are compared with the desired output and are presented in Table H of Annexure 4. The neural network developed for primary sodium pump trip event has three layers. Input layer consists of 5 nodes, the event related SCRAM parameters. The hidden layer can be varied and fine tuned and output layer consists of one output, i.e., the event itself. In case of primary sodium pump seizure, 54 samples are used for training the neural network. The input layer consists of 6 input nodes and output layer has one output node, the event. Both the training and testing samples cover the range of data from steady state to transients. The network is trained using the back propagation algorithm and the coding has been done in C-programming. Fine tuning has been carried out by various trials for finding the optimal number of hidden nodes and learning rates leading to least mean square error. The trial results for primary sodium pump trip and primary sodium pump seizure are shown in Fig. 4.10 and Fig. 4.11 respectively. The optimal mean square error value of 8.1×10^{-5} is achieved with number of hidden nodes 11 and learning rate 0.05 for 10^5 iterations for primary sodium pump trip. The optimal mean square error value of 2.8×10^{-5} is achieved with number of hidden nodes 12 and learning rate= 0.005 for 10^5 iterations for primary sodium pump seizure. The training time for 10^5 iterations is 4 minutes in a 2.66 GHz Intel Core 2 Duo PC.

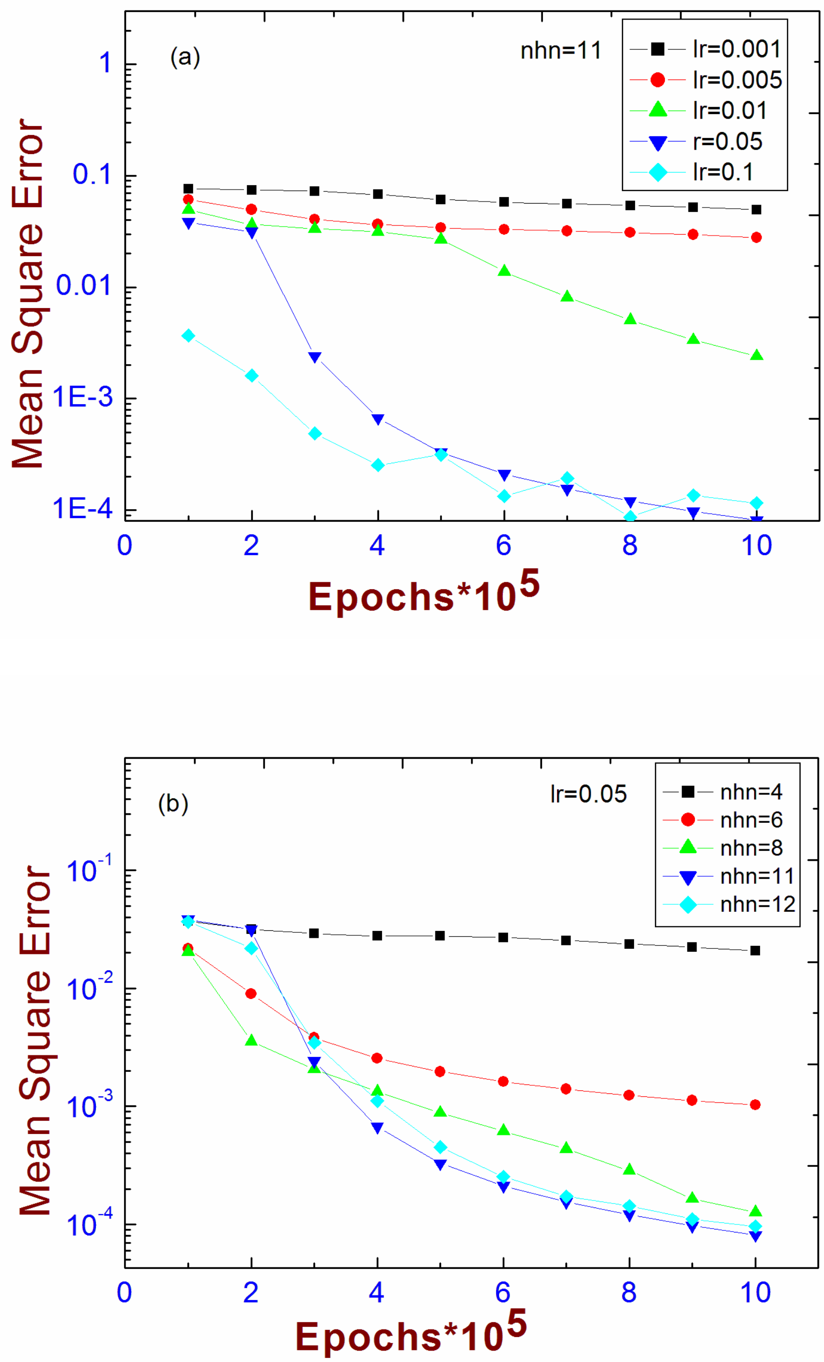


Fig. 4.10 Graph plotted between mean square error and number of epochs for (a) various learning rates (b) number of hidden nodes for primary sodium pump trip event

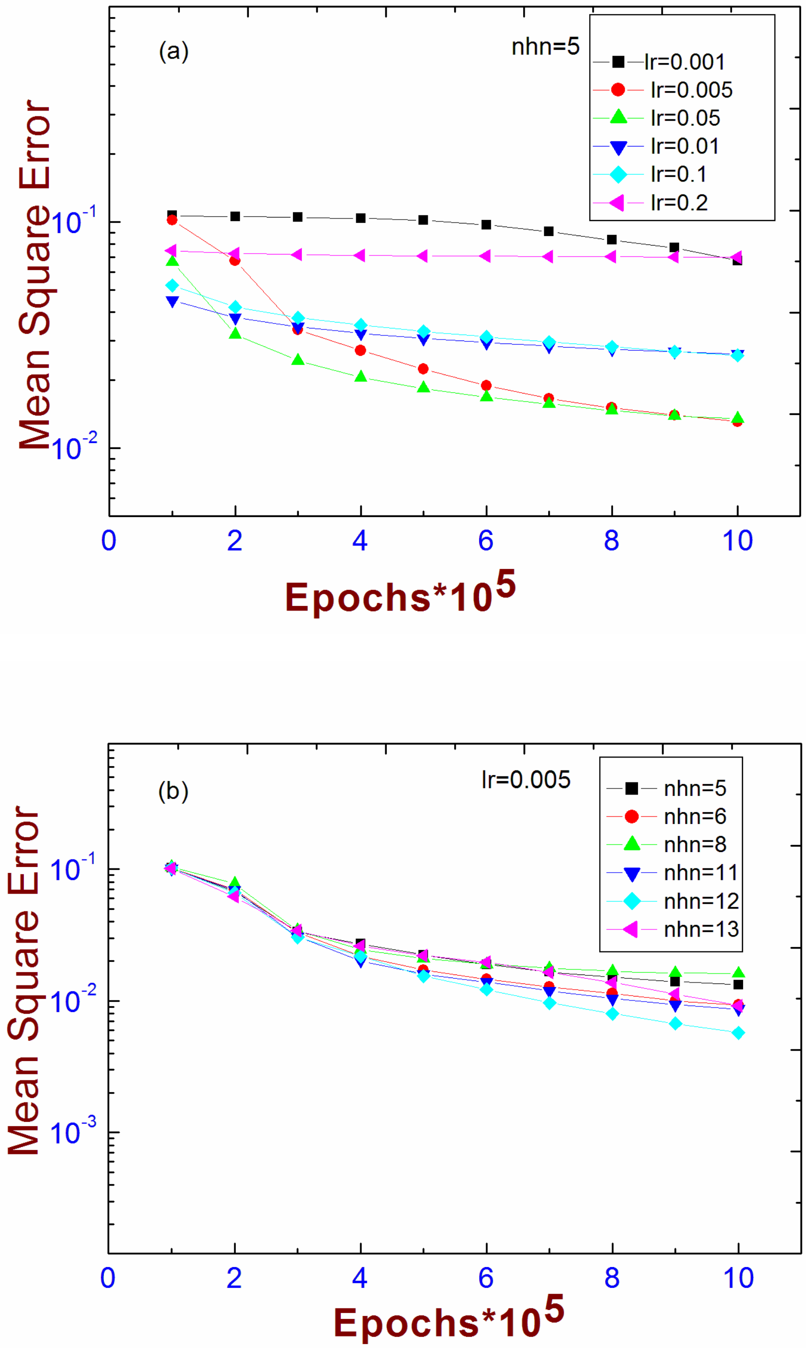


Fig. 4.11 Graph plotted between mean square error and number of epochs for (a) various learning rates (b) number of hidden nodes primary sodium pump seizure event

From Fig. 4.12 and Fig. 4.13 it can be observed that the actual outputs obtained during training and testing for primary sodium pump trip event are almost matching with the desired outputs after 10^5 iterations.

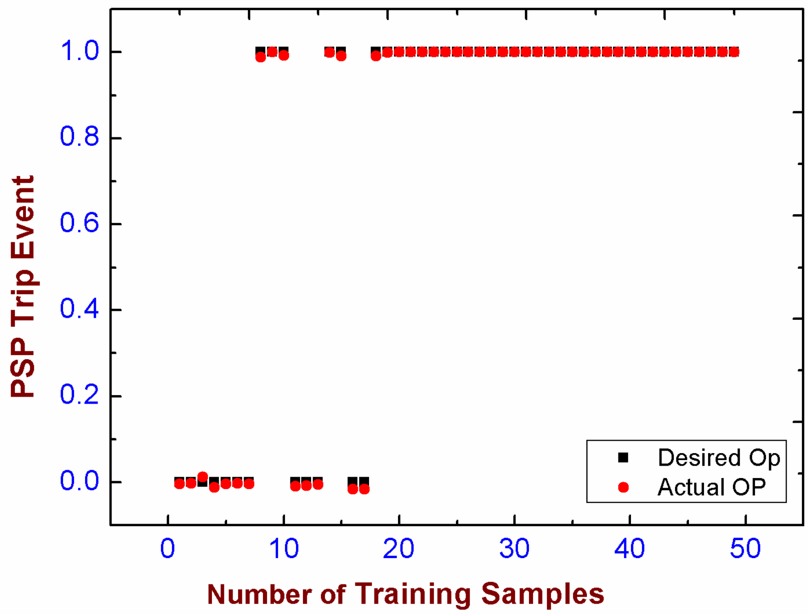


Fig. 4.12 Neural Network training results for primary sodium pump trip event for desired and actual output

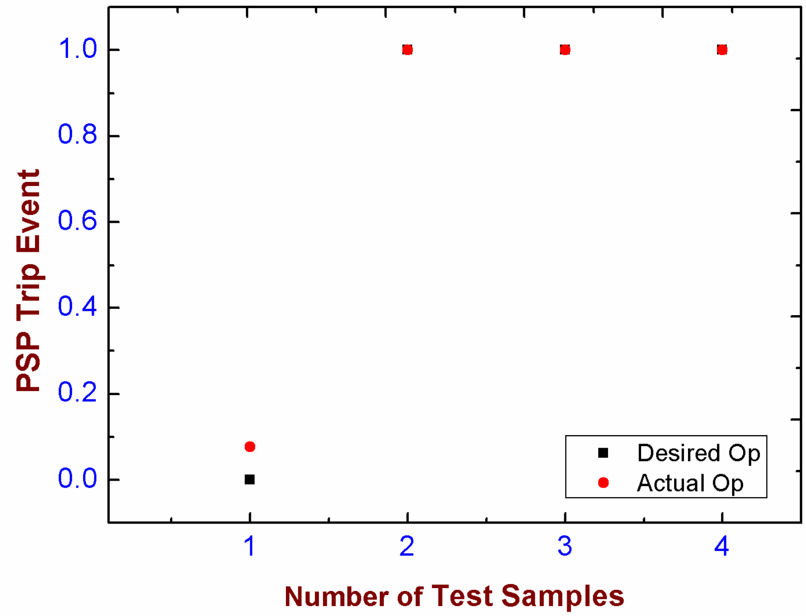


Fig. 4.13 Neural Network test results for primary sodium pump trip event for desired and actual output

For primary sodium pump trip, four distinct test samples and for primary sodium pump seizure, five distinct test samples were taken from the trained input range to validate the network. The neural network model is then validated against the test samples and it can be seen from training and testing results that the occurrence of primary sodium pump trip and primary sodium pump seizure events could be identified with negligible error as shown in Fig. 4.14 and Fig. 4.15 respectively. The difference limit between the desired and the actual output is set as ± 0.2. From the graph it can be seen that the difference is within the permissible range.

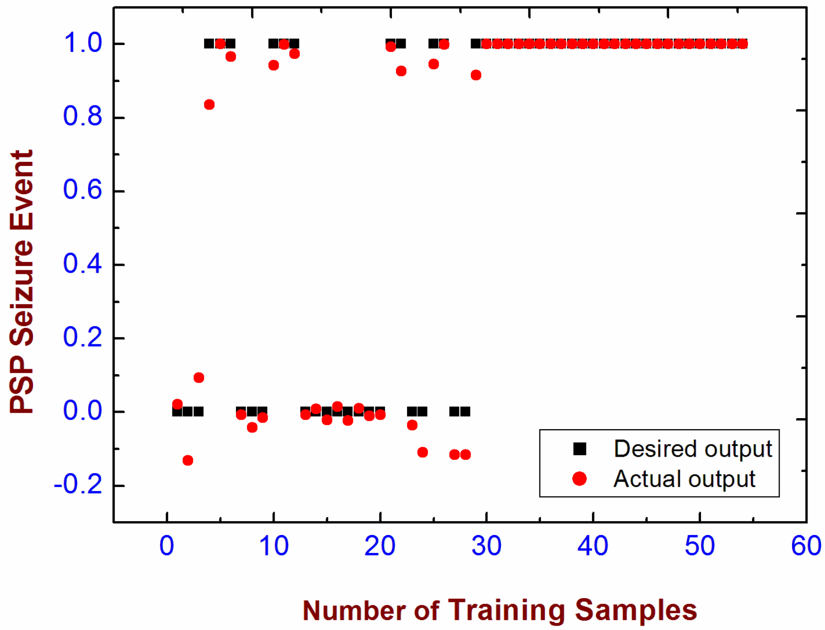


Fig. 4.14 Training results for primary sodium pump seizure event for desired and actual output

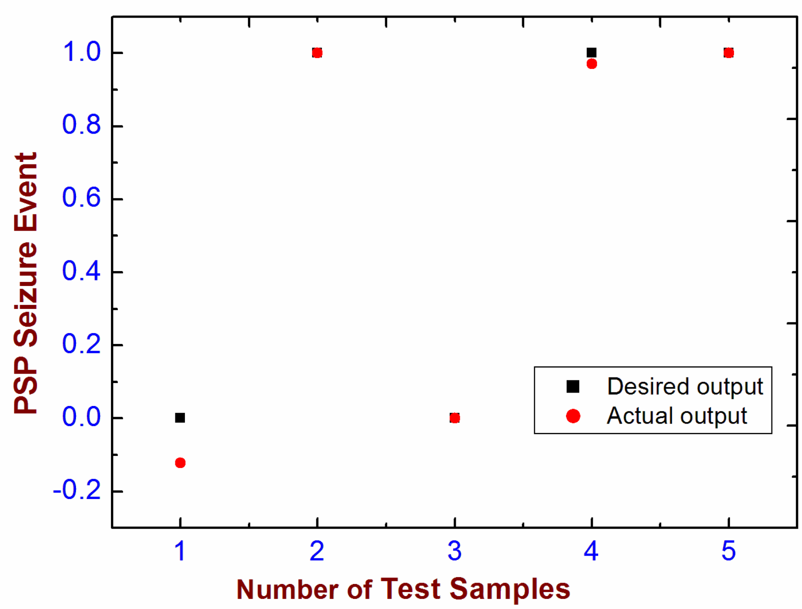


Fig. 4.15 Test results for primary sodium pump seizure event for desired and actual output

In Fig. 4.16 and Fig. 4.17, the scatter plot between the desired and artificial neural network outputs are presented for primary sodium pump trip and primary sodium pump seizure. The correlation coefficient primary sodium pump trip is 1 and primary sodium pump seizure is 0.99689. It shows how properly the variation in the neural network output is explained by the desired output. Correlation coefficient gives the linear relationship between the desired samples and the samples obtained from neural network model. The high degree of correlation coefficient represents the potential of artificial neural network model to identify the event with desired accuracy. The desired accuracy is achieved if difference between expected output and desired output comes in the range of ± 0.01.

