APPLICATION OF ARTIFICIAL NEURAL NETWORKS IN CONCEPTUAL DESIGN OF POST-TENSIONED SLABS

Ph.D. THESIS

GAURAV SANCHETI

2008RST101

DEPARTMENT OF CIVIL ENGINEERING

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

JULY 2016

APPLICATION OF ARTIFICIAL NEURAL NETWORKS IN CONCEPTUAL DESIGN OF POST- TENSIONED SLABS

This thesis is submitted as a partial fulfilment of the
Ph.D. programme in Engineering

GAURAV SANCHETI

2008RST101

DEPARTMENT OF CIVIL ENGINEERING

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

JULY 2016

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

CERTIFICATE

This is to certify that the thesis report entitled **"Application of Artificial Neural Networks in Conceptual Design of Post-Tensioned Slabs"** which is being submitted by **Mr. Gaurav Sancheti, ID. 2008RST101,** for the partial fulfilment of the degree of **Doctor of Philosophy** in Civil Engineering to the **Malaviya National Institute of Technology Jaipur** has been carried out by him under my supervision and guidance.

(Dr. Ravindra Nagar)
Professor,
Department of Civil Engineering
Malaviya National Institute of Technology,
Jaipur- 302017, Rajasthan, India

(Dr. Vinay Agrawal)
Asst. Professor,
Department of Civil Engineering
Malaviya National Institute of Technology,
Jaipur- 302017, Rajasthan, India

Place : Jaipur
Date : 08.07.2016

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY JAIPUR

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled '**Application of Artificial Neural Networks in Conceptual Design of Post-Tensioned Slabs'** in partial fulfillment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Civil Engineering, Malaviya National Institute of Technology Jaipur is an authentic record of my own work carried out at Department of Civil Engineering during a period from July, 2008 to June, 2014 under the supervision of Dr. Ravindra Nagar, Professor of Civil Engineering Department and Dr. Vinay Agrawal, Assistant Professor of Civil Engineering Department, Malaviya National Institute of Technology Jaipur.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institute.

Date: 08.07.2016 (**Gaurav Sancheti**)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Dr. Ravindra Nagar)
Professor,
Civil Engg. Department
MNIT Jaipur
Jaipur – 302017, Rajasthan, India
Supervisor

(Dr. Vinay Agrawal)
Asst. Professor,
Civil Engg. Department
MNIT Jaipur
Jaipur – 302017, Rajasthan, India
Supervisor

Place : Jaipur
Date : 08.07.2016

Acknowledgement

I pay my sincere regards and acknowledge Dr. Ravindra Nagar, Professor, Department of Civil Engineering, Malaviya National Institute of Technology Jaipur, for his valuable guidance without which this research work would not have been a success. Apart from being my supervisor, he was always there when I needed him. He taught me that for going deeper into the field, I must start playing with things and must not stop till the last stage is conquered. He always kept my moral high. From his busy schedule, he always spared time for assessing the progress of my work. His wide knowledge regarding the subject helped me in writing this thesis. I thank him from bottom of my heart for his inspiration and guidance he showered on me. I am also grateful to Dr. Vinay Agrawal, Assistant Professor, Department of Civil Engineering, Malaviya National Institute of Technology Jaipur, for providing me with his guidance and time.

I am extremely grateful to Prof.A.B.Gupta, who always kept my moral high and helped me out whenever I needed him. I am also thankful to members, DREC, for their support and guidelines regarding my thesis work. I also pay my sincere regards to Prof.K.C.Jain for helping me out in analysis of my thesis results.

I deeply acknowledge Er. Vinay Chandwani, Executive Engineer, Water Resourse Department, Rajasthan, Jaipur, for his valuable guidance. He helped me out in developing soft computing skills in MATLAB and Microsoft EXCEL, which are the basic programs on which my thesis completely depended. He taught me to develop such programs which would be fast and hence will save a lot of time in preparation of my results. I once again thank him for his help and support.

I am also thankful to Prof. Sudhir Kumar, Prof.B.L.Swami, Prof. Y.P. Mathur, Prof.Gunwant Sharma, Prof. Rohit Goyal, Prof. M. K. Shrimali, Prof.R.C.Gupta, Dr.J.K. Jain, Dr. Sanjay Bhatter, Dr. S.K Tiwari, Dr. Pawan Kalla, Dr. Arun Gaur, Dr.Nivedita Kaul, Shri.Rajesh Gupta, Dr. Sandeep Chaudhary, Dr. Mahesh Jat, Dr.Mahendra Chaudhary, Dr.Sumit Khandelwal, and all the other faculty members of Malaviya National Institute of Technology Jaipur.

I am grateful to the Director, MNIT Jaipur; Head, Civil Engineering Department; Dean R&D and Dean Academics, for providing me with the environment and space to carry out my research work.

I give my thanks to Sh. Rajesh Saxena, office in charge, Civil Engineering Department, MNIT Jaipur, Ramjilal Meena and Sher Singh who were always ready to extend me every possible help throughout my work.

From the bottom of my heart, I would like to thanks my father Shri. Shanti Swaroop Sancheti and mother, Smt. Indu Sancheti, the unforeseen effort and patience of whom, carries much weightage. The most valued efforts and guidance from my sister, Dr.Garima Sancheti, provided me with the urge of doing the best work. Her contribution towards my research work is hard to be stated here in words. I greatly acknowledge her support. I acknowledge here the tolerance and patience of the kids, Dhairya and Isha, which they kept during the progress of my research work. Lastly but not the least, I would like to acknowledge the efforts made by my wife, Paridhi Sancheti, in providing me with the environment to carry on my research work at home with an ease.

Finally, I would like to thank all those people who helped, guided and supported me in my study. Thankyou all!

Dated: 08.07.2016 **(Gaurav Sancheti)**

ABSTRACT

Conceptual design stage is a phase where the initial decisions are made without actually performing the detailed design. Whereas today several advance softwares are being extensively used for detailed engineering designs, there is no tool available with the designer at the conceptual stage of design. In the presented research work, an attempt has been made for developing such a tool which could help in giving conceptual engineering solutions at preliminary stage of design. This stage of conceptual design holds high significance as the major design parameters for any project are decided and fixed, by far and large, in this stage only. Results obtained at conceptual design stage can then be used to determine the quantity and cost of major project constraints. Design in this stage governs the overall economy of the project.

For achieving the said goal, Artificial Neural Networks (ANNs) have been employed. ANNs can be explained as a network of small computational units (artificial neurons) that resemble biological nervous systems in living organisms. These artificial neurons have high computation power. Each computational unit is interconnected to every other unit in the adjacent layer and hence a network of neurons is formed. In such a system of network, information or data is absorbed from the input values. This information is then continuously processed by the neurons of one layer and then passed on to another layer of neurons to be processed. This is how the output nearing to the target output is achieved.

In this study, effort has been made to model a neural network which would be capable of giving the conceptual design of PT slabs in terms of deflection and post tensioning steel requirement for various slab configurations. Design of three span post tensioned slabs has been performed using the standard software. As a part of research work, both single layered and double layered networks have been developed. The number of hidden layer neurons in case of single layered networks is taken as 5, 10, 15 and 20. On the other hand, the number of hidden layer neurons for the double layer networks is taken as 5, 7 and 9 in both the layers. Log-Sigmoid and Tan-Sigmoid are taken as the transfer functions with the linear function for the output layer. The training functions considered for training the networks are Levenberg-Marquardt

training algorithm (trainLM) and Resilient Backpropagation algorithm (trainRP). A large number of neural networks have been developed with all possible network architectures. These developed neural network models are trained up to 1000 epochs starting from 100 and with an increment of 100 epochs every time after recording the mean square error.

In order to evaluate the efficiency of the network performance, four different types of validation techniques are employed. The first one is the “Resubstitution validation technique”, where the entire database undergoes training and testing. The second technique is the “Holdout validation technique”, in which a part of data is kept separate for testing and the remaining database undergoes training. The third technique is the “Three way data split technique” in which one part of the database is kept for training, second part for validation and third part for testing of database. Fourth validation technique is the “K-fold validation technique” (K taken as 10) in which the entire database is divided into 10 parts out of which 10% data is kept for testing and remaining 90% database undergoes training. Each network undergoes validation at 1000 epochs and hence the best network is chosen having minimum mean square error (MSE) as compared to the MSE given by all other developed neural networks.

This research demonstrates that the conceptual design aids can be successfully developed for PT slabs, using ANNs. The research also outlines detailed methodology for validation of ANNs based conceptual design tools for their acceptance by the engineering community.

CONTENTS

CONTENTS

CONTENTS

CONTENTS

List of Tables

List of Tables

List of Figures

List of Figures

List of Figures

List of Abbreviations

S.No	Content	
1.	ANN	Artificial Neural Networks
2.	RCC	Reinforced Cement Concrete
3.	PT	Post-Tensioned
4.	MSE	Mean Square Error
5.	AI	Artificial Intelligence
6.	OCR	Optical Character Recognizer
7.	ADALINE	Adaptive Linear Element
8.	LMS	Least Square Error
9.	RBF	Radial Basis Function
10.	RPM	Rotation Per Minute
11.	LSF	Lime Saturation Factor
12.	OPC	Ordinary Portland Cement
13.	GGBS	Ground Granulated Blast furnace Slag
14.	CO_2	Carbon di Oxide
15.	$PM_{2.5}$	Particulate Matter with diameter 2.5 micrometer or less
16.	HCHO	Formaldehyde
17.	VOC’s	Volatile Organic Compounds
18.	RH	Relative Humidity
19.	POPs	Persistent Organic Pollutants
20.	PRAN	Probabilistic Resource Allocating Network
21.	PBLS	Polynomial-Based Layer Separation Algorithm

List of Abbreviations

S.No	Content	
22.	AASHTO	American Association of State Highway and Transportation Officials
23.	LRFD	Load and Resistance Factor Design
24.	RMS	Root Mean Square Error
25.	LL	Live Load
26.	LM	Levenberg-Marquardt algorithm
27.	RP	Resilient algorithm
28.	SSE	Sum of Square Error
29.	ACF	Auto Correlation Functions
30.	CCF	Cross Correlation Functions
31.	SDVR	Second Derivative of Validation error based Regularization algorithm

CHAPTER-1

Introduction

1.1 Introduction to conceptual design

The preliminary stage of design, where the design constraints are set using some thumb rule or some experience, to arrive at some appropriate decision, is known as the conceptual stage of design. The tentative results obtained at this conceptual stage, may be in the form of labour, material or machinery requirement, which governs the overall economy of the project. Whereas there are tools to perform detailed engineering designs, there is hardly any aid available for the structural designers at conceptual stage of design. Most of the important decisions are made at the conceptual design stage and therefore there is a need to empower structural engineers with a tool, which can help take decisions at initial stage of design. Once the important design parameters are determined close to actual values, design engineers would have the idea about the total quantity of the basic construction materials such as the amount of concrete, formwork, etc., required during the construction. This will help in determining total cost of the project quiet practically.

This is also important to have a control over the financial budget sanctioned for a particular project. At the conceptual design stage, most of the information or data made available regarding the project are incomplete or improper. At the same time, to meet the desired requirements of design, an engineer must have some approximate values of design parameters before the final decision could be made. These values influence the initial design of a structure at conceptual stage. Therefore, the decision body plays a very important role with human intelligence and experience clubbed together with some computational skills, at conceptual stage of design.

Previously researchers, (Aktan et al. 1984, Eldin et al. 1988, etc.) made great attempts for such knowledge-based systems, which could be helpful at conceptual stage of design. However, the task of collecting information from a knowledge bank was not only difficult but also time consuming. For this reason, practical use of these experiments could not be taken up. Researchers (William et al. 1992, De Paoli et al. 1996, Wang et al. 1997, Elazouni et al. 1997, Rivard et al. 2000, Rafiq et al. 2001, Grierson et al. 2002, Honigmann et al. 2003, Sisk et al. 2003, Ye et al. 2006, Park et al.

2007, Keller et al. 2009, Calvi. et al. 2010, Asmar et al. 2011, Quirant et al. 2011, Sunindijo et al. 2013 and many more) have taken up inductive learning techniques. The rules are generalized by conceptual design technique using the design data which is analogous to a knowledge bank. Although, these researches had shown a lot of promise, however a reliable and intelligent software tool that can be used by structural engineers for practical problems has still not developed. Present research attempts to bridge this gap, particularly in the field of conceptual design of PT slabs.

Methodology adopted by these recent researchers is the use of computers as 'decision makers'. Today computers are needed for day-to-day activities. Whether, general business or engineering or any other field, computers now a days have become the most important part of our life. "Although computers are used to model a variety of engineering activities, currently the main focus of computer applications is areas with well-defined rules. Activities related to the conceptual stage of the design process are generally untouched by computers" (Rafiq et al.2001). Efforts are being made worldwide to utilize the power of computer in intelligent decision making in engineering applications, using past knowledge as being used by an experienced engineer.