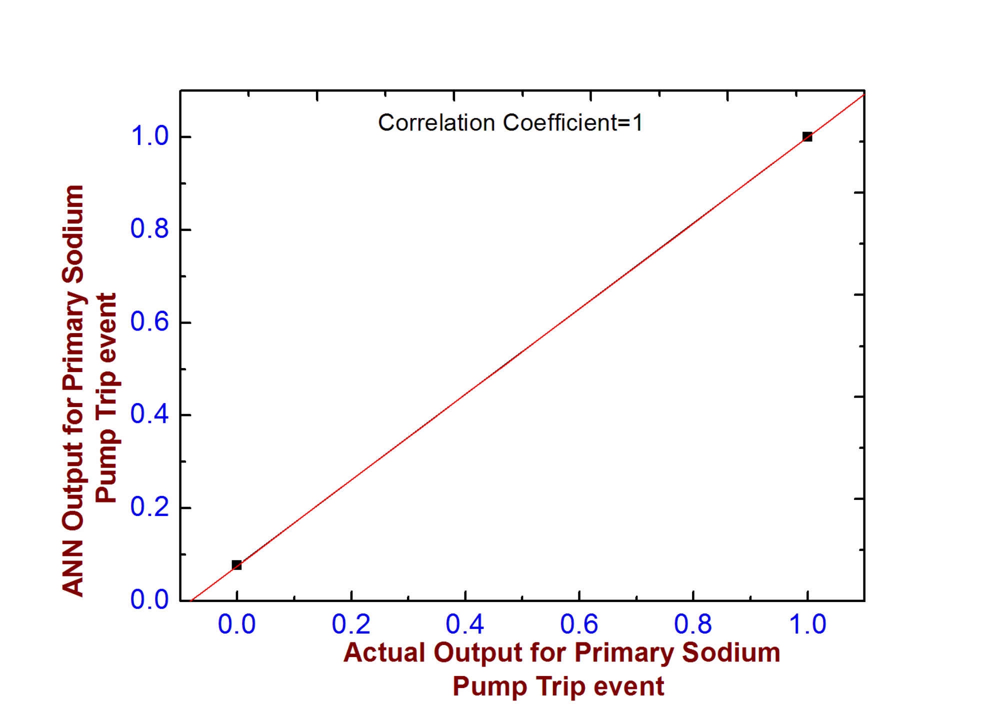


Fig. 4.16 Scatter plot between the results of desired and actual output for primary pump trip event

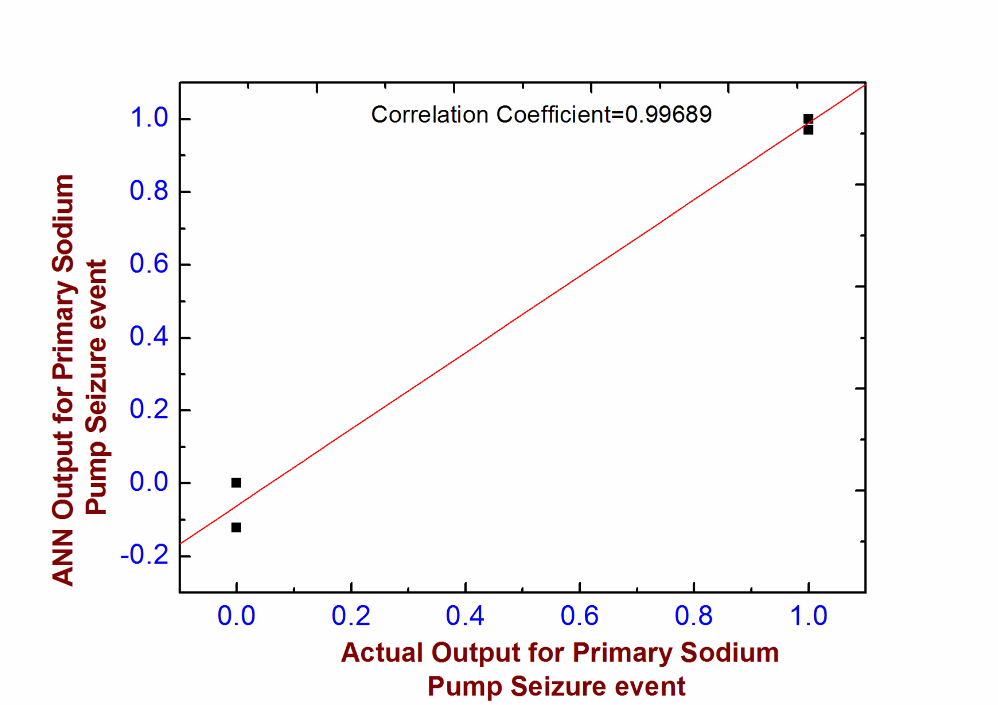


Fig. 4.17 Scatter plot between the results of desired and actual output for primary pump seizure event

The time taken to get the first SCRAM using simulator is 0.43 and 0.05 seconds respectively for primary sodium pump trip and primary sodium pump seizure. Artificial neural network model, once trained, is able to give the desired outputs in few milliseconds.

4.3 Event Detection System for PFBR Subsystems

An event detection system (EDS) has been developed for identifying various events in PFBR subsystems at the earliest time of occurrence. The event detection system is based on, data driven, single neural network model that helps the operator in detecting the events much faster and accurate compared to the conventional methods. It can efficiently simulate the intricate relationship between the individual inputs and outputs while identifying various events. By taking into consideration both explanatory perceptivity and extraordinary predictive capability of neural network, it is concluded that it can efficiently be applied to detect the nuclear reactor related events and understand the phenomena in depth [4.6].

It mainly consists of three stages of processing, namely, input, event identification stage and output stage. In the input stage, the data is taken from the database and processed into training and testing files. This data is used for training and testing the neural networks in the event identification stage. There are unique sets of parameters for different events and they are categorized under design basis events. At the event identification stage, neural network processes the data through a bank of artificial neuron layers and arrives at an optimum solution by adjusting the synaptic weights. Finally, the diagnostic results obtained in the event identification stage are displayed at the output stage. This type of problem is modeled as a pattern recognition problem in artificial neural networks wherein a set of input values (known as pattern) with respect to time represents a class of output. Thus, the events are classified into several classes based on the input patterns [4.11-4.18].

4.3.1 PFBR Related Events

A database of transient data of reactor process parameters ranging from steady state of operation to transient states have been generated using Dynamic Analysis-P code to train the neural network [4.5]. Standard back propagation learning algorithm and its variants have been applied and tested to arrive at the best suited algorithm. The subsystems considered here are primary sodium circuit and neutronics system. The event associated with neutronics system is uncontrolled withdrawal of control and safety rod. The events associated with primary sodium circuit are primary pipe rupture, primary sodium pump trip and primary pump seizure. Table 4.1 shows the events that are taken into consideration for identification using artificial neural network model. The events are modeled based on the relevant reactor process parameters with time dependent data as available on the simulator.

Table4.1. Events associated with PFBR Subsystems

Events	Description
1	Primary Sodium Pump Trip
2	Uncontrolled Withdrawal of Control and Safety Rod
3	Primary Sodium Pump Seizure
4	Primary Pipe Rupture

Control and safety rod withdrawal event, primary sodium pump trip event and primary pump seizure event have been explained in detail in previous section 4.2. In this section primary pipe rupture event is explained.

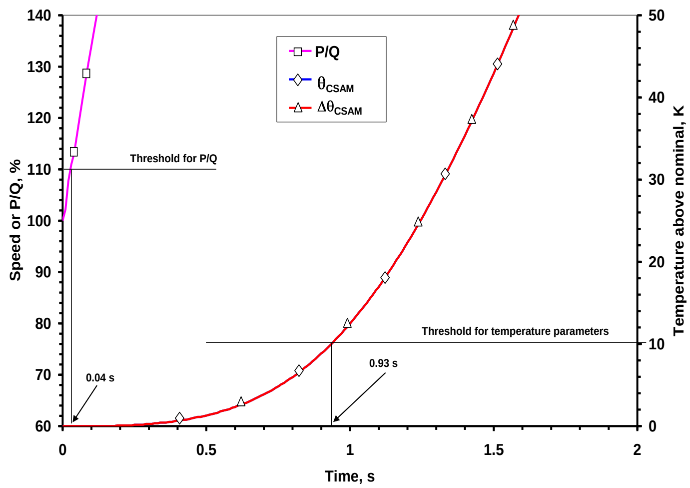


Fig. 4.18 Evolution of process values of temperature and power related SCRAM parameter during primary pipe rupture

In case of primary pipe rupture event, primary sodium flow by-passes the core back to the cold pool through the break and the core flow decreases rapidly. The core flow goes as low as to 30% at about 0.6 seconds before stabilizing at 32 %. The rapid reduction in the flow through the core results in the rise of sodium and core temperatures. Four effective SCRAM parameters viz. P/Q, θ_{CSAM}, ρ and $\Delta\theta_{CSAM}$, are available during the event. The evolution of process values of flow and temperature related SCRAM parameters for both the events are shown in Fig. 4.18.

4.3.2 Data Collection and Description of Algorithms Used in Training

The event related input data has been generated from in-house developed thermal hydraulics code and validated as per the event analysis reports of PFBR. The input dataset containing 172 samples has been chosen in such a way that it covers the entire range of operations from steady state to transient conditions. The dataset for training is presented in Table I of Annexure 5 and the output obtained using the test data are compared with the desired output and are presented in Table J of Annexure 5. The significant parameters namely reactivity (ρ), linear power (Lin P), central subassembly outlet temperature (θ_{CSAM}), increase in central subassembly temperature ($\Delta\theta_{CSA}$), increase of mean core temperature ($\Delta\theta_M$), power to flow ratio (P/Q), pump speed (N_p), are used to represent input nodes to the neural network. The nominal and threshold limits of parameter values associated with the events are shown in Table 4.2. The neural network designed for event detection system is a feed forward network with multilayer perceptron architecture. The network has seven input nodes in the input layer, four output nodes in the output layer and one hidden layer where hidden nodes can be varied.

Table4.2. Nominal and Threshold values for SCRAM parameters

SCRAM parameters	Nominal Value	Threshold
P/Q	1.1	0.99
N_P	590 rpm	-5% of nominal value
θ_{CSAM}	853 K	+10K of nominal value
$\Delta\theta_{CSAM}$	423K	+10K of nominal value
$\Delta\theta_M$	433K	+10K of nominal value
ρ	1.2 pcm	10 percent mili
Lin P	1250 MWt	+10% of nominal value

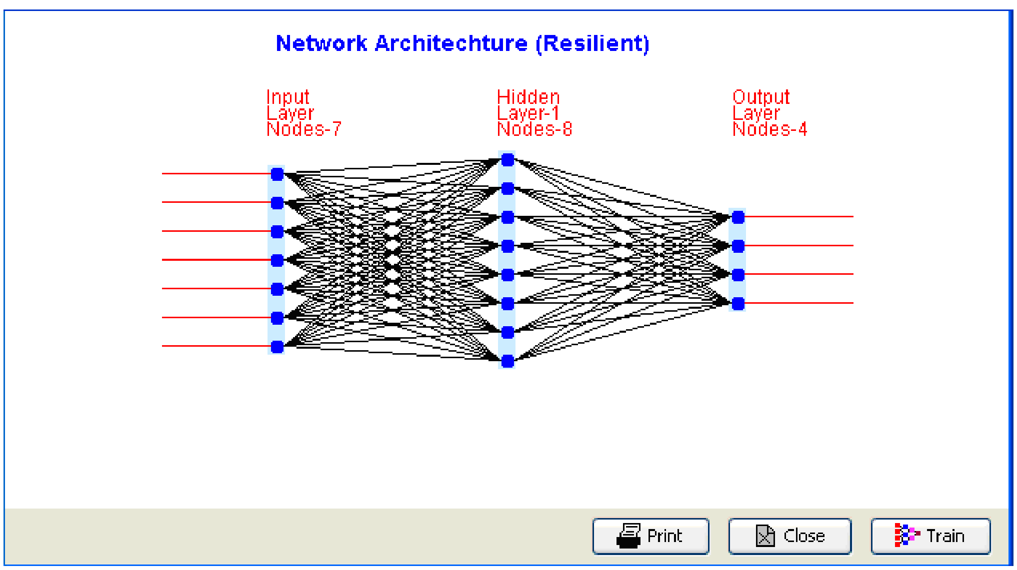


Fig. 4.19 Architecture of neural network of event detection system

The four output nodes in the artificial neural network designate four different events namely; primary sodium pump trip, control and safety rod withdrawal, primary sodium pump seizure and primary pipe rupture respectively. Fig. 4.19 depicts the three-layer neural network architecture used for identifying the four events. The network is trained using BIKAS (Bhabha Atomic Research Centre – Indian Institute of Technology

Kanpur-Artificial Neural Networks – Simulator) for back propagation algorithms and its variants, namely, standard back propagation, back propagation with momentum in pattern mode, back propagation with momentum in batch mode, quick propagation and resilient back propagation algorithm [4.19-4.21].

4.3.3 Fine Tuning of Neural Network

Out of 172 samples in the input dataset, 152 samples have been chosen for training. 20 distinct datasets, which are not included in the training set, are used for prediction. The performance goal error value is set at 1×10^{-4}. The learning rate factor used in weight optimization formula is standardized based on the experience gained from earlier artificial neural network simulation work and set as 0.7. The optimal number of hidden nodes is found to be 8 after carrying out the parametric study with various hidden nodes ranging from 5 to 12. After optimizing the key parameters, the network is trained with different variants of back propagation algorithm to find out the suitable model which produces optimal results.

Fig. 4.20 depicts the graph for standard back propagation algorithm with pattern mode learning. The error value starts with 5.7×10^{-3}. After ten thousand iterations the error reduces up to 7.4×10^{-4}. The time taken to run BIKAS simulator program is 30 minutes for ten thousand iterations in (2.66 GHz Intel Core 2 Duo processor). Fig. 4.21 indicates the back propagation algorithm with pattern mode learning and momentum parameter. The performance goal error value starts with 5.7×10^{-2}. It shows that the mean square error reduces to 9.1×10^{-4} after ten thousand iterations.