In the field of structural engineering, designers have to deal with several complex structural geometry and as such their analysis and design at conceptual stage, is really a challenge. This research work presents a novel idea, which focuses on the development of a decision support tool using ANNs. This decision support tool is to use by the expert community of structural designers for arriving at some decision conceptually for a particular structural geometry. One such type of structural geometry problem analyzed in this research work is the analysis of PT slabs. Although several sophisticated software are available for detail designing, but does not help when quick and accurate results are desired. Results show that there can be a significant gain when a predictive decision support tool is used at the conceptual design phase [Gil P.J.S. and Ferreira I.M.L., 2012].

1.2 Conceptual Design Tool: Artificial Neural Networks (ANN)

Artificial Neural Networks (ANNs) can be thought of a network of small computational units (neurons) that resembles biological nervous systems in living organisms. These artificial neurons have high computation power. Each computational unit is interconnected to every other unit in the adjacent layer and hence a network of neurons is formed known as Artificial Neural Network. In such a system of network, information or data is absorbed from the input values. This information is then continuously processed by the neurons of one layer and then passed on to another layer of neurons to be processed. It means that, each parameter is taken care of by several neurons. This is how the knowledge is distributed over the entire network and some output nearing to the target output is achieved.

The interest in neural networks in the field of structural engineering started early towards nineties. Until now, neural networks have been applied in various structural engineering problems such as prediction of deflection of slabs and beams, shear and flexural capacity of beams, conceptual design of buildings, optimization problems, etc.

As in this research work, idea is to identify such a tool which can help a design engineer at conceptual stage of design, ANNs can be best utilized. Its important characteristics such as learning, generalization, information processing, etc., have been used by various researchers (Ashour et al. 2005, Agrawal et al. 2011, Bilgil et al. 2008, Chao et al. 1994, Chen et al. 1999, Özcan et al. 2009, Hirooka et al. 1996, Jingling et al. 2012, Kim et al. 2004, Kerh et al. 2000, Liu et al. 2006, and many others) successfully. Hence ANNs are adopted in this research work of conceptual design of post tensioned slabs as its importance have been proved in the research problem relating to decision making and other complex design problems such as the determination of PT slab deflection and quantity of PT steel required at conceptual stage of design. Among all type of neural network architectures, multilayer feed forward back propagation neural network is most commonly used. In this network the perceptron is trained in forward direction only. The error, i.e., the difference between the actual and target output value, is propagated backwards using a gradient descent technique known as back propagation.

The basic structure of multi-layer backpropagation neural network is the assembly of input layer, hidden layer and output layer. A network cannot have more than one input and one output layers, whereas hidden layers can be more than one. All of these layers consist of neurons and these neurons are interconnected with each other to form a network. When the error is back propagated in each epoch (iteration) cycle, the weights are adjusted and again the forward propagation takes place. The connection weights are revised using generalized delta rule. This way a cycle of epochs continues till the no. of epochs are completed or the error comes to a desired level of satisfaction.

In this research work efforts have been made to develop a neural network application that will be more towards the practical field application and would contribute towards the preliminary stage of design of post-tensioned slabs.

1.3 Post-Tensioned Slabs

Today, post tensioned concrete is not a new name for construction industry. It has added several dimensions to the prevailing construction practices. The relatively thin slab depths and resulting lighter weight structures and smaller floor to floor heights are foremost among the prestressed flat-plate system's advantages (Gary M. Kosut 1983). Growth in this type of construction has occurred over the past 20 years in competition with other structural systems (Ned H. Burns 1985). In India, they have become popular mainly in the last decade.

The basic fundamental behind such a concept is to balance the service loads coming on the structural member by the forces of opposing nature induced in it by the applied prestress. Post tensioned flat slabs are being used more and more in building floors and bridge decks. More and more research works are being carried out on the post tensioned bridge decks and building slabs.

In the present study various configurations of post tensioned slabs have been considered for analysis using ANNs. The main area of focus in case of flat plates is deflection. This is due to the fact that these post-tensioned slabs are comparatively thin as compared to RCC slabs. Spans of these post tensioned slabs are generally larger which attracts deflection. Hence it is also required to estimate the quantity of post tensioning steel required to keep deflection under permissible limits. In order to achieve a balance between the amount of post tensioning steel required and the deflection in post tensioned slabs, various factors affecting these two parameters have been identified. These are the span, depth of slab, live load, column size and grade of concrete and these factors have been taken as the five inputs of the neural network. Amount of post-tensioning steel required in terms of weight and the deflection are taken as the two outputs of the neural network.

1.4 Research Objectives

For research on "Application of Artificial Neural Networks in Conceptual Design of Post-Tensioned Slabs" the following objectives were identified:

a) To study the feasibility of using Artificial Neural Networks for conceptual design of post-tensioned slabs.

b) To propose an architecture for a neural network for conceptual design of post-tensioned slabs. Components of network architecture includes, training algorithms, activation functions, number of hidden layers, number of nodes in hidden layers and number of epochs.

c) To propose a validation technique for validating the best performing network architecture. Four different validation techniques, namely, resubstitution, holdout, three-way data split and k-fold cross validation technique have been applied on the developed networks and their performance have been evaluated in terms of mean square error.

1.5 Thesis organization

The thesis has been organized in seven chapters:

Chapter-1: This chapter introduces the basics of all the components involved in the research work. A brief description of the problem, the application of ANNs and the research objectives have been summarised in this chapter.

Chapter-2: This chapter gives the basic idea about the definition, principles, types, components and working of artificial neural networks and Backpropagation algorithm. Detailed notes of the background of neural networks and about its architecture have been discussed. This chapter also gives the various positive and negative aspects of the artificial neural networks.

Chapter-3: In this chapter literature survey of the previously work done on artificial neural networks have been presented. Neural networks have been applied in number of civil and structural engineering problems and there are several journals that have encouraged this type of research on neural networks. Review of such papers has been presented in this chapter.

Chapter-4: Describes the method statement for the research work done including design of Post Tensioned slab at conceptual stage. Different training algorithms, transfer functions have also been discussed.

Chapter-5: This chapter basically deals with the selection of best neural model for the specified conceptual design problem. Various techniques of validating neural network such as, resubstitution, three-way data split, holdout and k-fold cross validation and the results as obtained by these techniques have been discussed.

Chapter-6: This chapter compares the result obtained by using the Artificial Neural Networks with the results obtained by standard Statistical Techniques. Both linear regression and polynomial regression models are considered for comparison.

Chapter-7: This chapter concludes the thesis by pointing out the facts in support of the objectives set out in the first chapter. In addition to this further research scope on the basis of present has also been discussed.

Annexures: Some of the major facts that formed the basis of this thesis have been presented in the Annexures as shown below:

Annexure I : Design Database

Annexure II : Comparison of results as obtained from software and as calculated manually.

Annexure III : Design of Post tensioned slab-an example

Annexure IV (A) : Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using Resubstitution validation technique.

Annexure IV (B) **:** Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using Holdout validation technique.

Annexure IV (C) **:** Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using Three-way data split validation technique.

Annexure IV (D) **:** Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using k-fold cross validation technique.

1.6 Summary

This chapter focuses mainly on the importance for conceptual design in engineering domain. From the above discussion it is clear that some helping tool is required at the preliminary stage of design to assist the designer for setting up primary design parameters. The suggested tool for conceptual stage design is from the field of artificial intelligence, i.e., Artificial Neural Networks (ANNs). Though ANNs have been utilized in several civil and structural engineering domains, but still its applications in design of PT slabs have not yet been analysed. Hence, this research work is taken up with the clear intention to prove the reliability of ANNs over other statistical techniques for analysing the engineering data.

CHAPTER-2

Artificial Neural Networks

2.1 Introduction

Intelligence in humans can be associated with the prediction of most suitable or most advantageous way out of the real world problems, and doing the needful accordingly for the same. If the same intelligence can be developed in machines, so that they can learn and give their own judgment, it will be a new beginning in the area of artificial intelligence, the benefits from which are hard to be thought today. This advancement will not only save time but will also help in reducing the mental and physical strain in humans, which can be put to some other important work. Artificial intelligence is the study field where the focus is made on building such program based machines, which could take their own decisions under complex situations, giving such results, which are quite close to actual results. Artificial Neural Network (ANNs) are one of the most advanced tool of Artificial Intelligence (AI) community. It has great applicability in general applications, in almost all the fields, including science and technology.

Concept of ANNs has been taken from the functioning of the human brain. As the name suggests, ANNs are such networks which are comprised of artificial neurons. The structure of an artificial neuron resembles the structure of a biological nervous cell. The fundamental idea of the neural networks approach is to describe a complex system using combinations of many small units [Jung S. et al. 2004]. The artificial neurons or nodes are highly interconnected with each other forming a complex network structure. These are powerful networks which do not require any programming support. Self-learning mechanism is one of the main characteristics of neural networks. These networks consist of a number of small but powerful computing units which are interconnected with each other through weighted links. These computing units are the processors which processes the information given to the network. The processed information is passed on to the next level of computing units through these weighted links only. An important aspect of neural networks is their capability to construct non-linear relationships between the input data and target outputs [Li Q.S., et al., 2000].

Many research works are in progress round the globe which are applying ANNs to several real world practical problems. These artificial networks work in such a way that maximizes their performance by ‘adapting’ from the surrounding environment and hence the success chances of their prediction are increased. It would be quite correct to say that ANNs can be considered as the technology of coming time with powerful computing capabilities in the entire world.

Because of their extremely parallel architecture, a neural network has the ability of processing information much quicker than traditional computational practices. Since the structure of neural network is inspired by the structure of brain, these networks have great generalization power and can tolerate noisy data to a great extent.

The power of Neural Networks to learn efficiently from the fed experience and to take reliable decisions makes it as good as a modern day computing technology. Since Neural Networks needs experience in form of data for learning process, therefore by providing the network with reasonable number of data, Neural Network can be made as a 'Learned Network’. This network, when trained optimally, can be used for obtaining effective results for a given problem. With more and more of experience that is fed in the selected neural model, its precision level can be further increased.

2.2 Background

It all started with a thought of building or creating a neuron, which is the basic element of a neural network. As per the available literatures, in the year 1943, neurophysiologist Warren McCulloch and the logician Walter Pitts drafted the first ever ANN model which was capable of computing arithmetic and logical functions. This was the simplest model consisting of two input neurons whereas there was only one neuron in the output layer. Thus, this was the year which may be marked as a year in which the initiative for the development of ANN has taken place. It was stated by McCulloch and Pitts that, neurons have some threshold value and the neurons fire only when this threshold value is surpassed. This Neural Network was designed to perform logically by keeping the weights fixed. Therefore they could perform arithmetic and logical computing, but at the same time these networks were not able to learn from given example or experience in form of data. The logic circuit of today is the same McCulloch and Pitts network. To overcome the problem of learning, Donald Hebb gave ‘Hebbian

Learning Rule' in the year 1949. Hebbian Learning Rule or theory described that how the weights can be modified amongst the neurons on the basis of information stored in the connection between them. Any two cells or systems of cells that are repeatedly active at the same time will tend to become 'associated', so that activity in one facilitates activity in the other. This theory was a big contribution towards the theory of neural networks as this theory made networks to learn. Inspired by this theory, Marvin Minsky, gave some great researches to the society. His doctorate thesis, "Theory of Neural-Analog Reinforcement Systems and its Application to the Brain-Model Problem", in 1954, was based on the neural network research work. His scientific paper, "Steps towards Artificial Intelligence" was the primary paper to discuss Neural Networks as a part of Artificial Intelligence in depth. A learning machine was developed by Marvin Minsky, in which weights were automatically adopted. With the very fast going research programs, and much advancement in the neural networks, the year 1957 witness the first ever successful neuron based computer. The developers of this neuro computer (the Mark I Perceptron) were Rosenblatt and Wightman. This device was primarily developed for handling classification problems. Optical character recognizer (OCR) was the first demonstration by this neuro computer. Out of the various advancements of neural networks, Widrow and Hoff, in 1960, developed a different type of processing unit, ADALINE (Adaptive Linear Element), with a powerful learning rule. Based on the McCulloch–Pitts neuron, it was a neural network with single layer comprising of a weight, a bias and a summation function. The learning rule for ADALINE converge it to the least squares error.

The year 1969 brought a black day in the history of neural networks. In this year, the publication of the book by Minsky and Papert with the name 'Perceptrons', brought the neural network research to a sudden stop. This book reflected the limitations of single layered Neural Networks compared to systems with multiple layers. It was proved in this book mathematically, that the perceptron are able to solve only those problems which are linearly separable. These perceptrons were a complete failure for application in EXCLUSIVE OR (XOR) logic function and other similar fundamental logic based functions. This resulted in the collapse of interest of the researchers towards Neural Networks and it almost put an end on the research for Neural Networks. The funding for neural network research was also completely stopped.