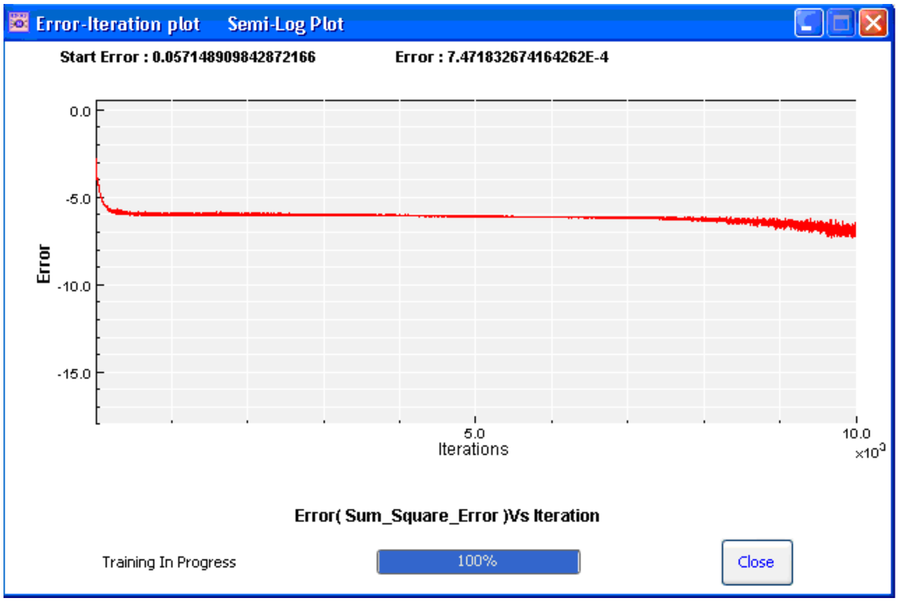


Fig. 4.20 Error vs. epoch curve for standard back propagation algorithm (pattern mode)

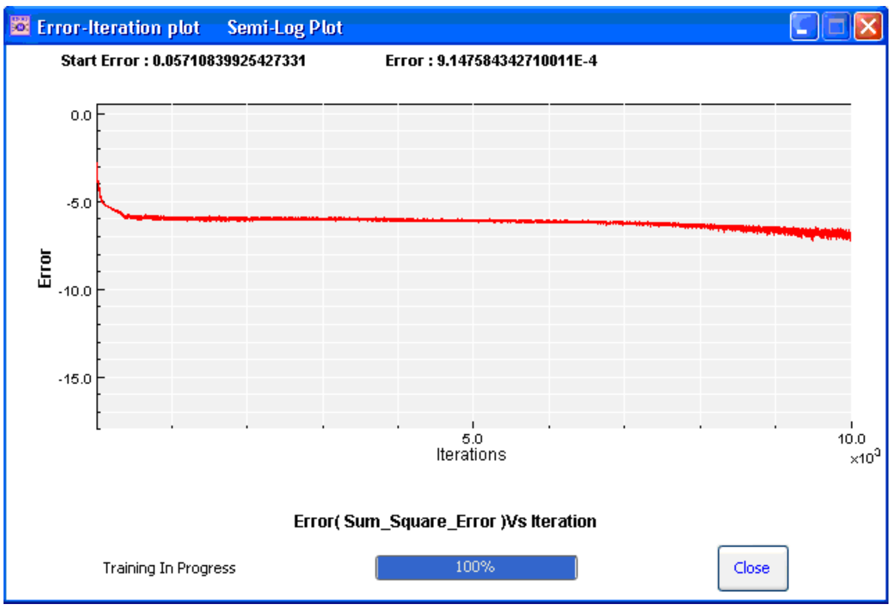


Fig. 4.21 Error vs. epoch curve for standard back propagation algorithm (pattern mode with momentum)

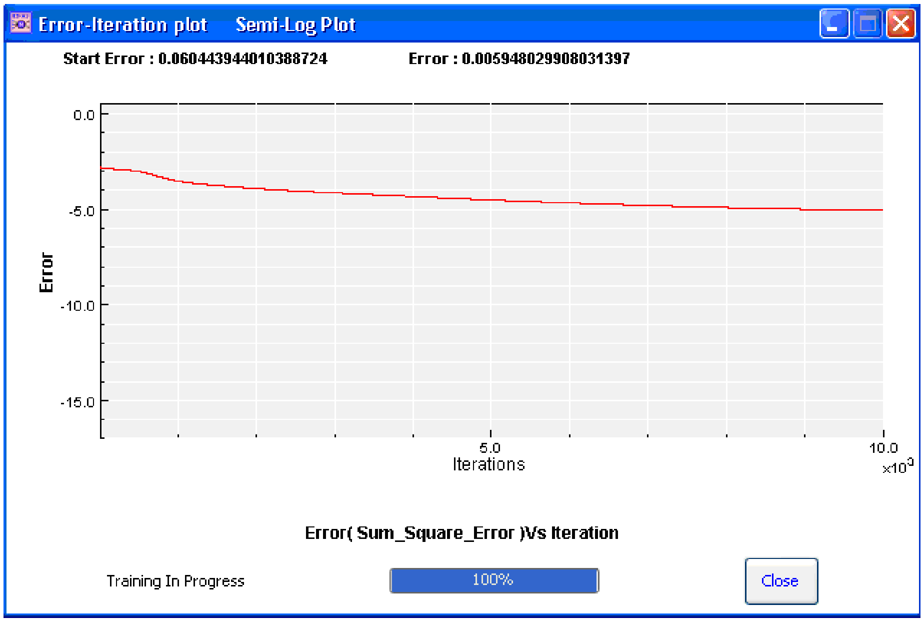


Fig. 4.22 Error vs. epoch curve for standard back propagation algorithm (batch mode with momentum)

Fig. 4.22 shows the standard back propagation algorithm with batch mode learning with momentum parameter. The performance goal error value starts with 6×10^{-2} and after ten thousand iterations the error factor reduces to 5.9×10^{-3}.

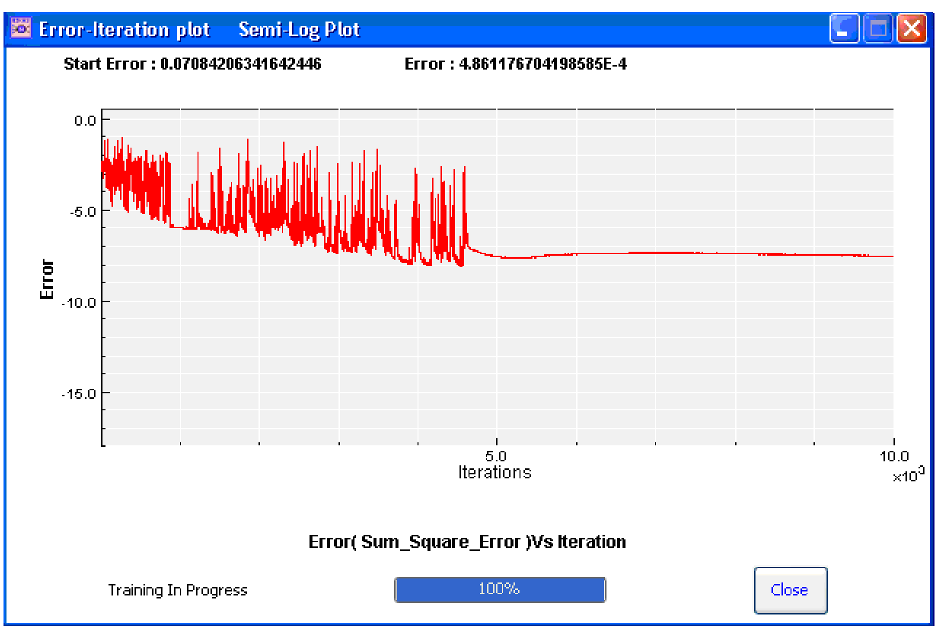


Fig. 4.23 Error vs. epoch curve for quick propagation algorithm

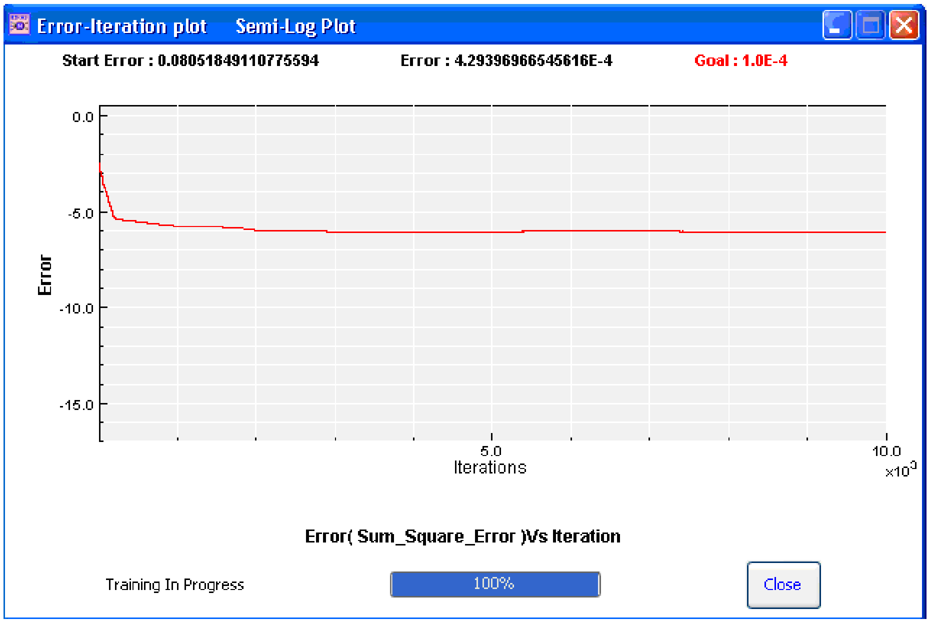


Fig. 4.24 Error vs. epoch curve for resilient back propagation algorithm

Fig. 4.23 shows the graph for quick propagation algorithm. The performance goal error value starts with 7×10^{-2} and after ten thousand iterations the error factor reduces to 4.8×10^{-4}. Fig. 4.24 depicts the graph of mean square error versus epochs. The performance goal error value starts with $8 \times 10^{-2}$and mean square error reduces 4.2×10^{-4} after 10000 iterations.

4.3.4 Results and Discussion

Multi layer feed forward artificial neural network model has been implemented and trained with back propagation algorithm and its variants to identify events related to PFBR subsystems. The best performing algorithm having faster convergence is found out during the training process.

From Fig. 4.25 shown above, it can be seen that the resilient back propagation algorithm is showing faster convergence and yields satisfactory results. The graph also shows that the back propagation algorithm with batch mode and momentum parameter is not able to converge to the required performance goal error even after ten thousand epochs. After training, testing has been carried out for resilient back propagation model

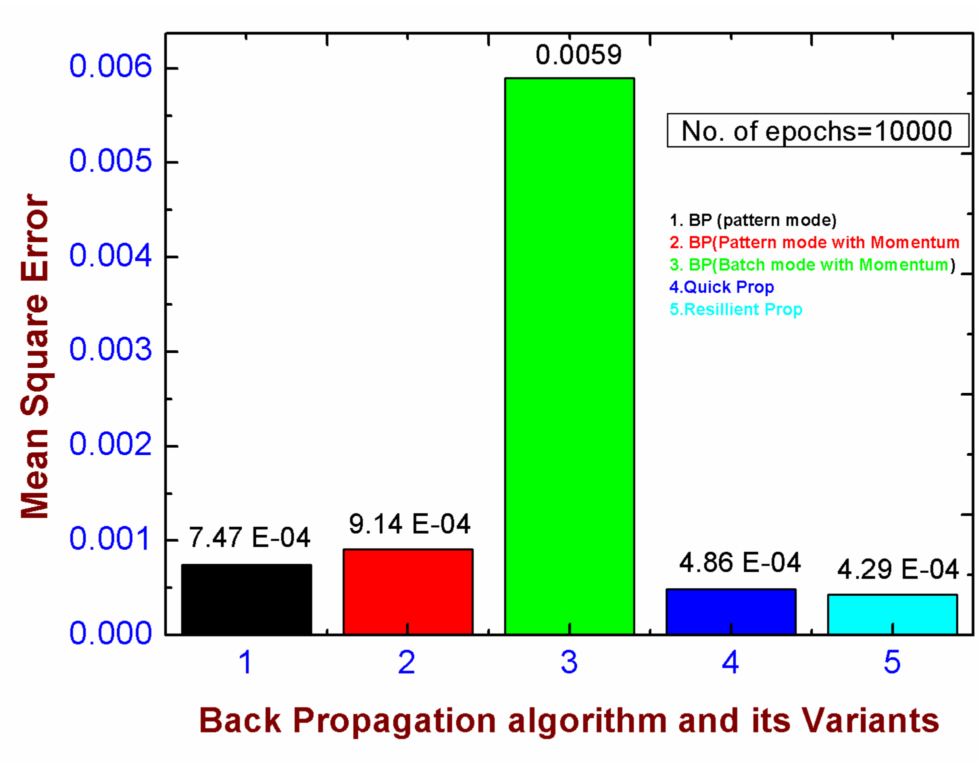


Fig. 4.25 Mean square error for five different algorithms for 10000 epochs

with 20 samples within the range of input data set. The training results are shown in Fig. 4.26 and the graph shows that the neural network results are almost matching with the desired outputs.

Few case studies are carried out with the transients of some selected event scenarios which are not used for training of the artificial neural network. Twenty distinct samples which are not used in training set are applied to the resilient back propagation model for prediction. The red color mark represents primary sodium pump trip event, the green mark represents control and safety rod withdrawal event, the blue color mark represents primary pump seizure event and the light green color mark represents primary pipe rupture event.

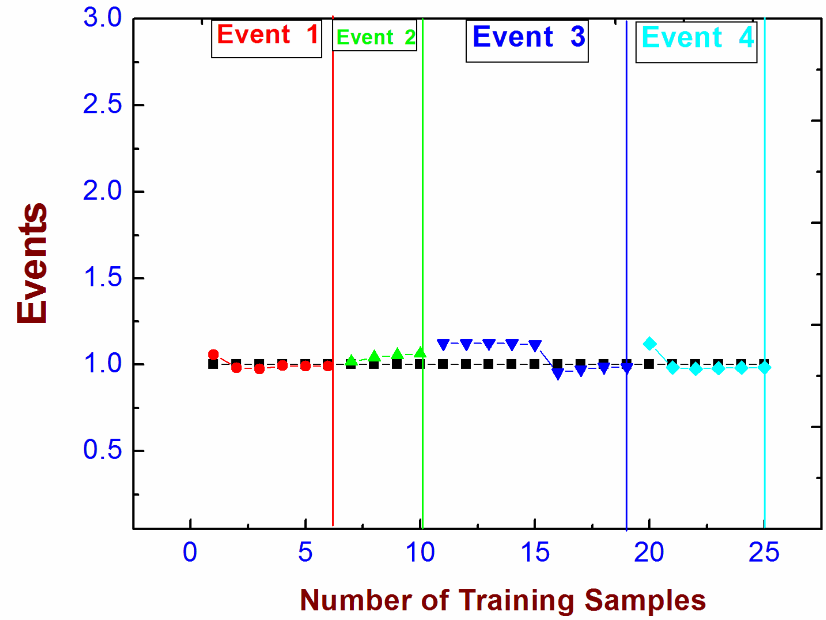


Fig. 4.26 Neural Network results for training sample index of four events for desired and actual output

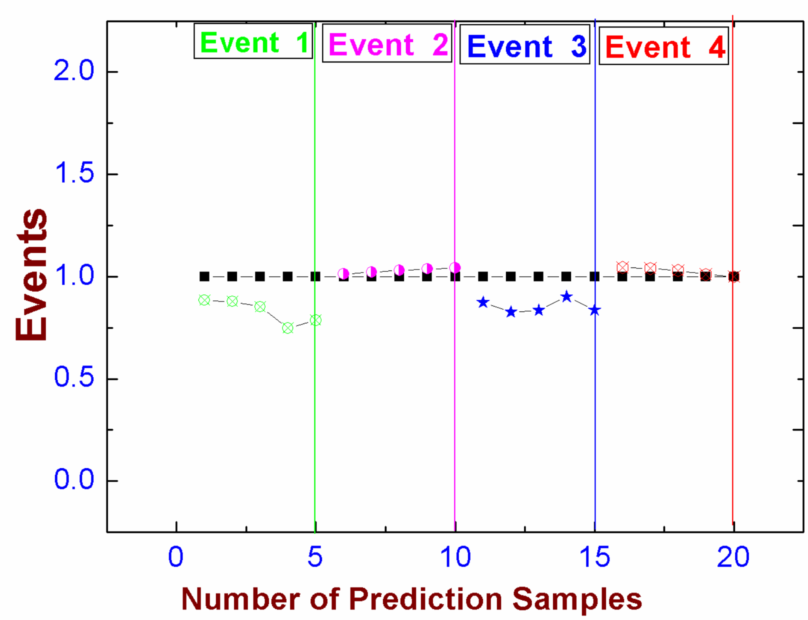


Fig. 4.27 Prediction results for four events for desired and output

Similarly, the green color mark represents primary sodium pump trip event, the pink color mark represents control and safety rod withdrawal event, the blue color mark represents primary pump seizure event and the red color mark represents primary pipe rupture event. The cross validation method has been used by using BIKAS simulator which yields similar results. The predicted results of artificial neural network shown in Fig. 4.27 are in agreement with the results obtained by the simulator data. It could be demonstrated that using neural network model, the occurrence of events can be identified with negligible error and in less computation time in comparison with the conventional model.

4.4 Conclusion

Different neural network architectures for identifications of events in nuclear power plants have been investigated. The systems associated with the PFBR are neutronics system and primary sodium circuit. In first case, uncontrolled withdrawal of control and safety rods is identified using standard back propagation algorithm. The second case is implemented using standard back propagation algorithm to identify primary sodium pump trip and primary sodium pump seizure events. In third case, an event detection system is modeled to identify four events: control and safety rods withdrawal, primary sodium pump trip, primary sodium pump seizure and primary pipe rupture. Back propagation algorithm and its variants are implemented for the identification of these events. After training the network with various algorithms, the analysis results show that the resilient back propagation algorithm is showing faster convergence and yields satisfactory results. It demonstrates that the occurrence of events can be identified with negligible error using the neural network model.

4.5 References

4.1 Khalafi H., Terman M.S., *Progress in Nuclear Energy*, 51 2009 135-140

4.2 Attieh I.K., Gribok A.V., Hines J.W., Uhrig R.E., *Proceedings of the Maintenance and Reliability Conference (MARCON)*, Knoxville, TN, USA 2000

4.3 Santosh T.V., Srivastava A., Sanyasi Rao V.V.S., Ghosh A.K., Kushwaha H.S., *Reliability Engineering and system Safety*, 94 2009 759-762

4.4 Varde P.V, Chowdhury R., Vinod G, Babar A.K, Kushwaha H.S, *International Conference on Quality, Reliability and Control, ICQRC*, 2001 http://www.barc.ernet.in/publications/nl/2002/200210-08.pdf

4.5 *Internal Report on DYANA-P Code-Modeling Details and Validation*, PFBR/66040/DN/112/Rev.A, 2003

4.6 *Internal Report on Analysis of Design Basis Events for Prototype Fast Breeder Reactor, 1st National Conference on Nuclear Reactor Safety*, Mumbai, 2002

4.7 *Internal Report on Neutronics model of PFBR Full Scope Operator Training Simulator*, Report number PFBR/08610/DN/1003/Rev.A, 2006

4.8 Mandal S., Sivaprasad P.V., Venugopal S., Murty K.P.N., Raj B., *Materials Science and Engineering A*, 485 2008 571-580

4.9 Chetal S.C., Bhoje S.B., Kale R.D., Rao ASLK., Mitra T.K., Selvaraj A., Sethi V.K., Sundaramoorthy T.R., Balasubramaniyan V., Vaidyanathan G., Conceptual Design of Heat Transport Systems and Components of PFBR-NSSS, *Technical Committee Meeting*, Kalpakkam, India, IAEA-TECDOC--907 1995 117-132 http://www.iaea.or.at/inisnkm/nkm/aws/fnss/fulltext/28014320.pdf

4.10 Partha Sarathy U.,*Thermal Hydraulic Studies for PFBR using PHOENICS*, Indira Gandhi Centre for Atomic Research, Kalpakkam, 2004 http://www.cham.co.uk/PUC/PUC_Melbourne/presentations/presentation_ups.ppt

4.11 Santosh T.V., Kumar M., Thangamani I., Srivastava A., Dutta A., Verma V., Mukhopadhyay D., Ganju S., Chatterjee B., Rao V.V.S.S., Lele H.G., Ghosh A.K., *Nuclear Engineering and Design*, 241 2011 177-184

4.12 Sorsa T., Koivo H.N., Koivisto H., *IEEE Transactions on Systems, Man and Cybernetics*, 21 1991 815-824

4.13 Parlos A.G., Fernandez B., *Proceedings of IEEE 29th Conference on Decision and*

Control, 3 1990 1703-1706

4.14 Roverso D., *Proceedings of Safe Process, IFAC Symposium on Fault Detection*, *Supervision for Technical Process*, Budapest, HU, 2000 478-483

4.15 Mo K., Leo S. J., Seong P.H., *Progress in Nuclear Technology*, 49 2007 262-272

4.16 Embrechts M.J., Benedek S., *Neural Network Proceedings*, *The 1998 IEEE International Joint Conference on IEEE World Congress on Computational Intelligence*, 2 1998 1438 - 1443

4.17 Santosh T.V., Vinod G., Babar A.K., Kushwaha H.S. Raj V.V., *Reliability Engineering & System Safety*, 83 2003 33-40

4.18 Ma J., Jiang J., *Progress in Nuclear Energy*, 53 2011 255-266

4.19 BIKAS Neural Networks Simulator, *Version 7.0.5*, BARC, 2005

4.20 McKennoch, S., Lui, D., & Bushnell, L., *IEEE World Congress on Computational Intelligence*, Vancouver, 2006

4.21 Reidmiller M., Braun H., *IEEE International Conference on Neural Networks*, 1 1993 586 – 591

Chapter 5
Optimization of Feed Forward Neural Network using Genetic Algorithm

This chapter presents description of genetic algorithm based neural network for parameter estimation of Fast Breeder Reactor subsystem. The parameter estimated here is temperature of the intermediate heat exchanger of FBTR. Genetic algorithm based neural network is a global search algorithm having less probability of being trapped in local minima problem, than a standard back propagation algorithm. Various development stages of genetic algorithm based neural network, such as, the preparation of the training set, weight extraction from the genetic population, training of the neural network and validation phase etc have been described in detail.

5.1. Introduction

Genetic algorithms (GA) are adaptive search and optimization techniques, mimicking the principles of natural evolution. Genetic algorithm has remarkable abilities which include, being able to solve non-smooth, non-continuous, non-differentiable fitness functions to escape the local optima and acquire a neighbourhood optimal solution. Genetic algorithms have been proposed as one of the potential candidates for optimization of weight parameters of neural network. For efficient and quick learning, the weight optimization in back propagation neural network has been carried out using genetic algorithm. Conventionally, standard back propagation network, performing gradient descent learning algorithms, have encountered difficulties by getting trapped in local minima problem. While genetic algorithm does not guarantee a global minimum solution, it can locate the neighborhood of optimum solution. It solves many complex problems by exploring virtually all regions of the state space and exploiting promising areas through mutation, crossover and selection operations applied to individuals in the populations [5.1]. This helps in reducing the large number iterations needed for training the standard back propagation network. Genetic algorithm encodes the parameters of neural network as string of properties of the network, i.e. chromosomes. A large population of chromosomes representing many possible parameters sets is generated and crossover, mutation and reproduction are then performed by replacing the unsuitable candidates in order to arrive at the best fit of optimized parameters.

Genetic algorithms work with population of individual strings, each representing a possible solution to the problem considered. Each string is assigned a fitness value for assessing how good the solution is. The string having high fitness values, participate in reproduction yielding new strings by cross breeding. The least fit individuals are discarded out. A whole new set of population, containing characteristics which are better than their ancestors, are generated by selecting the high fit individuals. Progressing in this way, after many generations, the entire population inheriting the best characteristics is formed. If the genetic algorithm is well implemented, the most promising areas of search space are explored, with the population having fitness values increasing towards the global optimum. A population is said to have converged, if 95% of the individuals constituting the population share the same fitness value [5.2, 5.3]. Fig. 5.1 represents the flow chart representation of the genetic algorithm based feed forward neural network.

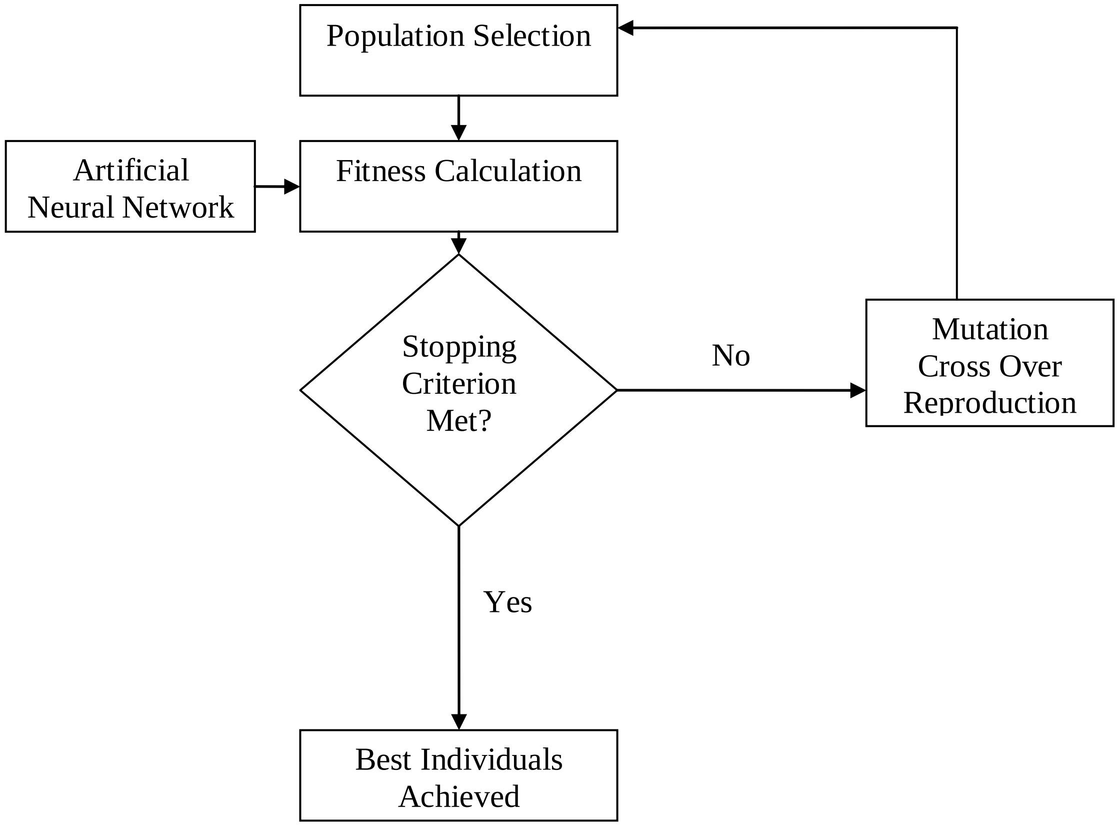


Fig. 5.1 Hybrid genetic algorithm for weight optimization of artificial neural network

5.2. Training of the Network

The training of present network has been accomplished using the genetic algorithm based neural network (GANN) to predict the temperature parameters of intermediate heat exchanger of FBTR [5.4, 5.5]. In order to estimate primary outlet temperature, secondary outlet temperature, the input parameters taken into consideration are primary inlet temperature, primary flow, secondary inlet temperature and secondary flow. The input and output dataset for training the neural network has been generated using the Quadratic Upstream Interpolation for Convective Kinetics (QUICK) scheme which is discussed in detail in section 3.3.2 of Chapter 3. Since the data covers a wide range, it has to be scaled down between zero to one which is also known as normalization. The normalization formula is given by equation 5.1.

$$d_{norm} = \frac{d}{d_{\max}} \qquad (5.1)$$

where d_{norm} the normalized value of the input and d is the input parameter value and $d_{\max}$ is the maximum value of the respective parameters. The normalized data is used for training and testing of the network.

Conventionally, a back propagation network determines its weights based on a gradient search technique and hence runs the risk of encountering the local minima problem [5.6]. On the other hand, genetic algorithm based neural network is found to be good at generating reasonably acceptable solutions with less number of iterations. The key idea is to hybridize genetic algorithm and neural network for weight optimization. Genetic algorithms use a direct analogy of natural behavior and with a population of individual strings, each representing a possible solution to the problem considered [5.7-5.10]. The optimization is based on evolution, and the "Survival of the fittest" concept. The back propagation neural network learning process consists of two stages: firstly employing genetic algorithm to search for optimal or approximate optimal connection weights and thresholds for the network, then using the back-propagation learning rule and training algorithm to adjust the final weights. The weights and thresholds of back propagation neural network are initialized as genes of chromosome, and then the optimal solution is searched through selection, crossover and mutation operators of genetic algorithm. This procedure is completed by applying a back propagation algorithm on the genetic algorithm having initial connection weights and thresholds. If the total mean squared error of back propagation network is bigger than the expected error, the weights and thresholds will be updated; otherwise, these are saved as initial value of propagation network training. After that, the weights are further adjusted under propagation learning rule to get the best results [5.11].

In the present study, the genetic algorithm has been modeled with 12 genes which represent the potential solutions to the problem and the genes are joined together to form a string, referred to as a chromosome. The back propagation network is having the configuration of 4-2-2 representing 4 input layers, 2 hidden layers and 2 output layers. The number of weights that are to be determined are $(4+2)\times 2=12$. With each weight being real number and the number of digits in individual gene or the gene length to be 5, the string length of the chromosome is $12\times 5=60$. This chromosome string represents the weight matrices of the input-hidden and hidden-output layers, in a linear form. An initial population size of p chromosomes is randomly generated. The weights from the individual chromosome are extracted using a weight formula and the fitness function is calculated using FITGEN algorithm [5.3]. These are described in the following section.