Paul J. Werbos in 1974 submitted his Ph.D. thesis, 'Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences' at Harvard university. In his research work Werbos used backpropagation technique for making the networks learn. Some major developments took up the pace in the starting years of 1980's which marked the interest in mind of the researchers for the ANNs. Energy analysis of feedback neural networks was one such development by John Hopfield in the year 1982. As per Hopfield there exist states of equilibrium in a feedback network when the weights associated with the network are symmetrical. The backpropagation was again discovered independently by Parker in the year 1985 and by Rumelhart and McClelland in the year 1986. It ensured a possibility for adjusting the weights in a multilayer feed forward neural network in a systematic manner of learning. This was the basis of, what is called, error backpropagation. With these achievements, more and more research was carried out on neural networks in almost all the fields of technology.

Lots of research and lot more conferences on the application of neural networks are being organized today. Presently neural networks are being applied in variety of fields such as banking, industries, electronics, auto-motives, medical, engineering, telecommunication, etc. and lot more. ANNs can be used in several problems concerning concrete technology even when the exact relationship between the inputs and outputs are not known [Sancheti G. et al, 2009]. No doubt that Artificial Neural Networks are such a technology which would help in giving logical decisions with minimum effort and time. It is a tool which saves time and gives precise results [Sancheti G. et al, 2009]. As the commercial application of Artificial Neural Networks will increase it will open an era for a new technology, 'the thinking machine' technology.

2.3 Biological Neural Networks

Human brain is the most versatile object of its own kind. It empowers the humans to think and take decisions of their own which are beneficial for him. It is a question of great interest that what makes the brain to take such logical decisions? What type of processing is being done in the brain which helps us to learn and remember? It had been already stated that ANNs are such networks whose structure resembles with the structure of the human brain. In order to accelerate the performance of ANNs and to map the processing of human brain to the ANNs, it becomes quiet essential for us to have some knowledge of the structure of human brain and the way in which it works.

Biological neural networks are a cluster of highly interconnected physical neurons which are present in the human brain. Fig 2.1 represents the structure of a biological neural network. Information in human brain is processed through biological neural networks. These networks consist of a very large number of neurons (living cells). The neurons are in the order of 1011 taken on an average. These neurons are connected with each other through synapse. If we talk about the cell body, it consists of three main parts. These are the cell body; which is also known as soma, the dendrites, and the axon. Cell body is the central unit which contains the nucleus. The structure of dendrites is like the branches of a tree and they carry electric signals to the neuron. Also they act as receptive networks of nerve fibers. Axon is in the form of long fiber and there is only one axon per nucleus. As the end of axon is divided into several small parts, it is capable to transfer the electric signals to various other cells.

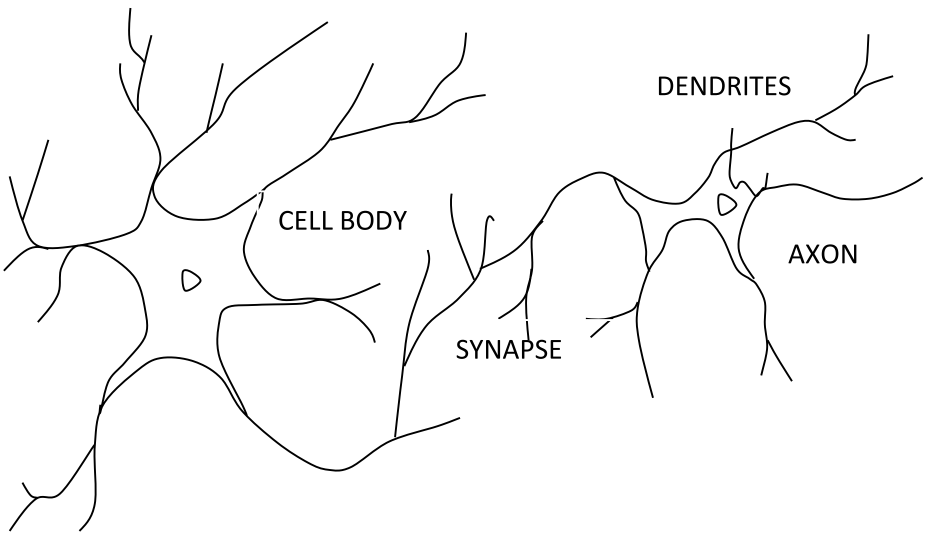


Fig 2.1: Biological Neuron

The signals from the cell body or the soma are carried by the axon to the other cells. Where the Axon of one particular cell meets the dendrite of other distinct cell, synapse is formed. The bond strength of synapses as well as the arrangement of neurons in a network is responsible for the establishment of a neural network function. A particular axon is associated to a number of synapses which in turn are connected to the large number of other neurons. Signals from one cell to another are transmitted at synapses involving a chemical reaction of complex nature. Due to this reaction, fluctuation of potential inside the cell body takes place.

When this potential becomes equal to the threshold value, a small pulse as a result of electric activity is formed which brings the cell to a state of firing. These pulses are often referred as electric signals. The pulses are of fixed duration and strength. The electrical activity is limited to the inside the neuron only and the chemical activity process outside at the synapse. The dendrites sense the signals from the neurons and hence work as receptors. The pulses generated as a result of electrical activity is transmitted to the cell by axon. Fig 2.2 shows a typical diagram of synapse along with its functional units.

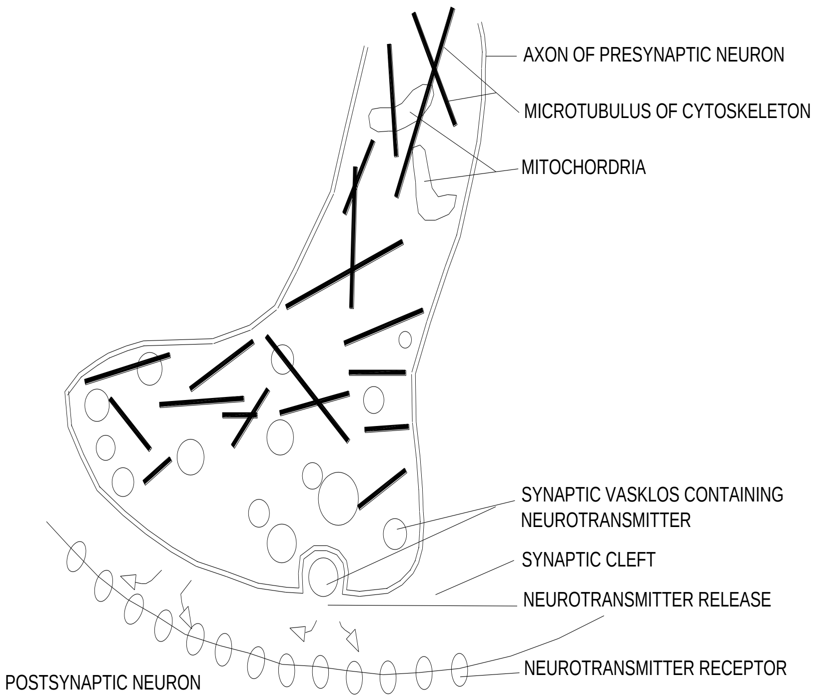


Fig 2.2: Chemical Signals at the Synapse

The working of Artificial Neural Networks is by and large similar to that of the biological nervous system. Neurons process the information by receiving the signals from different layers of neurons preceding it and then pass these signals to the next layer of neurons. Whenever there is a change of weight, it corresponds to the activity in the synaptic connection. As the weights are modified when the network undergoes a training process, in a similar way the strength modulus of synapse also gets continuously updated and gets modified accordingly.

Today the most complex and heavily concentrated network known is the human brain. Artificial Neural Networks can never touch the vicinity of complexity of the brain. However Artificial Neural Networks have been modeled on the basis of biological nervous system and hence there are certain similarities between the two networks. Taking, first, such units which are responsible for the structural composition of the network, it is seen that these units are basically the computing units which are highly interconnected. Also the function of the network is also determined by the connection between the neurons. Due to the large parallel structure that the biological neural network has, operation of all the neurons is possible at the same time. For this reason brain is able to perform the given complex task much faster and logically than the conventional computers. Artificial Neural Networks too consists of parallel system but the number of neurons is negligible as compared to the number of neurons in the biological nervous system. Artificial Neural Networks are in general focusing on problems relating generalizing and fault tolerance.

Table 2.1: Analogy between Biological Neural Nets and Artificial Neural Nets

Human Brain	**Artificial Neural Networks**
Neuron	Processing Element
Dendrites	Link Function (input)
Cell Body	Transfer Function
Axons	Link Function (output)
Synapses	Weights
Potential	Weighted sum
Threshold	Bias
Electric Signals	Activation

It is now very clear from above discussion that since Artificial Neural Networks have been carved out from the concept of biological neural networks, there definitely is an analogy between artificial neural networks and biological neural networks. Table 2.1 shows the analogy between Biological Neural Networks and Artificial Neural Networks.

2.4 Back Propagation Neural Network

The book by Minsky and Papert, ‘Preceptrons’, in 1969, proved that the preceptrons were not able to solve non-linear problems, which ultimately brought the research on ANNs to almost an end. In 1986, Rumelhart and McClelland developed backpropagation algorithm, which gave new life to the neural networks and was responsible for their resurgence. Backpropagation could get its name only after 1974 by the efforts made by Paul Werbos) but it were Rumelhart and McClelland who were able to apply them in ANNs for updating the network weights successfully. This theory provided neural networks with the most important characteristic which was the ability to learn and get trained.

Backpropagation may be defined as a methodology incorporated to make ANNs to learn the problem statics and perform to a desired level of accuracy as been expected by the user. The basic concept involved in the backpropagation process is that the difference between the output and the target values is sent back or is back-propagated and the weights are updated accordingly. The backpropagation algorithm basically involves two phases. The first one is the forward phase where the activations are propagated from the input to the output layer. The second one is the backward phase where the error between the observed actual value and the desired nominal value in the output layer is propagated backwards in order to modify the weights and bias values [Caglar N. 2009]. Among the many different types of ANN, the feed forward, multilayered, supervised neural network with the error backpropagation algorithm, generally known as the backpropagation network, is by far the most commonly applied owing to its simplicity [Kao C.Y. et al. 2003].

Back Propagation Neural Network (BPNN) for quite some time now, is being widely used in the field of civil and structural engineering. Several training algorithms exist but backpropagation commonly provides satisfactory results [Dahou Z. et al. 2009]. These networks are capable of handling problems relating to structural analysis to a high degree of accuracy. Results of these networks are quiet appreciable even in case of noisy data. The training of Neural Networks improves continuously with the iteration process of backpropagation algorithm. One iteration cycle of the backpropagation algorithm is known as an epoch.

Artificial Neural Networks is one of the components of Artificial Intelligence which in general are looked upon as small, but powerful computing devices. The power of any network depends on the learning process of the network. That means that the learning process is one of the most important parts in the processing of neural network. While the network is being trained, simultaneously it learns the complex non-linear relationship between the input and the output data sets. Neural networks are widely used for function approximation. Once the network undergoes the learning process the desired output data produced by the network is consistent with the target data sets.

Training process is a complete cycle in which initially the network is supplied with the input unit. This input unit propagates through the network in forward direction and ends with an output unit. The output unit obtained, when compared with the target value, gives the errors among the units. These errors are entirely responsible for adjustment of weights in a network in accordance with the specified learning rule for error minimization. This process is known as Backpropagation. Once the weights and biases are updated through backpropagation, the network is ready for the next cycle. Each cycle of backpropagation is termed as Epoch. For several epochs the weights and biases are adjusted to get the errors between the output and target units, minimized.

It has been already stated that backpropagation algorithm goes for such a combination of weights which tends to move towards the minimum error. This combination of weights obtained is considered as the solution of the trained network. This entire process employs the calculation of an error function for computing the gradient. For this, the error function must satisfy two conditions. Firstly, the error function must be continuous and secondly, it must be differentiable. A sigmoid function is most commonly used activation function with backpropagation algorithm.

Sigmoidal function has been represented in Fig 2.3, where, parameter c is a constant whose value is kept as one when working with backpropagation algorithm. As the value of c is increased from one to four, the sigmoid curve represents more of step function. Step functions were used in perceptrons and because of their disability as a differentiable and continuous function, they cannot be used as a learning algorithm.