5.2.1. Weight extraction

Suppose $X_1, X_2, ...X_d, ..., X_L$ represent a chromosome and $X_{kd+1}, X_{kd+2}, ...X_{(k+1)d}$ represent the k_{th} gene $(k \geq 0)$ in the chromosome. The weight can be calculated by the equation

$$W_k = \begin{cases} +M, & if \quad 5 \leq X_{kd+1} \leq 9 \\ -M, & if \quad 0 \leq X_{kd+1} \leq 5 \end{cases} \tag{5.1}$$

where,
$$M = \frac{X_{kd+2}10^{d-2} + X_{kd+3}10^{d-3} + ...X_{(k+1)d}}{10^{d-2}} \tag{5.2}$$

5.2.2. Algorithm FITGEN

{

Let $\bar{I}_i, \bar{T}_i \quad i = 1,2,\ldots,N;$ where $\bar{I}_i = (I_{1i}, I_{2i}, \ldots I_{li})$ and $\bar{T}_i = (T_{1i}, T_{2i}, \ldots T_{li})$ represent the input–output pairs of the problem to be solved by back propagation network with a configuration $l - m - n$ (l being input neurons, m hidden neurons and n output neurons). For each chromosome $C_i,\ i = 1,2,\ldots,p$ belonging to the current population P_i, whose size is p

{

Extract weights W_i, from C_i, Keeping W_i, as a fixed weight setting, train the back propagation network for the N input instances; calculate error E_i for each of the input instances using the formula,

$$E_i = \sum_j (T_{ji} - O_{ji})^2 \qquad (5.3)$$

Where, O_{ji} is the output vector calculated by back propagation network; find the root mean square E of the errors $E_i,\ i = 1,2,\ldots,N;\ j = 1,2,\ldots,N;$

i.e
$$E = \sqrt{\frac{\sum_i E_i}{N}} \qquad (5.4)$$

Calculate the fitness value F_i for each of the individual string of the population as

$$F_i = 1/E \qquad (5.5)$$

}

Output F_i for each $C_i,\ i = 1,2,\ldots,p$;

}

5.2.3. Genetic Algorithm Based Weight Determination

{

$i \leftarrow 0$;

Generate the initial population P_i of real-coded chromosomes C_i^J each representing a weight set for the back propagation network;

While the current population P_i has not converged

{

Generate fitness values F_i^J for each $C_i^J \in P_i$ using the algorithm FITGEN; Get the mating pool ready by terminating worst fit individuals and duplicating high fit individuals; Using the cross over mechanism, reproduce offspring from the parent chromosomes;

$i \leftarrow i+1$;

Call the current population P_i; Calculate fitness values F_i^J for each $C_i^J \in P_i$;

}

Extract weights from P_i to be used by the back propagation network;

}

5.3. Results and Discussion

The network has been trained using a supervised learning method, with 92 training samples using both standard back propagation algorithm and genetic algorithm based back propagation algorithm. The input and the corresponding learning output have been presented to the network till it learnt the desired relationship. The training data have been normalized to be in the binary form for speedy training of the network. About 90% of the data has been used in the training set and the rest of the data has been used for validation of the network model. In the genetic algorithm based network, a population is said to have converged when 95% of the individuals constituting the population share the same fitness value [5.3]. This criterion is achieved by the present network after 210 iterations with satisfactory results in comparison to 50000 iterations required when trained with standard back propagation algorithm. This indicates the efficiency of the genetic algorithm based back propagation algorithm.The network is validated using nine test samples and the graph is plotted. It can be observed that neural network based genetic algorithm is used to generate reasonably acceptable results with less number of iterations. Thus a lot of time can be saved using this model without sacrificing the appreciable computational accuracy. The difference between the desired and actual output is set as ± 0.1 which is coming well within the permissible limit.

While using the optimized neural network model to predict the sodium temperature, the comparison of results from the predictive values of testing samples which do not participate in training network and actual values of them is shown in Fig. 5.2 and 5.3. As can be seen from the figures, the temperature predicted by the genetic algorithm based neural network model is very close to the actual input data.

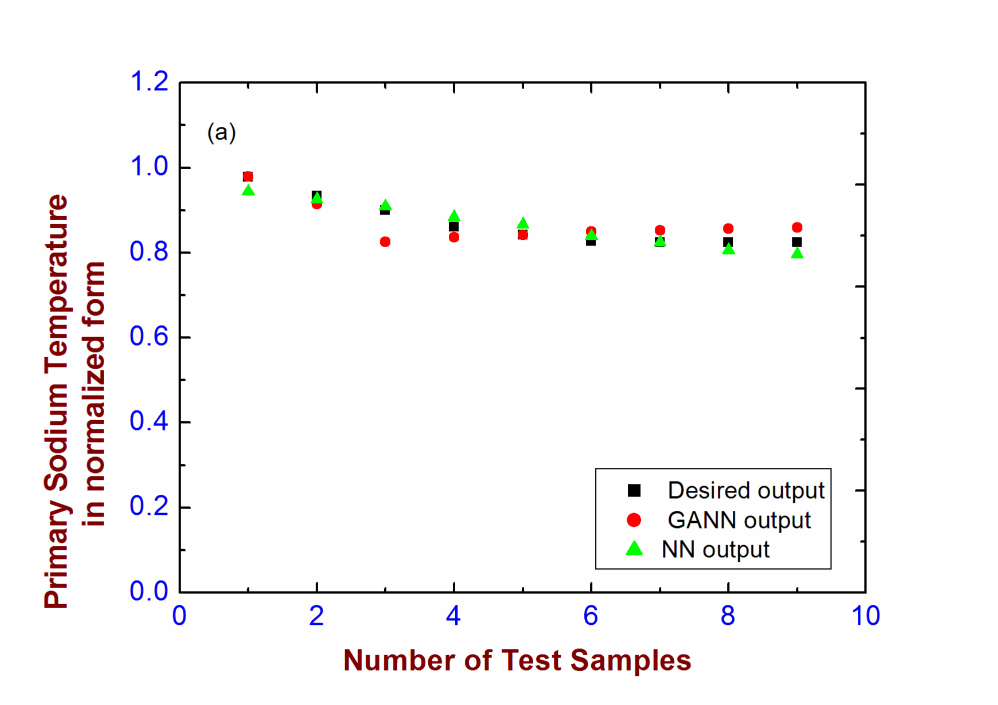


Fig. 5.2 Estimation of primary outlet temperature of Intermediate Heat Exchanger in normalized form for test sample index using artificial neural network (ANN) and genetic algorithm based artificial neural network (GANN)

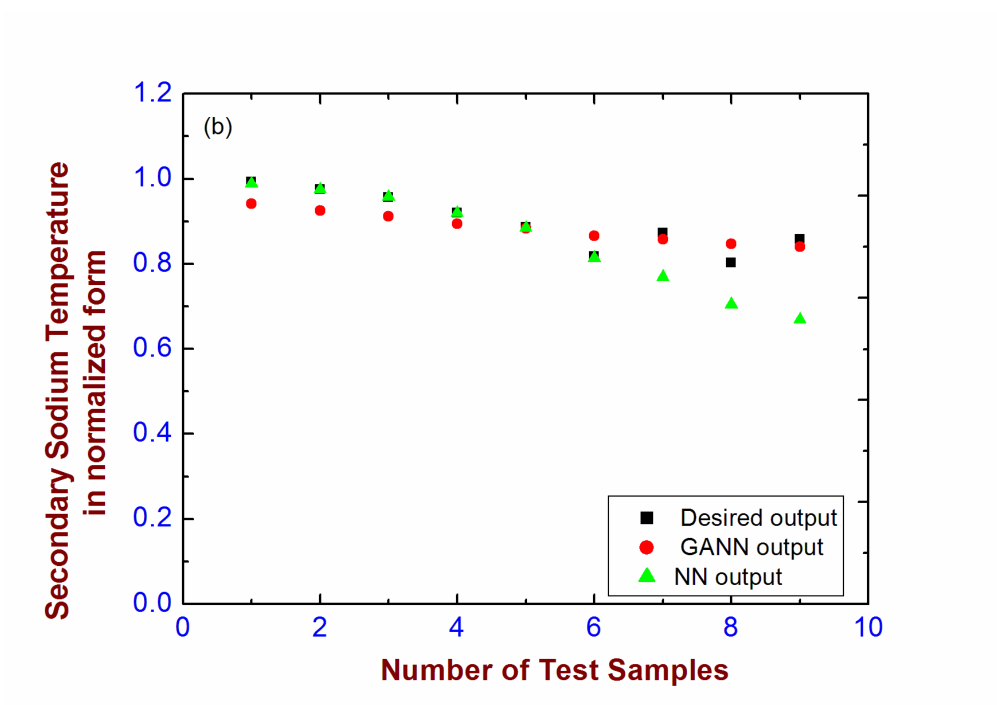


Fig. 5.3 Estimation of secondary outlet temperature of Intermediate Heat Exchanger in normalized form for test sample index using artificial neural network (ANN) and genetic algorithm based artificial neural network (GANN)

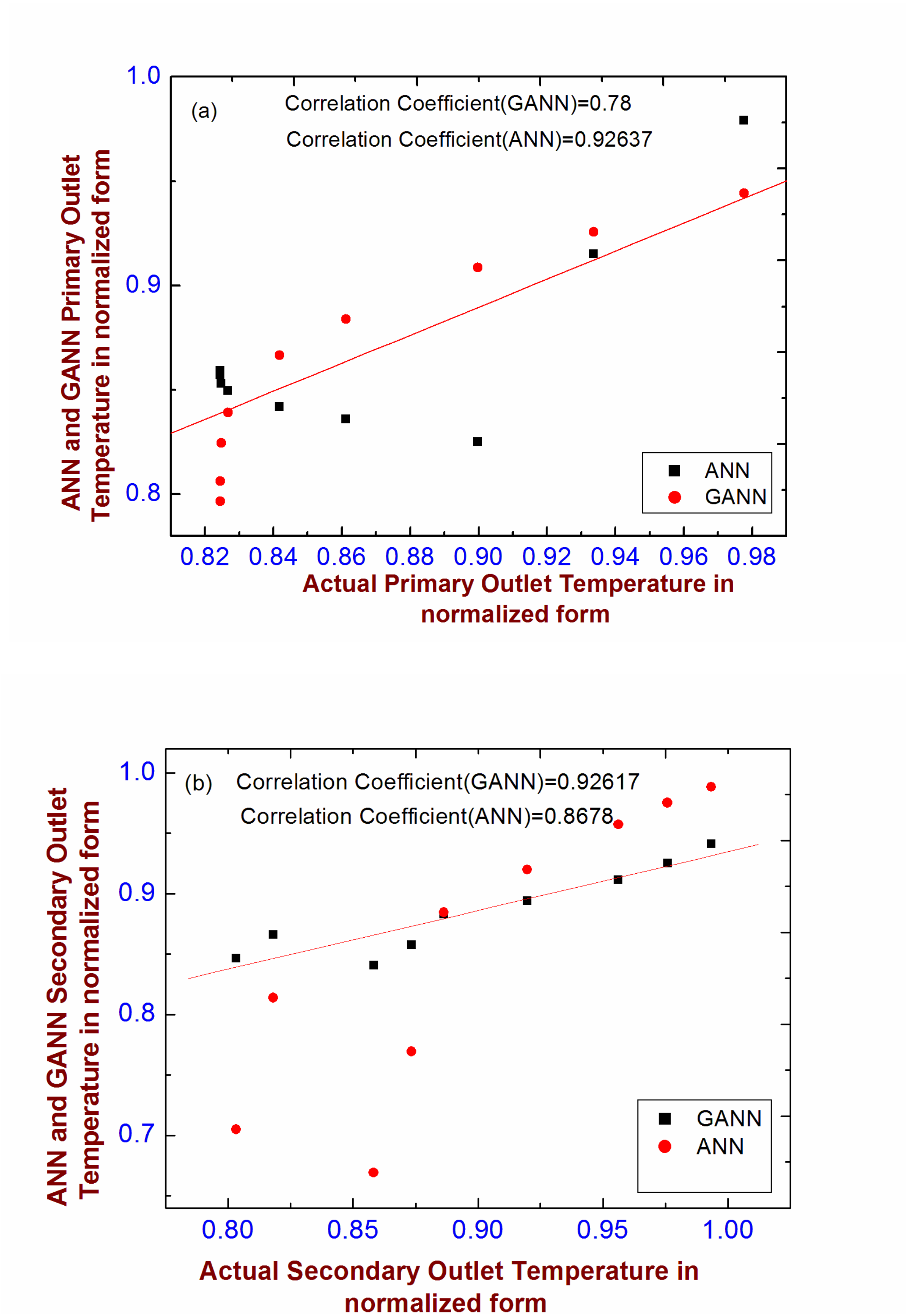


Fig. 5.4 Scatter plot between desired and actual predicted values of artificial neural network (ANN) and genetic algorithm based artificial neural network (GANN) for (a) primary outlet temperature in normalized form (b) secondary outlet temperature in normalized form for Intermediate Heat Exchanger

The scatter plot between the desired and actual output for primary and secondary sodium outlet temperatures have been presented for both artificial neural network based back propagation and genetic algorithm based back propagation network in Fig. 5.4. The correlation coefficient of 0.078 (for genetic algorithm based neural network), (0.92637 for standard back propagation network) for primary outlet temperature and 0.92617 (for genetic algorithm based neural network) and 0.8678 (for standard back propagation network) for secondary outlet temperature is obtained.

5.4. Conclusion

A genetic algorithm based neural network is developed for parameter estimation of nuclear reactor subsystem. The network is implemented to predict primary and secondary sodium temperatures of intermediate heat exchanger of Fast Breeder Test Reactor. The hybrid genetic algorithm is used for weight optimization to enhance the convergence speed. Various stages of development of genetic algorithm based neural network are described. The network is trained both with back propagation algorithm and genetic algorithm based neural network. From the results it could be concluded that genetic algorithm based neural network is a useful method for prediction of parameters in nuclear reactor subsystems with less number of iterations compared to back propagation algorithm providing acceptably good generalization ability and faster convergence. This has been proved to be a useful approach to improve upon the capability of parameter estimation using neural network.

5.5 References

5.1 Hardalac F., *Expert System Applications*, 36 2009 6708–13

5.2 Rajasekaran S., Vijayalakshmi Pai G. A., *PHI learning Pvt. Ltd.*, 2008 305-327

5.3 Rao H., Ghorpade V.G., Mukherjee A., Genetic Algorithm Based Weight Determination for Back Propagation Networks, *Proceedings of Trends in Computing*, *Tata McGraw-Hill*, 1996 73-80

5.4 Erdogan E., Gecknli M., *Annals of Nuclear Energy*, 30 2003 35-53

5.5 Ahmed A.A., Alfishawy N.A., Albrdini M.A., Mahmoud I.I., *Nature and Science*, 5 2011 64-74

5.6 Burse K., Manoria M., Kirar V. P. S., *World Academy of Science, Engineering and Technology*, 72 2010

5.7 Gupta J. N.D., Sexton R. S., *Omega*, 27 1999 679-684

5.8 Montana D. J., Davis L., *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, 1 1989 762-767

5.9 Montana D. J., *John Wiley & Sons*, 1995 85-104

5.10 Ileana I., Rotar C., Incze A., *Proceedings of The International Conference on Theory and Applications Mathematical and Informatics*, Thessaloniki, Greece, 2004 223-24

5.11 Fu Z.,Mo J., Chen L.,Chen W., *Materials and Design*, 31 2010 267–277

Chapter 6

Summary and Suggestions for Future Work

This chapter includes the important conclusions drawn from the results of the research work carried out in the area of parameter estimation and event identification in Fast Breeder Reactor subsystems using artificial neural network techniques. Also, it provides suggestion for future work to be carried out in this field which helps in achieving full-fledged online computational intelligence systems for aid of the operator in condition based maintenance of nuclear reactor subsystems.

6.1. Summary

This work has been motivated by a need to develop computational intelligence systems capable of coping with the complexity and nonlinearity of nuclear reactor subsystems. The objective of this work is to implement artificial neural network models for parameter estimation and event identification in Fast Breeder Reactors. Artificial neural network models have been developed using the input-output measurement data from the conventional models of the reactor to fulfill the above objective. The chapter wise summary of the thesis is given below.

- A brief introduction to nuclear power plant, its complexity and characteristics and the goals to achieve its safe operation are presented in Chapter 1. A detailed explanation of computational intelligence, the hierarchy of whole human engineered intelligence is provided. The application of computational intelligence in the operation of nuclear power plant is also described. The FBTR and PFBR subsystems are briefly covered in the Chapter 1. The scope of the present study namely, parameter estimation and event identification in Fast breeder Reactor systems, are also described in this chapter.