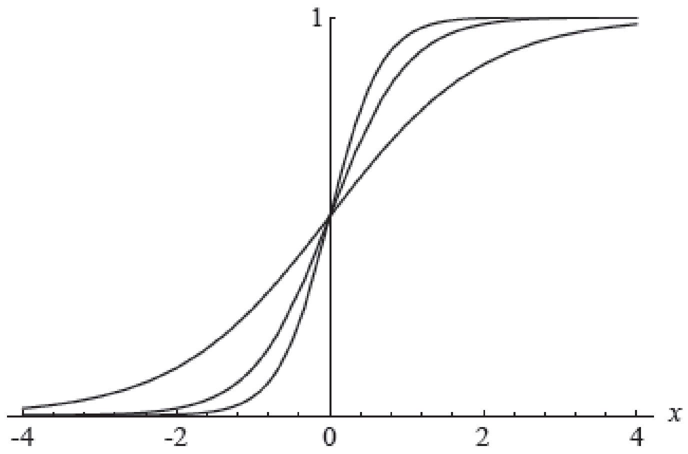


Fig.2.3: Sigmoidal function

$$S_c(a) = \frac{1}{1+e^{-ca}} \qquad \text{------------ } 2.1$$

The sigmoid function can be differentiated as follows:

$$\frac{d}{dx}s(a) = \frac{e^{-x}}{(1+e^{-x})^2} = s(a)(1-s(a)) \qquad \text{------------ } 2.2$$

A typical feed forward backpropagation neural network consists of an input layer, a hidden layer and an output layer. Such a network can have either single or multiple hidden layers. The number of neurons in input layer corresponds to the number of inputs in the given data set whereas the neurons in the hidden layers can be varied as desired. Output layer have neurons equal to the number of output nodes. Each neuron in a layer is connected with all other neurons of the next layer. The interconnecting bond of these neurons is referred to as ‘weights’. The updation of the weights in the network by backpropagating the errors is shown in Fig 2.4

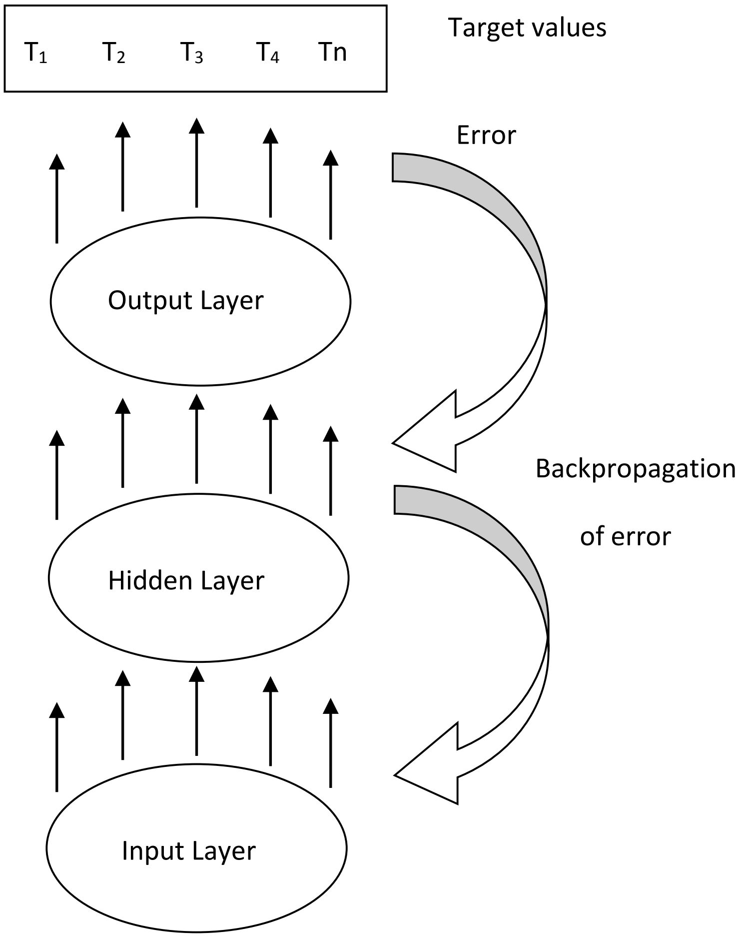


Fig 2.4: Feed Forward Backpropagation Neural Network

The non-linear relationship between the input units and the output units in backpropagation neural network are approximated by adjusting the weights associated with each unit within the network. This is so because backpropagation neural networks are based on the Least Mean Square (LMS) algorithm which was introduced by Widrow and Hoff in 1959. It is an adaptive algorithm and it uses the gradient decent method for error minimization. Also, backpropagation neural networks can be used for generalizing such data, which is not even included in the training sets. Further, this type of network gives quiet satisfactory results even with noisy data. Among all other type of neural networks, the feed forward back propagation neural network is the one which is the most frequent and widely used. It is one of the most popular networks of interest for the researchers. The Fig 2.5 indicates diagrammatically, how a typical feed forward back propagation neural network processes.

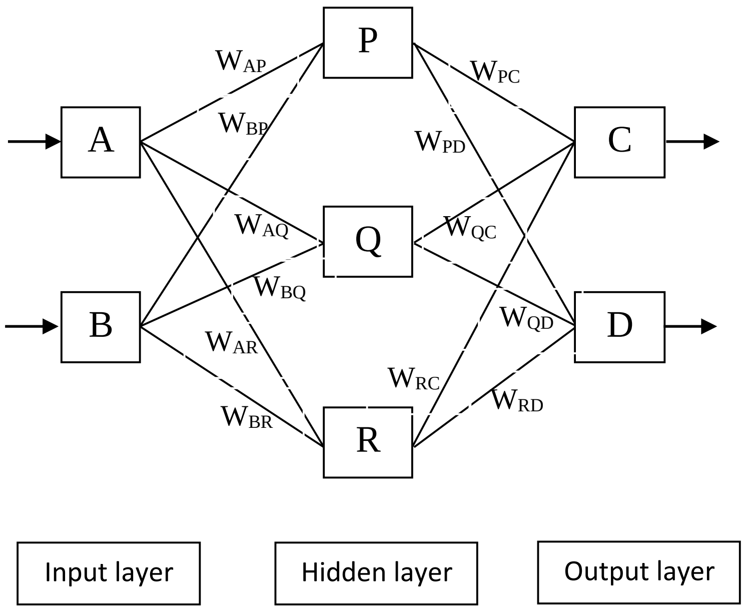


Fig 2.5: Artificial Neural Network Architecture

The basic steps involved in backpropagation algorithm have been explained below by taking an example of a neural network model. The neural architecture comprises of two input nodes, one hidden layers with three nodes and an output layer with two nodes. By creating an initial preliminary network with pre-identified input and output nodes and by providing the desired number of hidden layers and hidden layer neurons to the network, error of the output layer can be calculate as shown in eq-2.3 and eq-2.4.

$$\delta_C = out_C\,(1 - out_C)\,(target_C - out_C) \qquad \text{------------} \quad 2.3$$

$$\delta_D = out_D\,(1 - out_D)\,(target_D - out_D) \qquad \text{------------} \quad 2.4$$

The weights at each node of the output layer are updated as shown in eq-2.5 to eq-2.10.

$$W^{+}_{PC} = W_{PC} + \eta\delta_C out_P \qquad \text{------------} \quad 2.5$$

$$W^{+}_{QC} = W_{QC} + \eta\delta_C out_Q \qquad \text{------------} \quad 2.6$$

$$W^{+}_{RC} = W_{RC} + \eta\delta_C out_R \qquad \text{------------} \quad 2.7$$

$$W^{+}_{PD} = W_{PD} + \eta\delta_D out_P \qquad \text{------------} \quad 2.8$$

$$W^{+}_{QD} = W_{QD} + \eta\delta_D out_Q \qquad \text{------------} \quad 2.9$$

$$W^{+}_{RD} = W_{RD} + \eta\delta_D out_R \qquad \text{------------} \quad 2.10$$

Now, the error for the hidden layer neurons will be calculated. But here we don't know the target values for the hidden neurons. That is, we can't apply the same theory for calculation the error, as been used in case of the output layer. So in this case we will backpropagate the error from the output nodes as shown in eq-2.11 to eq-2.13.

$$\delta_P = out_P\,(1 - out_P)\,(\delta_C W_{PC} + \delta_D W_{PD}) \quad \text{----------} \quad 2.11$$

$$\delta_Q = out_Q\,(1 - out_Q)\,(\delta_C W_{QC} + \delta_D W_{QD}) \quad \text{----------} \quad 2.12$$

$$\delta_R = out_R\,(1 - out_R)\,(\delta_C W_{RC} + \delta_D W_{RD}) \quad \text{----------} \quad 2.13$$

Once the error of the hidden nodes has been identified, the weights of the hidden layer are updated as shown in eq-2.14 to eq-2.19.

$$W^{+}_{AP} = W_{AP} + \eta \delta_P in_A \quad \text{---------} \quad 2.14$$

$$W^{+}_{AQ} = W_{AQ} + \eta \delta_Q in_A \quad \text{----------} \quad 2.15$$

$$W^{+}_{AR} = W_{AR} + \eta \delta_R in_A \quad \text{----------} \quad 2.16$$

$$W^{+}_{BP} = W_{BP} + \eta \delta_P in_B \quad \text{---------} \quad 2.17$$

$$W^{+}_{BQ} = W_{BQ} + \eta \delta_Q in_B \quad \text{---------} \quad 2.18$$

$$W^{+}_{BR} = W_{BR} + \eta \delta_R in_B \quad \text{---------} \quad 2.19$$

This cycle is repeated number of times so as to achieve the goal set to evaluate the performance of the network. The number of nodes in hidden layer is varied according to how well the network had developed the relationship between the inputs and the outputs.

2.5 Advantages of Backpropagation

1. It does not require any pre programming
2. Backpropagation algorithm has ability to give precise results even if some of the data is missing out of the database.
3. Can be applied to any sort of simple as well as complex problems.
4. This algorithm has great tolerance towards the noisy data.
5. Any modification in the structure of the neural network does not affect the working of backpropagation algorithm.
6. As this algorithm is fast, it saves time.

7. Backpropagation have been applied in various fields and it has been seen that the results obtained are very close to the actual results maximum number of times.
8. The adaptation and generalization capabilities of the neural networks had greatly increased.
9. With backpropagation algorithm, neural networks can be applied to any real world problem very conveniently and the results obtained can be trusted upon under a particular error range.

2.6 Limitations of Backpropagation

1. A goal is to be set for error function at which the network is said to be well trained. To achieve this goal network is to be trained again and again and at different epochs. This process consumes a lot of time.
2. In absence of sufficient data i.e. when the data provided to the network is somewhat in less quantity, the generalization of network becomes weak and the results obtained may vary largely from the actual data.
3. There is always a chance of over fitting of the curve.
4. Backpropagation algorithm does not tell about the black box calculations, i.e. the relationship between the input and the outputs remain unknown.
5. Backpropagation algorithm cannot function if the error function is discontinuous or is not differentiable.

2.7 Summary

From the above discussion it is well clear that as the biological neural networks train themselves by experience, in the same way Artificial Neural Networks learn or train themselves by going through a large number of data provided by the user. We can say that these neural nets are analogous to biological neural nets in human brain. Neural networks are capable of approximating any given function which has been used in various research works. With different type of training and transfer functions, the network can be adjusted for the particular relation between inputs and outputs in the most suitable manner.

One of the most common and widely used networks is the Backpropagation. It is the feed forward neural network which uses Backpropagation algorithm. Whenever a nonlinear relationship has to be interpreted between input and the output values, such type of network architecture is used. This is achieved by minimizing the error between the outputs and the target values by adjusting the weights automatically. With the advancement in the field of computer science and technology, now even the complex neural networks can be made with ease which would be capable of answering almost every real world problem.

CHAPTER-3

Literature Review

3.1 Introduction

Artificial Neural Networks (ANNs) have captured the interest of more and more researchers in the last two to three decades. In this time duration ANNs have found their use in all sorts of real world problems. Also the field of ANNs is not limited and they are such a tool which can be applied to any sort of problem in any field. Surely, this is due to the biggest and the most important characteristic of neural networks, i.e., their ability to learn from the experience which is fed into the network (in form of a database provided to the network for training). Due to this characteristic, neural networks are able to perform very well in all sorts of complex situations even without knowing the relationship between the inputs and the outputs of the problem.

3.2 Literature review

In one of the research work free lime content in cement clinkers was estimated by **Jingling Y. *et al*. 2012**. For this work along with several mathematical models backpropagation neural network and Radial Basis Function (RBF) neural network were also applied to the problem. The entire data was normalised in the range of 0 to1. The network architecture consisted of six input nodes as; raw meal feeding quantity (t/hr), external fuel amount of raw meal (t/hr), rotation per minute of kiln (RPM), lime saturation factor (LSF), silicon ratio, and aluminium rate. Output considered was the lime content. This neural model consisted of only one hidden layer. Total of 100 data units were considered out of which 75 data units were used for training the network whereas the remaining 25 data units were used for testing the network. Out of all the models, RBF neural network model gave the best results. The results of backpropagation neural network were a litter higher than that of RBF neural networks but were far better than the results of other models.

Artificial Neural Networks were applied in conceptual design of communication towers by **Agrawal V. *et al.* 2011**. Various combinations of communication towers were made and analysed for optimised weight. These towers were separated into three categories as low, medium and high weight towers. The artificial neural network model was created to distinguish between the towers in these three categories. The neural model architecture consisted of 3 nodes in the input layer as height of the tower, base width and the panel ratio. The outputs considered were the weight and deflection of the towers. Single hidden layer was adopted with 8 hidden layer neurons in this network. Authors concluded that the developed network was able to clearly differentiate between the light, medium and heavy weight towers.

In one of the studies, an artificial neural network was developed for predicting the shear strength of reinforced circular concrete columns by **Caglar N. 2009**. The shear strength was calculated using the guidelines provided in ACI 318 – 2005. From the entire database, 31 data units were used for training the network whereas 16 data units were used for testing the network model. The data units used for testing the network were not included in the training set. The entire data was normalized in the range of 0 to +1. The goal chosen was 0.00001. Scaled conjugate gradient algorithm was used for training the network with sigmoidal activation function. Out of the various trials, the network architecture with 13 nodes in the input layer, 6 and 3 nodes respectively in the first and second hidden layers and single node in the output layer was chosen. The network was trained upto 20,000 epochs. The training MSE came out to be 0.00 with a correlation coefficient of 1.00. On the other hand testing MSE came out to be 0.003 with a correlation coefficient of 0.8330. It was concluded that the developed neural model can be effectively used for the prediction of shear strength of circular reinforced concrete columns.

Another example of application of neural network is the determination of chloride diffusion inside the reinforced concrete member by **Song H.W. *et al*. 2009**. Neural networks have been used here to save both, time and cost. Backpropagation learning algorithm has been used for training the network along with the tan-sigmoid transfer function. For determining the chloride diffusion coefficient, rapid chloride penetration test was conducted. Total of 120 data sets were provided to the neural network model for training. The neural model consisted of 8 input nodes; W/B ratio, unit weight of OPC, GGBS, flyash, silica fume, sand, coarse aggregate, and duration time in submerged condition. The output layer consisted of a single node as the diffusion coefficient of chloride ion. It was concluded that the neural network model was able to predict the values of chloride diffusion quiet precisely.

For estimating the bond between steel and concrete, an artificial neural network model was developed by **Dahou Z. *et al.* 2009**. This was done by determining the ultimate pull-out load on ribbed bars with 10mm and 12mm nominal diameter. Three different grades of concrete were used for the experimental purpose. Two ANN models, ANN-6 and ANN-2, were developed for determining the bond strength of concrete. Total of 112 data units were provided to the network, out of which eighty percent data was used for training the network and twenty percent for validating the selected network. Cross validation technique was employed for network validation. Neural network model ANN-6 consisted of six input nodes; diameter of the ribbed bar, the water to cement ratio, the gravel to sand ratio, the crushed to rolled gravel ratio, the type of cement and the concrete maturity. This network comprised of a single hidden layer with ten nodes and the only output considered was the ultimate pull out load. The value of correlation coefficient obtained was 0.91 for training and 0.89 for testing set. Neural network model ANN-2 consisted of two input nodes; the compressive strength of concrete at the time of the pull out test and the diameter of the ribbed steel bars. This network also consisted of a single hidden layer with four nodes and the only output considered was the ultimate pull out load. The value of correlation coefficient obtained was 0.97 for training and 0.94 for testing set. It was concluded that the neural network models developed were quiet precise in estimating the bond strength of steel and concrete.

For predicting the compressive strength of silica fume concrete, a neural network was developed by **Ozcan F. *et al*. 2009**. The developed neural network consisted of an input layer with six input nodes; cement, amount of silica fume replacement, water content, amount of aggregate, plasticizer content and age of samples. Single hidden layer with eleven nodes was used in the network architecture. The output layer comprised of single node as the compressive strength. Total of 240 data sets were created out of which 135 data sets were used for training the network, 50 data sets were used for validation of the neural network model whereas 55 data sets were utilized for testing the network model. The correlation coefficient for training came out to be 0.9944 whereas it was 0.9767 and 0.9724 for testing and validation respectively. It was concluded that the artificial neural networks can be a used effectively for such type of predictions for determining the compressive strength of concrete.

A backpropagation neural network was developed for identification of damage intensities of joints of a truss bridge structure by **Mehrjoo M. *et al.*, 2008**. Two type of truss bridge structures were used in the study. The first one was the simple warren truss system. The number of training patterns for the simple warren truss was 273. Total of 30 data units were kept for testing the network. The network consisted of an input layer with 32 nodes, a hidden layer with 50 nodes and an output layer with 5 nodes which constituted the network architecture. The root mean square error for training came out to be 0.66% whereas it was 1.65% for testing. The other one was the Louisville bridge truss. Number of training data sets for this network was 729 whereas 30 data units were used for testing the network. The network consisted of an input layer with 65 nodes, a hidden layer with 57 nodes and an output layer with 6 nodes. The root mean square error for training came out to be 0.79% whereas it was 1.77% for testing. In both the cases the network model was trained upto 75000 epochs. A very low error between the training and testing sets indicated the reliability of neural networks in damage detection of truss bridges.

An experiment was conducted for the determination of flow resistance in smooth open channels by **Bilgil A. and Altun H. 2008**. The experiment involved the prediction of friction coefficient of Manning's formula for open channel flow. A neural approach was used by the authors for the prediction of friction coefficient. The input parameters for training the network were obtained using the experimental results. The network model comprised of 6 input nodes; Reynolds Number, relative roughness, cross-sectional geometric shape, non-uniformity of the channel in profile and plan, Froude number and degree of flow unsteadiness. Output layer consisted of a single layer with one node as the friction coefficient. Out of the entire database, 50% data was used for training the network and remaining 50% data is used for network evaluation. A correlation coefficient of 0.9926 was obtained between the results obtained by artificial neural network model and the experimental results. It was concluded that the efficiency of neural approach was better than the Manning's approach.

In one of the experimental programs, neural networks were applied to estimate the shear strength of concrete beams devoid of shear reinforcement by **Jung S. and Kim K.S. 2008**. Various neural network models were created by varying the number of nodes in the hidden layers. The final selected network model consisted of 6 nodes in the input layer and 1 node in the output layer. The inputs considered were, the web width, effective depth, shear span to depth ratio, concrete compressive strength, tensile strength of reinforcing steel and longitudinal reinforcement ratio. The only output was the shear strength. Network model also comprised of 2 hidden layers with 11 and 7 nodes in first and second layer respectively. The data was normalized between 0 and 1. Tan-sigmoid was used as the transfer function for the network. Total of 350 data sets were provided to the network model for training whereas remaining 48 data units were kept for testing the network. The network was trained for 100,000 epochs. The network model was reported to be beneficial in giving non-conservative prediction.

For the analysis of non-linear structures under dynamic loading, **Joghataie A, *et al*. 2008**, developed a neural network with a different activation function. This activation function was based on the use of Prandtl–Ishlinskii operator. The training of network is done using genetic algorithm. The input layer comprised of displacement and velocity at the start of time and also the acceleration of ground during the time step. The outputs were the displacement and velocity at the end of time step. The results obtained by applying Prandtl neural network were quite near the results obtained by numerical integration. It was concluded that the new activation function developed works well when applied with neural networks.

In one of the studies, building-related symptoms prevailing in some of the individuals were studied by **Sofuoglu S.C., 2008.** These individuals were the occupants of an office building. Artificial neural networks, which have been employed in various other environmental studies successfully, were also used here for the prediction for such building-related symptoms. The total data consisted of hundred units out of which sixty data units were used for training, twenty for validation and the remaining twenty for testing the performance of adopted network. The input layer consisted of ten nodes with six pollutants; CO_2, $PM_{2.5}$, HCHO, VOC's, bacteria, fungi and four comfort variables; temperature, RH, light and noise. The output layer consisted of a single node as POPS2. This neural model with single hidden layer consisting of ten neurons gave the best performance.

In one of the investigations for designing reinforced concrete beam, **Rao H.S. *et al*. 2007**, developed a hybrid neural model having properties of both, feedforward neural network as well as of genetic algorithm. A total of 120 data sets were generated out of which 100 data units were used for training the neural network model whereas 20 data units were used for validation of network model. The network was modelled with 5 input nodes; Moment, Shear, Grade of concrete, Grade of steel and Width of beam. Output layer with 3 nodes as depth of beam, area of the reinforcement and spacing of stirrups, was considered. The data was normalized in the range of 0 to +1. Genetic Algorithms were used for determining the weights for the network units. The neural network model was trained upto 1000 epochs. It was concluded that the developed neural model was capable of providing safe design of reinforced concrete beams.

For determining the deformation capacity of rectangular reinforced concrete columns, an artificial neural network model was developed by **Inel M. 2007**. The data bank for this study was taken from one of the literatures of similar kind of work. Out of this data bank only that database was chosen which passed the criterion set up by the author. This constituted of total 237 data units which were used for training the network model. A feedforward backpropagation neural network was used in this study. The network architecture comprised of 9 input nodes; Aspect ratio, Longitudinal reinforcement ratio, Yield strength of longitudinal reinforcement, Uniaxial (cylindrical) concrete compressive strength, Yield strength of transverse reinforcement, Transverse steel spacing, Ratio of transverse steel parallel to the direction of loading, Axial load ratio and Confinement effectiveness factor. The only output for the network was taken as ultimate displacement drift capacity. For training the network, 197 data units were used and the remaining 40 data units were used for testing the data. Only one hidden layer was provided to the network in which the number of neurons was varied from 9 to 20. The network was trained upto 2000 epochs. The training MSE when the number of neurons in hidden layer were 18, came out to be 0.0001083 and the testing MSE with the same configuration of network model was 0.0000561. It was concluded that the neural models performed better than analytical models.

In an experiment, cement was replaced by flyash and silica fumes and the corresponding variations in the strength of concrete were studied by **Pala M. *et al*. 2007**. Artificial feed forward neural networks with backpropagation algorithm, has been employed for the evaluation of concrete strength. The neural model consisted of eight input nodes; fly ash replacement ratio, silica fume replacement ratio, total cementitious material, fine aggregate, coarse aggregate, water content, high rate water reducing agent and age of samples. The output layer consisted of a single node taken as the compressive strength of concrete. The neural architecture consisted of a hidden layer with nine nodes. Total data consisted of 144 units out of which 130 units were used for training the network while the remaining 13 data units were used for testing the network. The results showed that the proposed ANN model was successful in generalising the given problem with reasonably good predictions. It was concluded that this ANN model can prove to be good in estimating the target engineering properties of concrete.

Artificial neural networks were coupled with genetic algorithms to give an economic configuration of bridge decks by **Srinivas *et.al*. 2007**. Grillage analysis was performed for the designing the different configurations of deck slab. The live load considered was IRC 70R. Total of twelve inputs; span length, carriage way width, total depth, number of longitudinal girders, number of cross girders, spacing of longitudinal girders, spacing of cross girders, thickness of deck slab at mid, thickness of cantilever end of slab, thickness of web, width of bottom flange of main girder and thickness of bottom flange of main girder, were taken. Number of outputs taken were four; maximum bending moment due to dead load, maximum bending moment due to live load, shear due to dead load, shear due to live load. Single hidden layer was considered with ten numbers of neurons. For training of the selected network model, 86 data sets were used and 10 sets of data were used for validation. In this study it was concluded that, firstly, using artificial neural network with genetic algorithm reduces the computational time considerably as when alone genetic algorithm were used. Secondly, with this approach, computational effort will be reduced when design parameters are more and a satisfactory result will be obtained.

In one of the research works, artificial neural networks were used to estimate the scour depth below the spillways by **Md. Azamathulla H. *et al*. 2007**. A standard feed forward back propagation network was used along with cascade correlation scheme, some configurations of radial basis function and adaptive neuro-fuzzy inference system. Two network inputs were taken as; characteristic head and the discharge intensity over the spill ways. The only output was taken as the scour depth. For training of network eighty percent data was used and remaining twenty percent data was used for validation of the network. After several trials, the hidden nodes were taken as two in number. It was seen in this study that neural networks proved an edge over several other equation based methods for prediction of scour depth such as the formulae given by Veronese, Wu, Martins and Incyth and also formula derived by various other authors.

In one another experiment, damage in prestressed concrete beams were detected using artificial neural networks by **Jeyasehar C. A. *et al*. 2006.** A feed forward neural network model was created and backpropagation algorithm was used for training the network model. The typical network architecture consisted of five input nodes; applied load, natural frequency, deflection, crack width and ultimate load. Various other networks architectures, having inputs from two to five, were constructed for this study. With two hidden layers having five and seven nodes, the network comprised of an output layer with single node as extent of damage. Total of 900 data sets were used for training the network whereas 51 data sets were used to test the network model. The network model with five input nodes gave the best results. It was shown that the artificial neural networks trained only with dynamic data can detect the damage within ten percent error range.

A finite element model was created for a hundred year old suspension bridge and the damage in terms of vibration signature of the bridge was investigated using patter recognition by **Yeung W.T. *et al*., 2005**. Two different types of neural networks; PRAN and DIGNET, were used. The network model had undergone unsupervised learning. Total of fifteen damage data sets were used for training out of which six data sets were used for testing. It was concluded that the reliable damage detection rate of upto seventy percent can be achieved by these neural models.