- Basics of artificial neural network, its advantages and disadvantages and its comparison with biological neural network have been described in the Chapter 2. Different architectures of artificial neural network are discussed. Various learning procedures and activation functions used in artificial neural network are outlined. Neural network algorithms are also described in this chapter. A literature survey on applications of artificial neural network in operation and maintenance of various nuclear reactors is presented in this chapter.
- Modeling of two different subsystems of a nuclear reactor, one from FBTR and another from PFBR, for parameter estimation using artificial neural network algorithms have been described in the Chapter 3. A brief description of intermediate heat exchanger of FBTR and neutronics system of PFBR has been presented. Artificial neural network model has been developed to predict process parameters of intermediate heat exchanger subsystem of FBTR. The architecture of the neural network for intermediate heat exchanger and neutronics system is highlighted. The standard back propagation algorithm is described in detail. The temperature parameters of intermediate heat exchanger of FBTR and reactor power of PFBR are estimated using multilayer percepton and back propagation algorithm. Radial basis function is described at length and implemented for prediction of temperature parameters in intermediate heat exchanger starting from steady state to transient state of FBTR. The results are compared with standard

back propagation algorithm. Standard back propagation algorithm and its variants namely, standard back propagation, back propagation with momentum in pattern mode, back propagation with momentum in batch mode, quick propagation and resilient back propagation have been implemented using BIKAS simulator for estimating the reactor power of PFBR.

- Chapter 4 provides detailed description of four events related to two of the PFBR subsystems. The subsystems taken in to consideration are primary sodium circuit and neutronics subsystem of PFBR. The event associated with neutronics system is, uncontrolled withdrawal of control and safety rod event; and events associated with primary sodium circuit are, primary sodium pump trip and primary sodium pump seizure, primary pipe rupture. All the relevant parameters causing the occurrence of the events have been taken as input to the neural network. The event to be identified is given in output node of neural network. The network is trained using the back propagation algorithm and the coding has been done in C-programming. The neural network model is then validated against the test samples and from the results; it is shown that the occurrences of events could be identified faster with negligible error in comparison with the conventional models. Finally, four different events are integrated in a single neural network and back propagation algorithms and its variants are implemented. From the results it is observed that the neural network could be able to identify the events successfully.
- Chapter 5 describes the implementation of genetic algorithm based neural network for prediction of temperature parameters of intermediate heat exchanger

of FBTR. Sometimes, standard back propagation network with gradient descent learning algorithms may encounter difficulties by getting trapped into a local minima problem. Even though genetic algorithm does not guarantee a global minimum solution, it can locate the neighborhood of optimum solution much quicker than conventional strategies and provide encouraging results. The network has been trained using both standard back propagation algorithm and genetic algorithm based back propagation algorithm. It can be observed that genetic algorithm based neural network has been used to generate reasonably acceptable results with less number of iterations compared to standard back propagation algorithms. Thus, considerable amount of time can be saved using this model without sacrificing any appreciable computational accuracy.

6.2. Conclusions

In the present work, neural network approach for parameter estimation and event identification has been proposed. Multi layer neural networks are implemented for parameter estimation and event detection. The reactor data used in training the neural network has been generated from various sources such as theoretical calculations, historical database of reactor and event analysis reports. The results of the applications of neural network to nuclear reactor are summarized below.

- The developed artificial neural network model using standard back propagation algorithm is capable of predicting system parameters accurately and takes much less computational time in comparison with the conventional models of Fast Breeder Reactors.

- The artificial neural network model is also able to detect the events associated with the Fast Breeder Reactor subsystems.
- The model does not require any theoretical rigorous calculation for measuring the system parameters and its inherent attributes allow it to simulate non linear phenomena in complex systems.
- The radial basis function implemented in prediction of temperature parameter, is found to be an effective algorithm in terms of faster convergence criteria, compared to the standard back propagation algorithm.
- Of the variants of back propagation algorithms, implemented in reactor power estimation and event identification in PFBR systems, resilient back propagation algorithm is found to be faster in convergence with less number of iterations.
- The genetic algorithm based neural network implemented for estimation of temperature parameter of intermediate heat exchanger of FBTR yields reasonably acceptable results with less number of iterations compared to standard back propagation algorithm.

6.3. Scope for Future Work

This research work contributes to the theory and applications of neural network learning algorithms. Having successfully applied, the artificial neural network in operation of nuclear reactor applications, the following are suggested for future work.

- There is a need to develop a full-fledged online artificial intelligent system for Fast Breeder Reactors; use of neural network in combination with expert system will be appropriate in this context. After receiving the output from neural network, the outputs can be fed to the expert system which uses the entire knowledge database of nuclear reactor and can display the relevant corrective decisions.

- Each and every subsystem of Fast Breeder Reactor should be modeled using artificial intelligent techniques.
- All other possible events can be detected using various neural network algorithms which will help operator maintenance.
- Furthermore, various other learning methodologies can also be explored to achieve better convergence.
- Various learning methods such as support vector machine of kernel logistic regression or kernel least square regression can be implemented to achieve the better global minimum in learning.

Table A Presents the predicted primary and secondary sodium outlet temperature temperatures with respect to primary and secondary sodium inlet temperatures and primary and secondary flow for training samples of Intermediate Heat Exchanger of Fast Breeder Test Reactor

Table A

Primary Flow m^3/hr	Secondary Flow m^3/hr	Primary Inlet Temperature ^{0}C	Secondary Inlet Temperature ^{0}C	Primary Outlet Temperature ^{0}C	Secondary Outlet Temperature ^{0}C
100	50	400	200	300.4167	399.1665
105	50	400	200	305.0812	399.3295
100	55	400	200	290.9499	398.2730
100	50	410	200	305.4376	409.1248
100	50	400	205	302.9063	399.1873
105	55	400	200	295.9740	398.5950
105	50	410	200	310.3353	409.2959
105	50	400	205	307.4542	399.3463
100	55	410	200	295.4973	408.1866
100	50	410	205	307.9272	409.1456
105	55	410	200	300.7727	408.5248
105	55	400	205	298.5747	398.6302
105	50	410	205	312.7082	409.3127
100	55	410	205	298.2236	408.2298
105	55	410	205	303.3734	408.5599
110	55	410	205	308.0933	408.8134
105	60	410	205	294.3694	407.3536
105	55	420	205	308.1721	418.4897
110	60	410	205	299.3831	407.7975
110	55	420	205	313.1222	418.7556
110	55	410	210	310.5788	408.8423
105	60	420	205	298.7289	417.2245
105	60	410	210	297.1896	407.4182
105	55	420	210	310.7727	418.5248
110	60	420	205	303.9872	417.6902
110	60	410	210	302.0811	407.8513
110	55	420	210	315.6078	418.7845
110	60	420	210	306.6852	417.7439
115	60	420	210	311.4248	418.1025
110	65	420	210	298.1410	416.2229

Table A Continued

Primary Flow m^3/hr	Secondary Flow m^3/hr	Primary Inlet Temperature ^{0}C	Secondary Inlet Temperature ^{0}C	Primary Outlet Temperature ^{0}C	Secondary Outlet Temperature ^{0}C
110	60	430	210	311.2892	427.6364
110	60	420	215	309.3831	417.7975
115	65	420	210	303.1159	416.7949
115	60	430	210	316.2545	428.0122
115	60	420	215	314.0099	418.1477
110	65	430	210	302.3382	426.0431
110	60	430	215	313.9872	427.6902
115	65	430	210	307.5500	426.6423
115	65	420	215	305.8989	416.8712
115	60	430	215	318.8397	428.0573
110	65	430	215	305.2396	426.1331
115	65	430	215	310.3329	426.7186
120	65	430	215	315.0632	427.1910
115	70	430	215	302.2368	424.8966
115	65	440	215	314.7671	436.5660
120	70	430	215	307.1520	425.5960
120	65	440	215	319.7173	437.0604
120	65	430	220	317.7361	427.2563
115	70	440	215	306.2944	434.6592
115	70	430	220	305.2081	425.0153
115	65	440	220	317.5500	436.6423
120	70	440	215	311.4381	435.3918
120	70	430	220	310.0089	425.6989
120	65	440	220	322.3903	437.1257
120	70	440	220	314.2951	435.4942
125	70	440	220	318.9933	436.0835
120	75	440	220	306.6285	433.3944
120	70	450	220	318.5812	445.2894
120	70	440	225	317.1520	435.5966
125	75	440	220	311.4693	434.2178
125	70	450	220	323.4930	445.9055
125	70	440	225	321.7434	436.1725
120	75	450	220	310.5662	443.0941
120	70	450	225	321.4381	445.3918
125	75	450	220	315.6270	443.9549
125	75	440	225	314.3905	434.3492
125	70	450	225	326.2431	445.9944

Table A Continued

Primary Flow m^3/hr	Secondary Flow m^3/hr	Primary Inlet Temperature ^0C	Secondary Inlet Temperature ^0C	Primary Outlet Temperature ^0C	Secondary Outlet Temperature ^0C
120	75	450	225	313.5974	443.2442
125	75	450	225	318.5482	444.0864
130	75	450	225	323.1974	444.7911
125	80	450	225	311.2881	441.7374
125	75	460	225	322.7059	453.8235
130	80	450	225	316.0452	442.6765
130	75	460	225	327.5617	454.5597
130	75	450	230	326.0152	444.9070
125	80	460	225	315.1231	451.3701
125	80	450	230	314.3706	441.9210
125	75	460	230	325.6270	453.9549
130	80	460	225	320.0917	452.3510
130	80	450	230	319.0220	442.8393
130	75	460	230	330.3796	454.6754
130	80	460	230	323.0684	452.5138
135	80	460	230	327.6566	453.3294
130	85	460	230	316.1893	449.9458
130	80	470	230	327.1149	462.1883
130	80	460	235	326.0452	452.6765
135	85	460	230	320.8573	450.9914
135	80	470	230	331.9026	463.0394
135	80	460	235	330.5337	453.4744
130	85	470	230	319.9366	459.5087
130	80	470	235	330.0917	462.3510
135	85	470	230	324.8076	460.5997
135	85	460	235	323.8821	451.1872
135	80	470	235	334.7796	463.1844
130	85	470	235	323.0630	459.7272
135	85	470	235	327.8324	460.7956
140	85	470	235	332.3516	461.7150
135	90	470	235	321.3077	458.0384
135	85	480	235	331.7827	470.4039
135	85	470	240	330.8573	460.9914
140	90	470	235	325.8842	459.1801
140	85	480	235	336.4942	471.3624
140	85	470	240	335.2803	461.8913
135	90	480	235	324.9804	467.5294

Table A Continued

Primary Flow m^3/hr	Secondary Flow m^3/hr	Primary Inlet Temperature ^0C	Secondary Inlet Temperature ^0C	Primary Outlet Temperature ^0C	Secondary Outlet Temperature ^0C
135	90	470	240	324.4714	458.2929
135	85	480	240	334.8076	470.5997
140	90	480	235	329.7516	468.7197
140	90	470	240	328.9505	459.4103
140	85	480	240	339.4229	471.5387
135	90	480	240	328.1441	467.7839
140	90	480	240	332.8179	468.9499
145	90	480	240	337.2631	469.965
140	95	480	240	326.6215	466.0314
140	90	490	240	336.6853	478.4895
140	90	480	245	335.8842	469.1801
145	95	480	240	331.1062	467.2589
145	90	490	240	341.3157	479.5469
145	90	480	245	340.2368	470.1741
140	95	490	240	330.2307	475.4494
140	95	480	245	329.8169	466.3225
140	90	490	245	339.7516	478.7197
145	95	490	240	334.9023	476.7281
145	95	480	245	334.2082	467.5244
145	90	490	245	344.2894	479.7559
140	95	490	245	333.4261	475.7404
145	95	490	245	338.0043	476.9935
150	95	490	245	342.3729	478.0954
145	100	490	245	332.1109	473.9391
145	95	500	245	341.8004	486.4626
145	95	490	250	341.1062	477.2589
150	100	490	245	336.5052	475.2422
150	95	500	245	346.3473	487.6095
150	95	490	250	345.3857	478.3383
145	100	500	245	335.6665	483.2836
145	100	490	250	335.3332	474.2669
145	95	500	250	344.9023	486.7281
150	100	500	245	340.2401	484.6399
150	100	490	250	339.6377	475.5435
150	95	500	250	349.3601	487.8524
145	100	500	250	338.8887	483.6113

Table B Presents the predicted primary and secondary sodium outlet temperature temperatures with respect to primary and secondary sodium inlet temperatures and primary and secondary flow for test samples of Intermediate Heat Exchanger of Fast Breeder Test Reactor

Table B

Primary Flow m^3/hr	Secondary Flow m^3/hr	Primary Inlet Temperature ^0C	Secondary Inlet Temperature ^0C	Primary Outlet Temperature ^0C (Desired Output)	Secondary Outlet Temperature ^0C (Desired Output)	Primary Outlet Temperature ^0C (Actual Output)	Secondary Outlet Temperature ^0C (Actual Output)
100	55	400	205	293.6761	398.316162	293.8872	398.3713
105	55	410	210	305.9740	408.595032	306.1547	408.6037
105	60	420	210	301.5491	417.289062	301.9362	417.2373
110	65	420	215	301.0424	416.312866	301.0706	416.5668
115	65	430	220	313.1159	426.794922	313.2534	426.8904
115	70	440	220	309.2656	434.777893	309.6465	435.2324
120	75	440	225	309.6597	433.544495	309.8797	434.2891
125	75	450	230	321.4693	444.217773	321.7333	444.0524
125	80	460	230	318.2056	451.553711	318.3877	451.7146
130	85	460	235	319.3156	450.164398	319.4108	450.0202

Table C presents the predicted reactor power with respect to control and safety rod (CSR) positions for training samples of Neutronics System

Table C

Sl No.	CSR1 in mm	CSR2 in mm	CSR3 in mm	CSR4 in mm	CSR5 in mm	CSR6 in mm	CSR7 in mm	CSR8 in mm	CSR9 in mm	Reactor Power in Megawatt
1	490	491	492	493	494	495	496	497	498	1095.810
2	491	492	493	494	495	496	497	498	499	1058.816
3	492	493	494	495	496	497	498	499	500	1000.518
4	493	494	495	496	497	498	499	500	501	929.951
5	494	495	496	497	498	499	500	501	502	868.705
6	495	496	497	498	499	500	501	502	503	809.542
7	496	497	498	499	500	501	502	503	504	753.325
8	497	498	499	500	501	502	503	504	505	699.312
9	498	499	500	501	502	503	504	505	506	647.915
10	500	501	502	503	504	505	506	507	508	538.551
11	501	502	503	504	505	506	507	508	509	515.124
12	502	503	504	505	506	507	508	509	510	470.613
13	503	504	505	506	507	508	509	510	511	429.820
14	504	505	506	507	508	509	510	511	512	396.383
15	506	507	508	509	510	511	512	513	514	341.543
16	507	508	509	510	511	512	513	514	515	316.716
17	508	509	510	511	512	513	514	515	516	294.660
18	509	510	511	512	513	514	515	516	517	275.102
19	510	511	512	513	514	515	516	517	518	257.916
20	512	513	514	515	516	517	518	519	520	229.686
21	513	514	515	516	517	518	519	520	521	218.990
22	514	515	516	517	518	519	520	521	522	207.111
23	515	516	517	518	519	520	521	522	523	199.394
24	516	517	518	519	520	521	522	523	524	191.730
25	518	519	520	521	522	523	524	525	526	179.122
26	519	520	521	522	523	524	525	526	527	173.950
27	520	521	522	523	524	525	526	527	528	169.342
28	521	522	523	524	525	526	527	528	529	164.837
29	522	523	524	525	526	527	528	529	530	161.644
30	524	525	526	527	528	529	530	531	532	155.830
31	525	526	527	528	529	530	531	532	533	152.924
32	526	527	528	529	530	531	532	533	534	150.576
33	527	528	529	530	531	532	533	534	535	148.680
34	528	529	530	531	532	533	534	535	536	146.526