In one of the research work, concrete breakout strength was estimated for single anchors in tension using feed forward backpropagation neural networks by **Ashour A.F. *et al*. 2005**. The developed network consisted of four input nodes; embedment depth, anchor head diameter, concrete strength and anchor installation system whereas the output layer consisted of a single node as the tensile capacity of anchors. This tensile capacity is governed by the breakout strength of concrete. Four different neural networks were constructed with varying number of nodes in input and hidden layers.

Levenberg-Marquardt algorithm was used for training the neural network. Out of the total data, fifty percent data has been used for training of the neural network model, twenty five percent for validating the network and remaining twenty five percent for testing the network model. The chosen network gave a correlation coefficient of 0.907 with seven nodes in the hidden layer. The results obtained by the designed neural networks were in conformity with the formula given in ACI: 318-02.

In an experimental work, flutter derivatives for a bridge deck were estimated using artificial neural networks by **Jung S. *et al*. 2004**. The neural architecture consisted of 100 nodes in input layer, 20 nodes in each of the two hidden layers and 6 nodes in output layer. Total of 1694 data units were provided to the network for training and 363 data units were kept for testing which were not shown to the network. The network model was trained for 10,000 epochs. The average error for training came out to be 0.001803 and for testing, average error was 0.036673. Two other variations were made in this network. One variation was made by varying the input nodes while the other variation was done by data compression technique. In both the variations, the testing results were more than the standard network model. It was concluded that the standard neural network model very well estimated the flutter derivatives.

In one of the research work, feedforward backpropagation neural networks were modelled for determining the depth and diameter of steel reinforcing bars by **Zaid M. *et al*. 2004**. An inductive sensor was used for generating a number of images as a data set for neural networks. In order to acquire data, Polynomial-Based Layer Separation (PBLS) algorithm was used. Data filtering was done using two different design software packages, Fourier Processor and Signal Wizard, to avoid any local maxima or minima within the data. Two neural network models were created in this study. First one was for determining the depth of reinforcing bar. This network architecture comprised of 2 nodes in the input layer (peak and full width at half height), 2 hidden layers with number of nodes as 3 in the first hidden layer and 7 in second hidden layer. The only output considered was the depth of reinforcing bar. Tan- Sigmoid transfer function was used for this network model. Out of the total data of 41 bar depths at 6 different bar sizes, a data set consisting of 31 bar depths at 5 different bar sizes, were used for network training. The remaining data set with 10 bar depths at single bar size was kept for testing the network. Second neural model was created for determining the diameter of reinforcing steel bars. Architecture of this network was quite similar to the first neural model, difference being only that, it comprised of 4 nodes in first hidden layer. Data provided to the network for training consisted of 5 different bar size and the data for remaining one bar size was kept for testing. Results from both the networks for depth and diameter of steel reinforcing bars gave very good correlation when compared to actual results.

In one another research work, artificial neural networks were employed for determining the compressive strength of concrete by **Kim *et al*. 2004**. Total of ninty eight data sets were fed into the network for training. The data for carrying out the experiment was provided by two different ready mixed concrete companies. The input layer consisted of eight nodes; water-cement ratio, fine aggregate percentage, unit water content, unit cement content, unit fine aggregate content, unit coarse aggregate content, admixtures and slump whereas the output layer was with single node as specified compressive strength. For testing, ten data sets were used. The difference between the results, estimated and the tested, were 3.9% for specified strengths and 3.2% in case of average strengths. This study provided a base for proving the effectiveness of the application the neural networks in determining the compressive strength of concrete.

Artificial neural networks were applied with backpropagation algorithm, for updating the baseline finite element model of a highway bridge by **Feng M.Q. *et al*. 2004**. A comprehensive vibration data was collected by installing a typical sensor system on these bridges. From this data, model parameters such as natural frequencies and mode shapes of the bridges were estimated. These were used as the inputs for the neural network model. The output considered were the structural parameters such as mass and the stiffness elements. The network was trained using five thousand data sets upto one thousand epochs. The developed neural model gave high level of accuracy in determining the structural parameters.

For determining such a position of live load on the bridge pier which could produce worst effect on the pier, a neural network was developed by **Williams M.E. *et al*., 2004**. The position of live load has been determined using the guidelines of the AASHTO-LRFD, 1994 bridge design code. The piers in this study have been modelled in FB-PIER program. Both single column pier and multiple column piers have been investigated in the study. A two span bridge model was generated in LIVEGEN program. The neural network modelled for single column pier consisted of input layer with four input nodes; the number of design lanes, the span length, the girder spacing divided by the pier cap cantilever length, and the pile spacing divided by the width of the pile cap. Single hidden layer was used with six nodes. The output layer consisted of eight nodes representing the different truck and lane load position for each lane. Data from 28 different bridge structures were used for training the network. The network model was validated using three piers with worst load positions. The neural network designed for multiple column piers consisted of input layer with six input nodes; the number of design lanes, the span length, width of the clear roadway divided by the girder spacing, the pier column spacing divided by the girder spacing, the pier cap cantilever length divided by the pier column spacing and the pile spacing divided by the pile cap width. Single hidden layer was used with ten nodes. The output layer consisted of eight nodes representing the different truck and lane load position for each lane. Data from 47 different bridge structures were used for training the network. The network model was validated using piers with worst load positions. For a single column pier the validation error was upto 4% whereas for multiple column piers the validation error was upto 20%. This work proved the reliability of neural networks in predictions of factors which are essential in the design of bridge piers.

In one of the research works, artificial neural networks with backpropagation algorithm were used for structural damage identification by **Kao C.Y. *et al*. 2003**. The process involved firstly, the system identification and secondly, the detection of structural damage. The network consisted of 301 input nodes and 5 output nodes. No hidden layer was used in this network architecture. The training of network was taken upto 3000 epochs. The results from the ANN showed good proximity with the results of laboratory investigation. It was concluded that ANN approach of damage detection is quite feasible.

The reinforced concrete columns were studied for their confinement efficiency under the concentric loading by **Tang C.W. *et al*. 2003**. These columns consisted of rectilinear transverse steel. An artificial neural network approach was established for this purpose. Data was collected from one of the literatures and consisted of a set of 55 columns with square cross section. Out of this 45 data units were used for training the network whereas the remaining 10 data units were used for testing the network model. The network architecture comprised of 6 input nodes, 1 hidden layer with 14 nodes and an output layer with 2 nodes. Input parameters considered were, the cylinder compressive strength of concrete, area of concrete in the core, volumetric ratio of transverse steel in concrete core, the distance between the laterally supported longitudinal bars, spacing of transverse steel, and yield strength of transverse steel. Outputs considered were the maximum axial stress and strain of confined concrete. Inputs of the network model were normalized in the range of -1 to +1 whereas the outputs were normalized in the range of 0.2 to 0.8. The correlation coefficient for maximum strain of confined concrete came out to be 0.9983 in case of training and it was 0.9217 for testing sets. Also, correlation coefficient for the maximum strength of confined concrete came out to be 0.9988 in case of training and it was 0.9911 for testing sets. These results were much better than the results of other researchers, as shown by the author in his paper. It was declared that the predictions from the neural model were better than other parametric models.

Backpropagation neural networks were used for the evaluation of pavement performance by **Attoh-Okine N. O. *et al*. 2002**. A rough set was used for data mining along with the neural networks. The network was trained using seventy five samples, validated using fifteen samples and tested using twenty samples. Two different cases were considered. In the first case the data is directly fed into the network. The network architecture in this case comprised of seven nodes in the input layer, five nodes in single hidden layer and having a single output. The root mean square error (RMS) in this case was four percent. In the second case, the data reduced by rough set theory was given to the network. This network consisted of five input nodes, five nodes in hidden layer and was with a single output node. The root mean square error (RMS) in this case was one percent. It was concluded that this approach will be beneficial with large data sets.

In one other research work, backpropagation neural networks were applied to determine and classify the flaws in concrete structure by **Xiang Y. *et al*. 2002**. Feature vector to identify the flaw in concrete structure, were selected using the bispectrum. The network architecture consisted of an input layer with 36 nodes, hidden layer with 20 nodes and the output layer with 3 nodes. Three different types of data sets were prepared. In the first case, the data without any noise was used for training and testing the network. In the second case, the training and testing sets were provided with some noisy data. In the third case only the testing data consisted of some noise whereas the training data in this case was kept clean. The total number of data units for training was 156 and for testing the network model 195 data units were used. It was concluded that neural network provided a convenient methodology or technique for classification problem.

In one of the studies, artificial neural networks were used with dynamic backpropagation for the analysis of deformed behaviour of culvert structure by **Kerh *et al*. 2000**. The structure analysed were under static loading. Weight matrix and bias vector in neural networks were replaced by stiffness matrix and force vector respectively. The results obtained by neural networks were compared with the analytical solutions as well with the finite element solutions. The results in terms of displacement plot showed more deformations at the centreline in the downward direction. Artificial neural networks generated the same results as been obtained by the analytical method.

A study was conducted to determine the damping at high amplitude in tall buildings by **Li Q.S., *et al*. 2000**. It is quite difficult and tedious when it comes to estimation of the damping values at field. To overcome this problem, General Regression neural network with Genetic Algorithm was used in this study. For the first direction, the network architecture comprised of 4 nodes in input layer, 128 nodes in hidden layer and one node in output layer. The total numbers of data units were 128 out of which 120 data units were provided to the network for training and the remaining 8 data units were kept for testing the data. For the other direction, the network architecture comprised of 4 nodes in input layer, 105 nodes in hidden layer and one node in output layer. The total numbers of data units were 105 out of which 97 data units were provided to the network for training and the remaining 8 data units were kept for testing the data. The maximum absolute error for direction one and two came out to be 0.046 and 0.028 respectively. It was concluded that ANN were very promising in estimation of amplitude dependant damping in buildings.

In one of the experiments, mechanical behaviour of concrete was studied under the high temperature by **Mukherjee *et al*. 1997**. A feedforward neural network with backpropagation algorithm was used to determine the stress-strain relationship of concrete. Under three distinct conditions of load and temperature, the behaviour of concrete was studied. In the first condition of varying load under isothermal environment, the network consisted of an input layer with five input nodes; current strain, temperature, elastic modulus, compressive strength and ultimate strain. Two hidden layers with five nodes in each layer were taken along with the single output node as current stress. The predictions by neural network model in this case were in close proximity with the experimental results. In the second condition of varying temperature under constant load, the network architecture consisted of an input layer with six nodes; current temperature, load level, elastic modulus, compressive strength, ultimate strain and the coefficient of thermal expansion. The output layer consisted of a single node as strain at the corresponding load and temperature. This network consisted of two hidden layers with eighteen nodes in each layer. In this case also, the results from the network were quiet appreciable. In the third condition of varying temperature with overall restrained condition, the network with six nodes; temperature, elastic modulus, compressive strength, ultimate strain, the coefficient of thermal expansion and the rate of heating, in input layer were considered. The network consisted of two hidden layers with twelve nodes in each layer. The output layer consisted of a single node corresponding to restrained strain. The network results agreed to the experimental results quiet precisely. This experiment proved neural networks to be superior to several other mathematical models.

An artificial neural model was developed for estimating the concrete strength by **Lai S. *et al*. 1997**. The developed network comprised of eight input nodes; class of cement, fine sand, coarse sand, fine aggregate, coarse aggregate, cement content, water-cement ratio and plasticizer. The output layer consisted of a single node as compressive strength. The hidden layer neurons were initially taken as twenty and were changed as per the various trials. It was concluded that when the number of hidden layer neurons was in the range four to eight, the performance of the network remained almost the same. The solutions given by neural networks were convincing.

In one of the research works, a multilayer feed forward neural network was designed for the determination of asphalt-concrete pavement cracks by **Mohamed S.K. *et al*. 1994**. For evaluation of the problem, four classifiers were considered; transverse cracking, longitudinal cracking, diagonal cracking and combined cracking which included two or more cracks having different directions. Total of 230 data sets were used for training of neural network whereas 20 datasets were used for testing the neural model. The network consisted of five node input and output layer with single hidden layer having five nodes. The results were quiet encouraging as the neural network classifiers did better than the other classifiers.

An artificial neural network model, which could be helpful in calculating the construction productivity**,** was proposed by **Chao L.C. *et al*. 1994**. Two different networks were created for this purpose. The first network architecture consisted of 5 input layer nodes, 15 nodes in hidden layer and an output layer with 2 nodes. The network was provided with 200 data sets for training and was tested with 1000 data sets. Neural network model was trained upto 5000 epochs. The neural model was able to generate results with an average error under 0.21% and the maximum error within 1.55%. The second network architecture consisted of 8 input layer nodes, 48 nodes in hidden layer and an output layer with 2 nodes. The network was provided with 1048 data sets for training and was tested with 3000 data sets. Neural network model was trained upto 7500 epochs. The neural model was able to generate results with an average error under 0.29% and the maximum error within 2.82%. It was concluded that the neural networks were able to map the complex relation between the environmental conditions influencing the job and the productivity efficiently.