Table C Continued

Sl No.	CSR1 in mm	CSR2 in mm	CSR3 in mm	CSR4 in mm	CSR5 in mm	CSR6 in mm	CSR7 in mm	CSR8 in mm	CSR9 in mm	Reactor Power in Megawatt
35	529	530	531	532	533	534	535	536	537	144.772
36	530	531	532	533	534	535	536	537	538	143.170
37	531	532	533	534	535	536	537	538	539	141.708
38	532	533	534	535	536	537	538	539	540	140.230
39	533	534	535	536	537	538	539	540	541	139.118
40	534	535	536	537	538	539	540	541	542	137.856
41	535	536	537	538	539	540	541	542	543	136.809
42	536	537	538	539	540	541	542	543	544	135.944
43	537	538	539	540	541	542	543	544	545	134.934
44	538	539	540	541	542	543	544	545	546	134.184
45	539	540	541	542	543	544	545	546	547	132.394
46	540	541	542	543	544	545	546	547	548	132.577
47	541	542	543	544	545	546	547	548	549	132.040
48	542	543	544	545	546	547	548	549	550	131.250
49	543	544	545	546	547	548	549	550	551	130.641
50	544	545	546	547	548	549	550	551	552	130.134
51	545	546	547	548	549	550	551	552	553	129.537
52	546	547	548	549	550	551	552	553	554	129.084
53	547	548	549	550	551	552	553	554	555	128.546
54	548	549	550	551	552	553	554	555	556	128.048
55	549	550	551	552	553	554	555	556	557	127.671
56	550	551	552	553	554	555	556	557	558	127.296
57	551	552	553	554	555	556	557	558	559	126.909
58	552	553	554	555	556	557	558	559	560	126.550
59	553	554	555	556	557	558	559	560	561	126.196
60	554	555	556	557	558	559	560	561	562	125.828
61	555	556	557	558	559	560	561	562	563	125.508
62	556	557	558	559	560	561	562	563	564	125.265
63	557	558	559	560	561	562	563	564	565	124.944
64	558	559	560	561	562	563	564	565	566	124.640
65	559	560	561	562	563	564	565	566	567	124.370
66	560	561	562	563	564	565	566	567	568	124.114
67	561	562	563	564	565	566	567	568	569	123.898
68	562	563	564	565	566	567	568	569	570	123.664
69	563	564	565	566	567	568	569	570	571	123.460
70	564	565	566	567	568	569	570	571	572	123.245
71	565	566	567	568	569	570	571	572	573	123.016
72	566	567	568	569	570	571	572	573	574	122.786

Table C Continued

Sl No.	CSR1 in mm	CSR2 in mm	CSR3 in mm	CSR4 in mm	CSR5 in mm	CSR6 in mm	CSR7 in mm	CSR8 in mm	CSR9 in mm	Reactor Power in Megawatt
73	567	568	569	570	571	572	573	574	575	122.593
74	568	569	570	571	572	573	574	575	576	122.410
75	569	570	571	572	573	574	575	576	577	122.230
76	570	571	572	573	574	575	576	577	578	122.038
77	571	572	573	574	575	576	577	578	579	121.912
78	572	573	574	575	576	577	578	579	580	121.770
79	573	574	575	576	577	578	579	580	581	121.574
80	574	575	576	577	578	579	580	581	582	121.444
81	575	576	577	578	579	580	581	582	583	121.295
82	576	577	578	579	580	581	582	583	584	121.154
83	577	578	579	580	581	582	583	584	585	121.002
84	578	579	580	581	582	583	584	585	586	120.886
85	579	580	581	582	583	584	585	586	587	120.770
86	580	581	582	583	584	585	586	587	588	120.619
87	581	582	583	584	585	586	587	588	589	120.525
88	582	583	584	585	586	587	588	589	590	120.405
89	583	584	585	586	587	588	589	590	591	120.278
90	584	585	586	587	588	589	590	591	592	120.168
91	585	586	587	588	589	590	591	592	593	120.070
92	586	587	588	589	590	591	592	593	594	119.975
93	587	588	589	590	591	592	593	594	595	119.858
94	588	589	590	591	592	593	594	595	596	119.757
95	589	590	591	592	593	594	595	596	597	119.666

Table D presents the predicted reactor power with respect to control and safety rod positions for test samples of Neutronics System

Table D

CSR 1 in mm	CSR 2 in mm	CSR 3 in mm	CSR4 in mm	CSR5 in mm	CSR6 in mm	CSR7 in mm	CSR 8 in mm	CSR9 in mm	Reactor Power in Megawatt (Desired Output)	Reactor Power in Megawatt (Actual Output)
499	500	501	502	503	504	505	506	507	578.160	578.962
505	506	507	508	509	510	511	512	513	356.209	355.928
511	512	513	514	515	516	517	518	519	236.559	235.332
517	518	519	520	521	522	523	524	525	182.209	181.884
523	524	525	526	527	528	529	530	531	157.050	157.179

Table E presents the predicted control and safety rod withdrawal event output with respect to effective (Safety Control Rod Accelerated Movement) SCRAM parameters for training samples

Table E

ρ	Lin P Megawatt	θ_{CSA} ^{0}K	Output
0	1250	850.0	0
0.7	1253	850.0	0
1.4	1256	850.0	0
2.2	1259	850.0	0
2.9	1262	850.0	0
3.5	1265	850.1	0
4.2	1268	850.1	0
4.7	1271	850.1	0
5.5	1274	850.1	0
6.0	1277	850.2	0
6.5	1280	850.2	0
7.2	1283	850.2	0
7.7	1286	850.3	0
8.2	1289	850.3	0
8.8	1292	850.4	0
9.2	1295	850.5	0
9.6	1298	850.6	0
10.0	1301	850.7	1
10.4	1304	850.8	1
10.9	1307	850.9	1
11.3	1310	851.0	1
12.0	1313	851.2	1
12.4	1316	851.4	1
12.8	1319	851.5	1
13.2	1322	851.6	1
13.6	1325	851.8	1
0	1328	852.0	0
0	1331	852.4	0
0	1334	852.8	0
0	1337	853.2	0
0	1340	853.6	0
0	1343	854.0	0
0	1346	854.4	0
0	1349	854.8	0
0	1352	855.2	0
0	1355	856.0	0
0	1358	856.4	0
0	1361	856.8	0
0	1364	857.2	0
0	1367	857.6	0
0	1370	858.0	0
0	1373	859.2	0

Table E Continued

ρ	Lin P Megawatt	θ_{CSA} ^{0}K	Output
0	1376	859.6	1
0	1379	859.6	1
0	1382	859.6	1
0	0	859.6	0
0	0	859.6	0
0	0	859.6	0
0	0	860.0	1
0	0	860.4	1
0	0	860.8	1
0	0	861.2	1
0	0	861.6	1

ρ= reactivity
Lin P= Linear Power
θ_{CSA}=central subassembly temperature

Table F presents the predicted control and safety rod withdrawal event output with respect to effective (Safety Control Rod Accelerated Movement) SCRAM parameters for test samples

Table F

ρ	Lin P Megawatt	θ_{CSA} ^{0}K	Desired Output	Actual Output
9.6	1298	850.6	0	0.020912
10.4	1304	850.8	1	1.000000
0	1382	859.6	1	0.991220
0	0	859.6	0	-0.010640
0	0	861.2	1	1.000000

Table G presents the predicted primary sodium pump trip event output with respect to effective (Safety Control Rod Accelerated Movement) SCRAM parameters for training samples

Table G

P/Q	$N_P rpm$	$\theta_{CSA}\,^0$K	$\Delta\theta_{CSA}\,^0$K	$\Delta\theta_M\,^0$K	Output
0.00	590	853	423	433	0
0.10	590	853	423	433	0
0.15	586	853	423	433	0
0.20	584	853	423	433	0
0.30	575	853	423	435	0
0.35	570	854	424	435	0
0.40	565	854	424	435	0
0.45	560	855	425	436	1
0.50	535	855	425	436	1
0.60	525	857	427	437	1
0.65	520	858	428	438	0
0.70	515	859	429	438	0
0.80	510	861	431	439	0
1.00	495	863	433	440	1
1.05	490	864	434	441	1
1.10	488	865	435	442	0
1.12	486	865	435	442	0
1.15	485	865	435	443	1
1.20	480	866	436	443	1
1.30	474	869	439	444	1
1.35	470	870	440	445	1
1.40	466	871	441	446	1
1.45	462	872	442	447	1
1.50	458	873	443	448	1
1.55	454	874	444	449	1
1.60	450	875	445	450	1
1.65	446	876	446	451	1
1.70	442	877	447	452	1
1.75	436	878	448	453	1
1.80	433	879	449	454	1
1.85	430	880	450	455	1
1.90	427	881	451	456	1
1.95	425	882	452	457	1
2.00	422	883	453	458	1
2.05	420	884	454	459	1
2.10	417	885	455	460	1
2.15	415	886	456	461	1
2.20	412	887	457	462	1
2.25	410	888	458	463	1
2.30	407	889	459	464	1
2.35	405	890	460	465	1
2.40	402	891	461	466	1
2.45	400	892	462	467	1

Table G Continued

P/Q	$N_P rpm$	$\theta_{CSA}\,^0$K	$\Delta\theta_{CSA}\,^0$K	$\Delta\theta_M\,^0$K	Output
2.50	398	893	463	468	1
2.55	396	894	464	469	1
2.60	393	895	465	470	1
2.65	391	896	466	471	1
2.70	390	897	467	472	1
2.75	388	900	470	473	1

P/Q=power to flow ratio
Lin P= Linear Power
θ_{CSA} =central subassembly temperature
N_P =pump Speed in revolution per minute in ^{0}K
$\Delta\theta_{CSA}$ = change in central subassembly temperature in ^{0}K
$\Delta\theta_M$ = mean core temperature in ^{0}K

Table H presents the predicted primary sodium pump trip event output with respect to effective (Safety Control Rod Accelerated Movement) SCRAM parameters for test samples

Table H

P/Q	$N_P rpm$	$\theta_{CSA}\,{}^0$K	$\Delta\theta_{CSA}\,{}^0$K	$\Delta\theta_M\,{}^0$K	Desired Output	Actual Output
0.25	580	853.5	423.5	434.5	0	0.06789
0.55	530	856.2	426.2	437.0	1	0.99999
1.25	476	868.0	438.0	444.0	1	0.99998
2.30	407	889.0	459.0	464.0	1	1.00000

Annexure 5

Table I presents the predicted events(primary sodium pump trip, uncontrolled withdrawal of control and safety rod, primary pump seizure and primary pipe rupture) output with respect to effective (Safety Control Rod Accelerated Movement) SCRAM parameters for training samples

Table I

Reactivity	P/Q	Linp (Megawatt)	Theta(CSA) k	DeltaTheta(CSA) k	DeltaThetaM k	Np(rpm)	OP1	OP2	OP3	OP4
0.3	0.910	1260	853.0	423.0	433.0	590	1	0	0	0
0.5	0.915	1270	853.0	423.0	433.0	586	1	0	0	0
1.2	0.920	1280	853.0	423.0	433.0	584	1	0	0	0
1.3	0.925	1285	853.5	423.5	434.5	580	1	0	0	0
1.3	0.930	1290	853.7	423.7	435.0	575	1	0	0	0
1.4	0.935	1294	854.5	424.5	435.0	570	1	0	0	0
1.4	0.940	1296	854.2	424.2	435.5	565	1	0	0	0
1.5	0.945	1300	855.0	425.0	436.0	560	1	0	0	0
1.5	1.000	600	862.0	432.0	439.5	505	1	0	0	0
1.5	1.100	300	863.0	433.0	440.0	495	1	0	0	0
1.5	1.050	125	864.0	434.0	441.0	490	1	0	0	0
1.5	1.150	125	865.0	435.0	443.0	485	1	0	0	0
1.5	1.200	125	866.0	436.0	443.5	480	1	0	0	0
1.5	1.250	125	868.0	438.0	444.0	476	1	0	0	0
1.5	1.300	125	869.0	439.0	444.5	474	1	0	0	0
1.5	1.350	125	870.0	440.0	445.0	470	1	0	0	0
1.5	1.400	125	871.0	441.0	446.0	466	1	0	0	0
1.5	1.450	125	872.0	442.0	447.0	462	1	0	0	0
1.5	1.500	125	873.0	443.0	448.0	458	1	0	0	0
1.5	1.550	125	874.0	444.0	449.0	454	1	0	0	0
1.5	1.600	125	875.0	445.0	450.0	450	1	0	0	0
1.5	1.650	125	876.0	446.0	451.0	446	1	0	0	0
1.5	1.700	125	877.0	447.0	452.0	442	1	0	0	0

Table I Continued

Reactivity	P/Q	Linp (Megawatt)	Theta(CSA) k	DeltaTheta(CSA) k	DeltaThetaM k	Np(rpm)	OP1	OP2	OP3	OP4
1.5	1.750	125	878.0	448.0	453.0	436	1	0	0	0
1.5	1.800	125	879.0	449.0	454.0	433	1	0	0	0
1.5	1.850	125	880.0	450.0	455.0	430	1	0	0	0
1.5	1.900	125	881.0	451.0	456.0	427	1	0	0	0
1.5	1.950	125	882.0	452.0	457.0	425	1	0	0	0
1.5	2.050	125	884.0	454.0	459.0	420	1	0	0	0
1.5	2.100	125	885.0	455.0	460.0	417	1	0	0	0
1.5	2.150	125	886.0	456.0	461.0	415	1	0	0	0
1.5	2.200	125	887.0	457.0	462.0	412	1	0	0	0
1.5	2.250	125	888.0	458.0	463.0	410	1	0	0	0
1.5	2.300	125	889.0	459.0	464.0	407	1	0	0	0
1.5	2.350	125	890.0	460.0	465.0	405	1	0	0	0
1.5	2.400	125	891.0	461.0	466.0	402	1	0	0	0
1.5	2.450	125	892.0	462.0	467.0	400	1	0	0	0
1.5	2.500	125	893.0	463.0	468.0	398	1	0	0	0
1.5	2.550	125	894.0	464.0	469.0	396	1	0	0	0
1.5	2.600	125	895.0	465.0	470.0	393	1	0	0	0
1.5	2.650	125	896.0	466.0	471.0	391	1	0	0	0
1.5	2.700	125	897.0	467.0	472.0	390	1	0	0	0
1.5	2.750	125	900.0	470.0	473.0	388	1	0	0	0
1.0	0.904	1253	853.0	423.0	433.0	590	0	1	0	0
1.5	0.908	1256	853.0	423.0	433.0	590	0	1	0	0
2.5	0.918	1262	853.0	423.0	433.0	590	0	1	0	0
3.0	0.924	1265	853.1	423.1	433.0	590	0	1	0	0
3.5	0.928	1268	853.1	423.1	433.0	590	0	1	0	0
4.0	0.932	1271	853.1	423.1	434.0	590	0	1	0	0
4.5	0.936	1274	853.1	423.1	434.0	590	0	1	0	0
5.0	0.938	1277	853.2	423.2	434.0	590	0	1	0	0
5.5	0.944	1280	853.2	423.2	435.0	590	0	1	0	0
6.0	0.948	1283	853.2	423.2	436.0	590	0	1	0	0
6.5	0.953	1286	853.3	423.3	437.0	590	0	1	0	0