3.3 Summary

Comprehensive literature review have been performed for proving the applicability of ANNs in the field of civil and structural engineering. It is seen that the researchers round the globe are trying to develop ANNs with varying architectures, probably suiting to their problem area. Several other researchers are having a keen eye on the performance of ANNs and as a result, several algorithms for training have been developed. Also, improvement in results by combining ANNs with Genetic Algorithms or Fuzzy Logics are attracting the attention of several researchers. In a broader picture, this indicates towards the necessity of a conceptual tool for the designers at the preliminary stage of design.

CHAPTER 4

Methodology: Problem Statement and Network Selection

4.1 Introduction

In the previous chapters, we have already seen that, till date, a number of researches have been done on neural networks and evaluation of its efficiency in the field of civil and structural engineering. Although, civil engineering is a very wide field, but neural networks have been applied in almost all the civil engineering problems, such as for determining various properties of concrete, in bridges, buildings, pavements, dams, reinforced cement concrete members and structures, steel structures, water tanks, prestressed concrete, for various test on steel and concrete, irrigation structures, rainfall runoff forecasting, etc. and in many more areas of civil engineering. Results have always been convincing and have indicated towards the requirement of more exploration in this field. All these factors strongly prompted for the execution of this thesis work.

The main objective of this research work is to prove the applicability of neural networks in estimating the deflection and post-tensioned steel (PT steel) requirement in post tensioned slabs (PT slab). The detailed methodology adopted for this research work is presented here.

4.2 Problem Statement

This research work comprises of designing a two way, three span continuous PT slabs with various configurations so as to generate a database for the application of ANNs. Deflection and the weight of PT steel for each configuration of PT slab have been determined using the structural software for design of post tensioned members, ADAPT-PT. The results obtained by this software have been verified by manual design of ten randomly selected PT slab configurations.

4.3 Design of PT-slab

The basic guidelines for the design of PT-slab are from the book "*Design of Prestressed Concrete*" by R.I.Gilbert and N.C.Mickleborough (2005). The main steps involved are as follows:

1. First step involves the determination of major inputs required for designing purpose such as the geometry of slab (i.e. its length, width and the initial thickness), grade of concrete, column size and loading on the PT-slab.

    a. In the present problem, square slabs have been considered having spans of 7, 8, 9, 10, 11 and 12 meters.

    b. Initial thicknesses of PT-slabs considered are as 170, 190, 210, 230 and 250 mm.

    c. Only live load has been considered on the PT-slab and has been taken as 3, 4 and 5 KN/m^2.

    d. Columns with square cross-section of 450mmx450mm, 600mmx600mm and 750mmx750mm with 3m height are adopted.

    e. Grade considered for concrete are M35, M40 and M45.

    f. Characteristic strength of PT steel is taken as 1860 MPa.

2. Now the drop size is taken as L/6 and B/6 in the respective directions and the thickness of drop is taken as T/4. Here L, B and T are the respective length, width and thickness of the PT-slab to be designed.

3. It has been assumed that in addition to the dead load of the PT-slab, 30% of live load (LL) will always be there on roof. Hence PT-slab has been designed to balance a total sustained load of DL+ 30% LL by the prestressing cables.

4. Assuming a suitable cover, cable profile is then established.

5. Now the losses of prestress is determined such as the loss of prestress due to friction, due to elastic deformations, anchorage loss and various time dependent (creep and shrinkage) losses.

6. After all the losses have been calculated, the effective prestress is determined.

7. Corresponding to the effective prestress, the total prestressing force at the jack is calculated.

8. Now the total jacking force divided by the characteristic strength of PT-steel gives the required area of PT-steel.

9. Calculate the positive and negative moments in the PT-slab and drop panel and check this for the permissible stresses.

10. Estimate the moment of inertia of the column strip and the middle strip.

11. Calculate the deflection (max deflection at the center of panel).

This research work comprises of designing a two way, three span continuous Post-Tensioned (PT) slabs as shown in Fig 4.1. A detailed design is performed with all the applied formulae and has been produced in the annexure III.

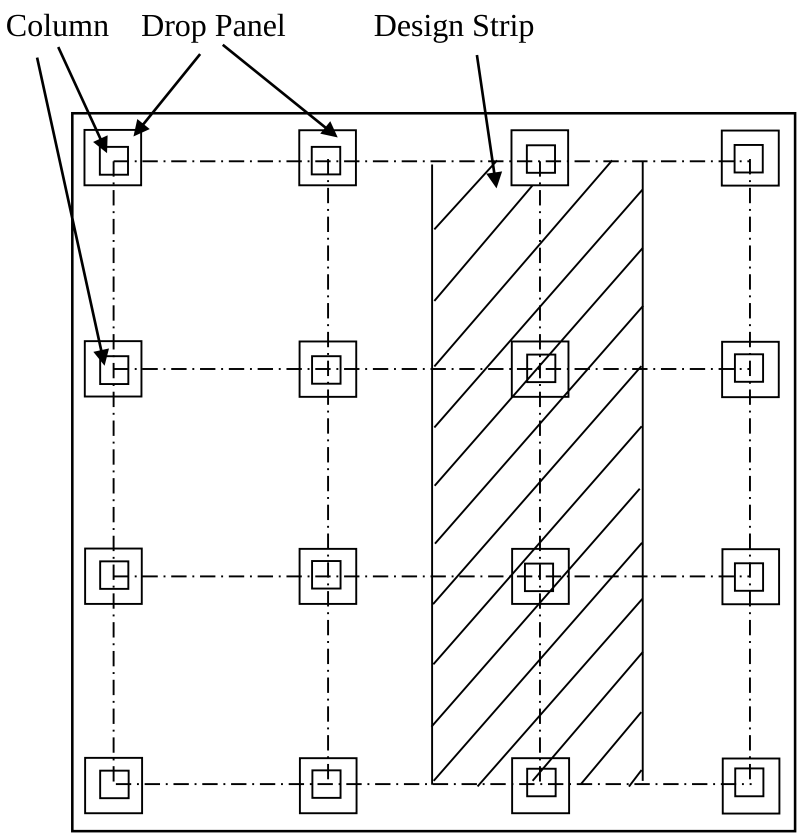


Fig 4.1: Geometry of PT-slab

4.4 Network Development

For the present case, feed forward neural networks with backpropagation algorithm are used. For developing the network, neural network toolbox from the standard software, MATLAB, is utilized. In such a network, the information is first propagated towards the output nodes passing through the hidden nodes with some initial random weights. Now the difference in the output given by these networks and the target values are evaluated and this difference in values is termed as an error. These errors are now propagated in backward direction and the associated weights of hidden layers are updated. Now again, with these updated weights the information is sent ahead. This iteration is made till the desired level of accuracy in results or the goal is achieved. If the network is trained with sufficient data, one can expect quiet convincing results. It is now quiet clear that ANNs perform on the experience, which is to be fed into the network. For the same reason, several networks have been developed in this study. Total of five input nodes are provided to the network which are the Span of PT-slab, depth of PT-slab, live load, column size and grade of concrete. The variation in these input parameters is shown in Table 4.1:

Table 4.1: Various inputs considered for the network

Inputs	**Variations**	**Units**
Span of PT-slab	7, 8, 9, 10, 11 and 12	m
Depth of PT-slab	170, 190, 210, 230 and 250	mm
Live load	3, 4 and 5	kN/m^2
Column size	450x450, 600x600 and 750x750	mm^2
Grade of concrete	M35, M40 and M45	

The output layer consisted of two nodes as the deflection of PT-slab and the weight of PT-steel. Network architecture consisted of both single layered and double layered networks. The number of hidden layer neurons are taken as 5, 10, 15 and 20 for single layered networks and (5, 5), (7, 7) and (9, 9) for double layered networks. Levenberg-Marquardt training algorithm (trainLM) and Resilient Backpropagation algorithm (trainRP) are used as the training functions whereas Log-sigmoid and Tan-sigmoid are used as the transfer function. Total of 810 data sets were generated out of which 11 data units for design of PT slab, failed in design. Hence the final database consists of 799 data units to be fed into the network for training the data. Four different validation techniques have been employed for validation of the selected network. These are the Resubstitution, Holdout, Three way data split and the k-fold cross validation techniques.

Various NN models have been developed for determining the deflection and weight of post tensioning steel required in PT slabs. The difference in each network lies in their architecture i.e. the number of hidden layers, the number of hidden nodes or neurons in the hidden layer, the number of input nodes and the number of output nodes. As there is no set guidelines for the selection of ANN model for a particular type of problem, here in this research work, architecture of ANN model is approached by making several trials.

For deciding the number of layers in a network, firstly the single layer networks were used. The analysis of the result showed poor conformance of the training results with the target dataset. This compelled for trying two layered ANN models. Results obtained by two layered networks were not only encouraging but also giving the realistic values. A comparative study, observations and result analysis for the same have already been incorporated in the thesis. Since, application of double layered networks gave us the desired level of accuracy in the problem domain, there was no valid reason for going for ANN models with three layers.

Determining the optimum number of nodes in each layer of ANN model is quiet important. A network model may not be able to converge well with a lesser number of nodes. At the same time, with excess nodes, the problem of overfitting may occur (Reich and Barai 2000). In this study, the optimized number of nodes were determined by making several trials. For single layered networks, hidden layer neurons were taken as 5, 10, 15 and 20 whereas for double layered the hidden layer neurons were taken as 5, 7 and 9 in each layer. It was observed that for single layered networks, training values were very close for 15 and 20 number of nodes. Similarly for double layered networks training values for 7 and 9 number of neurons were found to be quiet close. Hence higher number of nodes have not been experimented upon in this research work.

Primarily, five major design parameters have been taken as the inputs for the ANN models. These five inputs namely span of slab, depth of slab, live load per unit area of slab, column size and grade of concrete. These parameters mostly governs the overall design of the flat slab. The outputs for the network were taken as deflection and weight of post tensioning steel in the PT slab. For larger spans deflection is one of the major criterion for design and quantity of steel, on the other hand, is a major factor governing the oval economy of the project.

These networks are trained using several existing training algorithms; the popular one being backpropagation algorithm [Dahou Z. et al. 2009]. As per the limited literature available on the training algorithms, Levenberg-Marquardt backpropagation algorithm and Resilient backpropagation algorithm gives encouraging results [Kisi.O and Uncuoglu. E 2005] and hence these training algorithms have been used for training the developed NN models.

For activating the NN models, again there are several transfer functions readily available in the MATLABs NN toolbox. In PT slabs, the deflection values are primarily positive (sagging) but at some instances, it also comes out to be negative (hogging). The values for the weight of post tensioning steel are always positive. Hence, we have used Log sigmoid (logsig) transfer function which gives values of output between 1 and 0 with sigmoidal variation and Hyperbolic tangent sigmoid (tansig) transfer function which gives the output in the range of -1 to +1.

4.5 Data Preprocessing

Data preprocessing means the preparation of data in an acceptable range prior to the feeding of the data in the established neural network model for training. It is done for normalizing the data provided to the network. This is important as the data generated for the inputs and the outputs may be of different magnitude which may differ quiet largely. For example, in the present case, spans of PT-slabs have been taken in meters whereas the deflection has been taken in millimeters. Neural model in such cases may not be able to judge well and hence the results produced may not be up to the desired level of precision. Hence, it can be thought of to bring down all the inputs and the outputs in the same range say from 0 to +1 or from -1 to +1 as the case may be. In the present study data have been normalized in between the range of -1 to +1. Since the entire data is distributed uniformly after applying preprocessing, the training becomes rapid and the network becomes more efficient.

4.6 Creating the Network Neuron

As stated earlier, ANNs are such networks which consist of a number of small and powerful computing units called nodes or neurons. A neuron can be considered as a processing element which receives the input signals from the input layer and generates an output pulse for the next connected neuron. Each neuron is connected to every other neuron by a weight and a bias which indicates the strength of the network.