Table I Continued

Reactivity	P/Q	Linp (Megawatt)	Theta(CSA) k	DeltaTheta(CSA) k	DeltaThetaM k	Np(rpm)	OP1	OP2	OP3	OP4
7.0	0.958	1289	853.3	423.3	438.0	590	0	1	0	0
7.5	0.962	1292	853.4	423.4	439.0	590	0	1	0	0
8.0	0.966	1295	853.5	423.5	440.0	590	0	1	0	0
9.0	0.970	1298	853.7	423.7	441.0	590	0	1	0	0
10.0	0.972	1301	853.9	423.9	442.0	590	0	1	0	0
10.5	0.976	1304	854.0	424.0	444.0	590	0	1	0	0
11.0	0.978	1307	854.2	424.2	445.0	590	0	1	0	0
11.5	0.980	1310	854.4	424.4	446.0	590	0	1	0	0
12.0	0.982	1313	854.5	424.5	448.0	590	0	1	0	0
12.5	0.984	1316	854.6	424.6	450.0	590	0	1	0	0
13.0	0.988	1322	855.0	425.0	452.0	590	0	1	0	0
13.2	0.990	1325	855.4	425.4	454.0	590	0	1	0	0
13.4	0.991	1328	855.8	425.8	455.0	590	0	1	0	0
13.6	0.992	1331	856.2	426.2	456.0	590	0	1	0	0
13.8	0.994	1334	856.6	426.6	458.0	590	0	1	0	0
14.0	0.996	1337	857.0	427.0	459.0	590	0	1	0	0
14.2	0.997	1340	857.4	427.4	460.0	590	0	1	0	0
14.4	0.999	1343	857.8	427.8	461.0	590	0	1	0	0
14.6	1.000	1346	858.2	428.2	462.0	590	0	1	0	0
14.8	1.010	1349	859.0	429.0	463.0	590	0	1	0	0
15.0	1.020	1352	859.4	429.4	464.0	590	0	1	0	0
15.3	1.030	1355	859.8	429.8	465.0	590	0	1	0	0
15.7	1.040	1358	860.2	430.2	467.0	590	0	1	0	0
16.0	1.050	1361	860.6	430.6	468.0	590	0	1	0	0
16.2	1.060	1364	861.0	431.0	469.0	590	0	1	0	0
16.5	1.080	1367	862.2	432.2	470.0	590	0	1	0	0
16.8	1.090	1370	862.6	432.6	471.0	590	0	1	0	0
17.0	1.100	1376	862.8	432.8	472.0	590	0	1	0	0
17.2	1.120	1376	863.0	433.0	473.0	590	0	1	0	0
18.8	1.350	1394	865.0	435.0	482.0	590	0	1	0	0
19.0	1.380	1397	866.4	436.4	484.0	590	0	1	0	0
19.6	1.400	1400	867.0	437.0	485.0	590	0	1	0	0

Table I Continued

Reactivity	P/Q	Linp (Megawatt)	Theta(CSA) k	DeltaTheta(CSA) k	DeltaThetaM k	Np(rpm)	OP1	OP2	OP3	OP4
20.0	1.440	1400	868.0	438.0	488.0	590	0	1	0	0
0.0	0.940	1300	853.0	423.0	433.0	560	0	0	1	0
0.2	0.980	1200	853.0	423.0	433.0	530	0	0	1	0
0.4	1.000	1100	853.0	423.0	433.0	500	0	0	1	0
0.7	1.100	1000	853.0	423.0	433.0	470	0	0	1	0
1.5	1.200	800	853.0	423.0	433.0	440	0	0	1	0
2.0	1.250	700	853.0	423.0	433.0	410	0	0	1	0
2.8	1.300	600	854.0	424.0	434.0	380	0	0	1	0
4.5	1.400	400	854.1	424.1	434.5	350	0	0	1	0
5.2	1.450	300	855.0	425.0	435.0	348	0	0	1	0
6.4	1.500	200	854.5	425.5	435.5	346	0	0	1	0
7.6	1.540	150	856.0	426.0	436.0	344	0	0	1	0
8.4	1.580	125	857.0	427.0	436.5	342	0	0	1	0
10.0	1.600	125	858.0	428.0	437.0	340	0	0	1	0
11.1	1.630	125	859.0	429.0	438.0	334	0	0	1	0
12.0	1.680	125	860.0	430.0	439.0	330	0	0	1	0
12.6	1.700	125	861.0	431.0	440.0	328	0	0	1	0
13.6	1.740	125	863.0	433.0	441.0	326	0	0	1	0
14.0	1.760	125	864.0	434.0	442.0	324	0	0	1	0
14.8	1.780	125	865.0	435.0	443.0	322	0	0	1	0
14.0	1.860	125	880.0	450.0	462.0	322	0	0	1	0
13.6	1.870	125	883.0	453.0	466.0	322	0	0	1	0
13.2	1.880	125	886.0	456.0	468.0	322	0	0	1	0
12.8	1.890	125	890.0	460.0	470.0	322	0	0	1	0
12.4	1.900	125	892.0	462.0	472.0	322	0	0	1	0
12.0	1.900	125	896.0	466.0	476.0	322	0	0	1	0
11.6	1.900	125	900.0	470.0	480.0	322	0	0	1	0
10.4	1.900	125	904.0	476.0	482.0	322	0	0	1	0
10.2	1.900	125	908.0	480.0	486.0	322	0	0	1	0
10.0	1.900	125	912.0	482.0	490.0	322	0	0	1	0
9.2	1.900	125	916.0	486.0	494.0	322	0	0	1	0
9.0	1.900	125	920.0	490.0	498.0	322	0	0	1	0

Table I Continued

Reactivity	P/Q	Linp (Megawatt)	Theta(CSA) k	DeltaTheta(CSA) k	DeltaThetaM k	Np(rpm)	OP1	OP2	OP3	OP4
8.0	1.900	125	926.0	496.0	504.0	322	0	0	1	0
2.0	1.000	1255	853.0	423.0	433.0	560	0	0	0	1
4.0	1.100	1260	853.0	423.0	433.0	322	0	0	0	1
6.0	1.200	1265	853.0	423.0	433.0	322	0	0	0	1
8.0	1.250	1270	853.0	423.0	433.0	322	0	0	0	1
9.0	1.300	1275	853.0	423.0	433.0	322	0	0	0	1
10.0	1.350	1280	853.0	423.0	433.0	322	0	0	0	1
13.0	1.450	1300	854.0	424.0	433.0	322	0	0	0	1
14.0	1.500	1200	854.1	424.1	433.0	322	0	0	0	1
15.0	1.540	1100	855.0	425.0	434.5	322	0	0	0	1
16.0	1.600	1000	854.5	425.5	434.5	322	0	0	0	1
17.0	1.600	900	856.0	426.0	434.5	322	0	0	0	1
18.0	1.600	800	857.0	427.0	435.0	322	0	0	0	1
19.0	1.600	700	858.0	428.0	436.0	322	0	0	0	1
20.0	1.600	600	859.0	429.0	438.0	322	0	0	0	1
21.0	1.600	500	860.0	430.0	440.0	322	0	0	0	1
21.0	1.600	450	861.0	431.0	441.0	322	0	0	0	1
21.0	1.600	350	862.0	432.0	441.5	322	0	0	0	1
20.0	1.600	250	863.0	423.0	442.0	322	0	0	0	1
17.0	1.600	125	872.0	442.0	451.0	322	0	0	0	1
16.5	1.600	125	874.0	444.0	452.0	322	0	0	0	1
16.0	1.600	125	877.0	447.0	453.0	322	0	0	0	1
15.5	1.600	125	880.0	450.0	454.0	322	0	0	0	1
15.0	1.600	125	883.0	453.0	456.0	322	0	0	0	1
14.5	1.600	125	886.0	456.0	457.0	322	0	0	0	1
14.0	1.600	125	890.0	460.0	459.0	322	0	0	0	1
13.5	1.600	125	892.0	462.0	460.0	322	0	0	0	1
13.0	1.600	125	896.0	466.0	461.0	322	0	0	0	1
12.0	1.600	125	904.0	476.0	463.0	322	0	0	0	1
11.5	1.600	125	908.0	480.0	465.0	322	0	0	0	1
10.0	1.600	125	912.0	482.0	466.0	322	0	0	0	1
9.0	1.600	125	916.0	486.0	467.0	322	0	0	0	1

Table I Continued

Reactivity	P/Q	Linp (Megawatt)	Theta(CSA) k	DeltaTheta(CSA) k	DeltaThetaM k	Np(rpm)	OP1	OP2	OP3	OP 4
8.0	1.600	125	920.0	490.0	469.0	322	0	0	0	1
7.0	1.600	125	922.0	492.0	170.0	322	0	0	0	1

ρ= reactivity

P/Q=power to flow ratio
Lin P= Linear Power
θ_{CSA}=central subassembly temperature

N_P=pump Speed in revolution per minute in ^{0}K
$\Delta\theta_{CSA}$= change in central subassembly temperature in ^{0}K

$\Delta\theta_M$ = mean core temperature in ^{0}K
OP=Output
DOP=Desired Output
AOP=Actual Output

Table J presents the predicted events(primary sodium pump trip, uncontrolled withdrawal of control and safety rod, primary pump seizure and primary pipe rupture) output with respect to effective (Safety Control Rod Accelerated Movement) SCRAM parameters for test samples

Table J

ρ	P/Q	Lin P	$\theta_{CSA}\,^{0}\mathrm{K}$	$\Delta\theta_{CSA}\,^{0}\mathrm{K}$	$\Delta\theta_{CSA}\,^{0}\mathrm{K}$	Np	DOP1	DOP2	DOP3	DOP4	AOP1	AOP2	AOP3	AOP4
1.5	0.95	1310	855.7	425.7	436.5	535	1	0	0	0	0.8846	-0.1184	0.2154	0.2146
1.5	0.96	1250	857.0	427.0	437.5	525	1	0	0	0	0.8782	-0.1188	0.2613	0.1931
1.5	0.97	1200	858.0	428.0	438.0	520	1	0	0	0	0.8528	-0.1175	0.2975	0.1589
1.5	0.98	1000	859.5	429.5	438.5	515	1	0	0	0	0.7481	-0.0726	0.2506	0.0313
1.5	0.99	800	861.0	431.0	439.0	510	1	0	0	0	0.7866	0.1067	0.1006	-0.019
17.4	1.16	1379	863.2	433.2	474.0	590	0	1	0	0	-0.0266	1.0133	0.0282	-0.0092
17.7	1.20	1382	863.4	433.4	476.0	590	0	1	0	0	-0.0267	1.0222	0.0283	-0.0123
18.0	1.24	1385	863.6	433.6	479.0	590	0	1	0	0	-0.0269	1.0299	0.0282	-0.0151
18.2	1.28	1388	863.8	433.8	480.0	590	0	1	0	0	-0.0268	1.0382	0.0271	-0.0181
18.4	1.30	1391	864.6	434.6	481.0	590	0	1	0	0	-0.0272	1.0420	0.0293	-0.0201
15.0	1.80	125	867.0	437.0	445.0	322	0	0	1	0	-0.1241	0.0011	0.8744	0.0559
15.4	1.82	125	868.0	438.0	447.0	322	0	0	1	0	-0.1238	0.0374	0.8265	0.0614
15.3	1.83	125	870.0	440.0	450.0	322	0	0	1	0	-0.1239	0.0472	0.836	0.0547
14.9	1.85	125	874.0	444.0	456.0	322	0	0	1	0	-0.1242	0.0474	0.9021	0.0284
14.8	1.85	125	877.0	447.0	458.0	322	0	0	1	0	-0.1238	0.0759	0.837	0.0488

Table J continued

ρ	P/Q	Lin P	$\theta_{CSA}\,{}^0\mathrm{K}$	$\Delta\theta_{CSA}\,{}^0\mathrm{K}$	$\Delta\theta_{CSA}\,{}^0\mathrm{K}$	Np	DOP1	DOP2	DOP3	DOP4	AOP1	AOP2	AOP3	AOP4
19.5	1.60	175	864.0	434.0	443.0	322	0	0	0	1	-0.0345	-0.0797	0.003	1.0446
19.0	1.60	125	865.0	435.0	444.0	322	0	0	0	1	-0.0367	-0.0460	-0.0139	1.0397
18.5	1.60	125	867.0	437.0	446.0	322	0	0	0	1	-0.0329	-0.0377	-0.0177	1.0304
18.0	1.60	125	868.0	438.0	447.0	322	0	0	0	1	-0.0256	-0.0275	-0.0230	1.0105
17.5	1.60	125	870.0	440.0	449.0	322	0	0	0	1	-0.0222	-0.0191	-0.0260	0.9990

PUBLICATIONS

List of Publications

Journals

- **Subhra Rani Patra***, R.Jehadeesan, S.Rajeswari, S.A.V Satya Murty, Artificial Neural Network model for IHX of Nuclear Reactor, *International Journal of Computer Applications,* 1 2010 65-72
- **Subhra Rani Patra***, R. Jehadeesan, S. Rajeswari, Indranil Banerjee, S.A.V Satya Murty, G. Padmakumar, M. Sai Baba, Neural Network modeling for evaluating Sodium temperature of Intermediate Heat Exchanger of Fast Breeder Reactor, *Advances in Computing* , 2 2012 16-22
- **Subhra Rani Patra***, R. Jehadeesan, H. Seetha, T. Jayanthi, S. Rajeswari, S.A.V Satya Murty, M. Sai Baba, Event Identification in Prototype Fast Breeder Reactor Subsystem Using Artificial Neural Network, *International Journal of Applied Information System*, 4 2012 1-9
- **Subhra Rani Patra*,** R. Jehadeesan, S. Rajeswari, S.A.V. Satya Murty, M. Sai Baba, Development of Genetic Algorithm based Artificial Neural Network for Parameter Estimation of Fast Breeder Reactor Subsystem, International Journal of Soft Computing and Engineering 2 2012 87-90
- **Subhra Rani Patra*,** R. Jehadeesan, T.V. Santosh, T. Jayanthi, S. Rajeswari, S.A.V Satya Murty, M. Sai Baba ,Neural Network Model for an Event Detection System in Prototype Fast Breeder Reactor, *Cognitive Computation,* (Under Review)

Conferences Presentations

- **Subhra Rani Patra***, R.Jehadeesan, M.L Jayalal, S.Rajeswari, K Natesan, S.A.V Satya Murty, M Sai Baba, Neural Network based Event Identification for Neutronics System in

PFBR, *5th International Conference on Theoretical, Applied, Computational and Experimental Mechanics (ICTACEM)*, IIT Kharagpur, pp: 247-249 2010

- **Subhra Rani Patra***, R.Jehadeesan, M.L Jayalal, S.Rajeswari, S.A.V Satya Murty, M Sai Baba, Neutronics Power Estimation of Prototype Fast Breeder Reactor using Artificial Neural Network Model, Recent Researches in Mathematical Methods inElectrical Engineering and Computer Science, Gregoire Thomas, Cyril Fleaurant,Thomas Panagopoulos, Emmanuele Chevassus-Lozza (Eds.), *Applied Computing Conference*, Angers, France, pp: 88-94 2011
- **Subhra Rani Patra***, R.Jehadeesan, M.L Jayalal, S.Rajeswari, , S.A.V Satya Murty, Development of Neural Network model of a subsyetm for PFBR Simulator, *1st International Conference on Recent Advances in Mathematical Sciences and Applications (RAMSA)*, G.V.P Engg. College, Vishakhapatnam, Decemember, 2009
- **Subhra Rani Patra***, Sujith Jayadev, R.Jehadeesan, S.Rajeswari, N. M. Meenachi, S.A.V Satya Murty, Neural Network model for a Reactor subsystem using real time data, *2nd International Conference Asian Nuclear Prospects (ANUP),* Mahabalipuram, 2010

THANK YOU