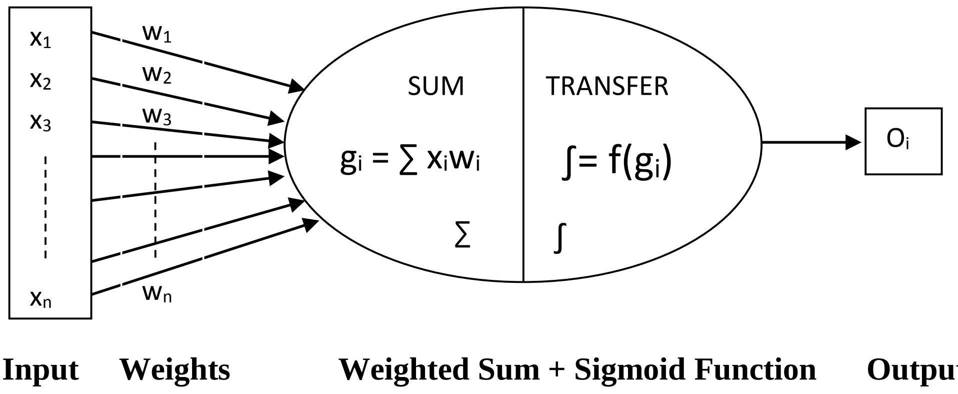


Fig 4.2: Structure of an Artificial Neuron

Fig 4.2 shows the mathematical model of an artificial neuron. The processing of information in the network takes place through these interconnected artificial neurons only. It implies that these artificial neurons play a very important role in the learning process of the network in addition to the validation and testing of the networks.

This artificial neuron receives signals from all the inputs which are fed into the network. Some random weights are initially associated with each of the inputs. As the signals reach the neuron, weighted summision takes place, i.e., each input vector is multiplied with its corresponding weight vector and all the multiplied units are then added. Mathematically it can be represented as:

$$g(x) = \sum x_i w_i = x_1 w_1 + x_1 w_1 + x_1 w_1 + \text{- - - - - - - - -}\ x_n w_n \qquad \text{------------} \quad 4.1$$

Now the entire weighted sum goes through a sigmoidal transfer function. With the help of a transfer function a relationship is developed between the inputs and the outputs which help the network to learn and generalize in a better way. Transfer function (T) can be expressed as:

$$T = f(x) = f(\sum x_i w_i) \qquad \text{------------} \quad 4.2$$

The number of neurons in the network is adjusted by trial and error method. In this research work, the single layer networks with five neurons initially are adopted and are increased by five additional neurons up to twenty five neurons. When switched to double layer networks, starting has been done by taking five neurons in each of the two layers and increased their number by two neurons till there were nine neurons in each layer. Maximum of nine neurons, in case of two layer networks, have been taken since two layered networks showed much better results than single layered networks with only nine neurons in each layer.

4.7 Activation/ Transfer Function

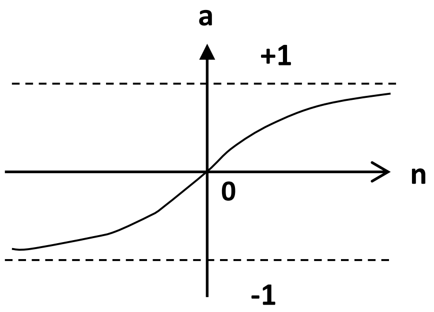
Transfer functions which are also known as activation functions are used for approximating the relationship between the inputs and the outputs of the database provided to the network. With backpropagation algorithm, sigmoidal transfer functions are utilized. This is because of the fact that these functions are differentiable and hence are capable of converging the errors to a minimum. Other non-differentiable functions are not capable of doing so and hence are not of much interest among the researchers working in the field of artificial intelligence. These sigmoidal functions are used for classification of various types of patterns related problems and also empower the neural network for solving problems related to nonlinearity. These are capable of bringing the large input values to really a small value between 0 to +1 or from -1 to +1 depending upon the type of sigmoidal transfer function used. Table 4.2 shows different types of transfer functions.

In this study transfer functions have been chosen which could be best suited to the database and which are capable of giving the appropriate results. Sigmoidal functions because of their advantage of being differentiable and bringing the error to a minimum, are the first choice to be included in this study. For the same reason log sigmoid transfer function (logsig) having range from 0 to +1 and hyperbolic tangent sigmoid transfer function (tansig) having range from -1 to +1 are taken as the activation functions in the selected neural network architecture for the hidden layers. For the output layer linear transfer function (purelin) has been adopted to take the actual values.

Table 4.2: Various transfer functions

Notation	Transfer Function	Performance
compet	Competitive transfer function	It returns 1 for the maximum input value and 0 at all other input values
hardlim	Hard limit transfer function	It gives only two options; either 1 (when threshold is reached) or 0
hardlims	Symmetric hard limit transfer function	It gives only two options; either 1 (when threshold is reached) or -1
logsig	Log sigmoid transfer function	It returns the values of output between 1 and 0 with sigmoidal variation
poslin	Positive linear transfer function	It gives the positive values between 1 and 0 with linear variation
purelin	Linear transfer function	It gives the value of output from -1 to +1 with linear variation
radbas	Radial basis transfer function	It uses radial function to give the output values in both directions.
satlin	Saturating linear transfer function	It varies the values linearly from 0 to +1 and beyond these values gives constant result as +1
satlins	Symmetric saturating linear transfer function	It varies the values linearly from +1 to -1 and beyond these values gives constant result as +1 or -1
softmax	Softmax transfer function	It gives values for inputs between +1 and 0 maintaining their aspect ratio
tansig	Hyperbolic tangent sigmoid transfer function	It gives the output in the range of -1 to +1
tribas	Triangular basis transfer function	When the input is 0 this function returns +1 and as the input reaches +1 of -1, the output linearly varies and ends up at 0.

Hyperbolic tangent sigmoid transfer function (tansig) has the same shape as of hyperbolic tangent curve as shown in Fig 4.3:



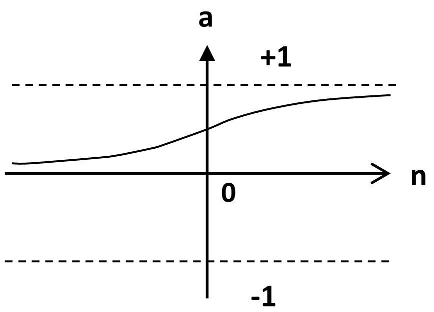
a = tansig(n)

Fig 4.3: Tan-Sigmoid Transfer Function

The expression for the algorithm of hyperbolic tangent sigmoid transfer function (tansig) is given as:

$$\text{Tansig}(n) = 2 / [1 + \exp(-2n)] - 1 \qquad \text{------------} \quad 4.3$$

Log sigmoid transfer function (logsig) resembles of the same shape as of the logarithmic function as shown in Fig 4.4:



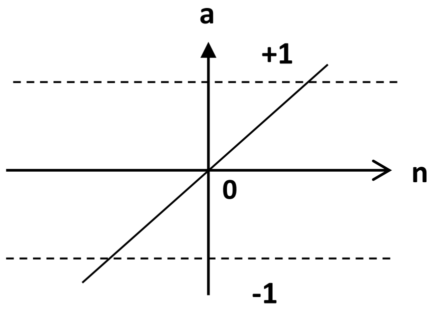
a = logsig(n)

Fig 4.4: Log-Sigmoid Transfer Function

The expression for the algorithm of log sigmoid transfer function (logsig) is given as:

$$\text{Logsig}(n) = 1 / [1 + \exp(-n)] \qquad \text{------------} \quad 4.4$$

Linear transfer function (purelin) represents the linear variation between the inputs and the outputs and is graphically shown in Fig 4.5:



a = purelin(n)

Fig 4.5: Purelin Transfer Function

The expression for the algorithm of linear transfer function (purelin) is given mathematically as:

$$Purelin(n) = n \qquad ------------ \quad 4.5$$

4.8 Hidden Layer(s)

Hidden layers are one of the most important components of the multilayered feed forward neural networks with backpropagation algorithm. This is the layer in which the hidden neurons make their place. The number of hidden layers in a neural network may be taken as one or more depending on the complexity of the problem under consideration. Increase in the number of hidden layer provides much better generalization of the problem statement. Neural networks with more than one hidden layer are best suited for problems relating to functions approximation. In general a neural network model with two hidden layers gives precise results for maximum real life problems. It implies that more the number of hidden layers more accurate will be the results given by the network.

4.9 Hidden Layer Neurons

A neuron is one of the most important components of the neural network architecture. It is responsible for computing the threshold of the inputs and transferring the output signals further to the next neuron. All the information of the network remains inside these neurons only. These are also referred to as small and powerful computing devices which process the entire database provided to the network. It is a matter of concern that what should be the number of neurons to be adopted in the hidden layer(s) so as to arrive at the desired level of accuracy between the output and the target values. It has been already discussed before that there is no rule or guideline yet to show us the way towards selecting the number of neurons in the hidden layer(s). The best way is to try out developing some neural networks and train them having various neurons in the hidden layer and check the networks precision. Generally it is seen that for the single hidden layer, the number of hidden layer neurons to achieve at a particular goal is much more than the number of hidden layer neurons of two or more hidden layers for the same problem. Also in case of single layer neurons, time taken for the convergence of error is comparatively more than that taken by the multilayer networks. From the above discussion it can be said that with increase in number of neurons, whether in single layer networks or in multilayer networks, the networks power of learning increases too.

4.10 Output Measurement

After the network has undergone training, it becomes quiet essential to measure the degree of correctness of the trained data as compared to the target data. This task is accomplished by using an output function (also referred as performance of objective function). This function evaluates the difference between the output and the target values so that the adopted network architecture can be validated. Basically, two types of performance functions are commonly used. The first one is the Sum of Squared Errors (SSE).

$$SSE = \sum_{i=1}^{n} (t_i - o_i)^2 \qquad \text{------------} \quad 4.6$$

Here, n represents the total number of database units provided to the network, t_i is representing the i^{th} target value and o_i is standing for the output value as been given by the neural network.

One another measurement function for estimating the error between the target and the output values is the Mean Square Errors (MSE), in which the errors are first squared, then summed up and finally, the average is determined. The expression for the MSE is given as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(t_i - o_i)^2 \qquad \text{------------} \quad 4.7$$

Here, n represents the total number of database units provided to the network, t_i is representing the i^{th} target value and o_i is standing for the output value as been given by the neural network.

4.11 Training Function

It is of much importance that network architecture developed must be able to learn from the examples which are fed into it. For serving this purpose several training functions or algorithms are utilized as shown in Table 4.3.

Table 4.3: Various training functions

Notation	**Training Algorithm**
trainb	Batch training with weight and bias learning rules.
trainbfg	BFGS quasi-Newton backpropagation.
trainbr	Bayesian regularization.
trainc	Cyclical order incremental update.
traincgb	Powell-Beale conjugate gradient backpropagation.
traincgf	Fletcher-Powell conjugate gradient backpropagation.
traincgp	Polak-Ribiere conjugate gradient backpropagation.
traingd	Gradient descent backpropagation.
traingda	Gradient descent with adaptive lr backpropagation.
traingdm	Gradient descent with momentum backpropagation.
traingdx	Gradient descent with momentum and adaptive lr backprop.
trainlm	Levenberg-Marquardt backpropagation.
trainoss	One step secant backpropagation.
trainr	Random order incremental update.
trainrp	Resilient backpropagation
trains	Sequential order incremental update.
trainscg	Scaled conjugate gradient backpropagation.

These training algorithms help the network to learn the output values corresponding to each input unit of the database. This process comprise of adjusting the weights corresponding to each node with each cycle of iteration so as to minimize the error. Thus when the neural network model undergoes such training for a number of times, it develops its own intelligence and gives acceptable results even when such data is tested by the network, which was not the part of the network's training. This is the main objective of the training function for the neural network.

In the present problem we have used Levenberg-Marquardt backpropagation algorithm (trainLM) and Resilient backpropagation algorithm (trainRP) for training the datasets provided to the neural network. With these training functions training becomes fast and the network's generalization power is increased as compared to the other training functions.

4.12 Validation

Validation refers to a process under which one can make surety of the correctness of his work. This process proves that the person either has or is arriving at the right destination. It shows the level of correctness of data. When the system is complex and the required reliability is high, system validation can best be accomplished by performing a number of tests that exercise different portions of the system, the results being mathematically combined to obtain the overall reliability figure, [Swem *et al.* 1994]. However as per [Twomey *et al.* 1998], for large samples, any validation methodology can be adopted to validate the network but when the database is small, the type of validation technique used will have an effect on the performance of the validated set. When the problem database is fed into the network for training the neural network model, with each increase in the number of epochs, the error goes on decreasing and stage comes at which the error becomes almost constant i.e., there is no considerable change in target and the output value. At this stage the neural network model is said to be fully trained. Now it becomes necessary at this time to evaluate the efficiency of the trained network. For the same purpose, the portion of database, which are not the part of training, is shown to the network for achieving the outputs. These outputs are compared with the target values and MSE is computed. This process is repeated with different possible neural network architectures for acquiring the networks efficiency. This whole procedure is known as validation.

When any network undergoes validation, its reliability can be proved on the basis that how well the data fits in the selected network. More fine the data is fitted in the selected network, more accurate will be the results obtained. The main problem in aspect of validation is that the level of the actual noise, which is the unpredictable part of the measured data, is always unknown, [Zhang *et al.* 2009]. Validation is a crucial part to the development of a sound empirical model. To be beneficial, system models must be validated to assure the users that the model emulates the actual system in the desired manner. This is especially true of empirical models, such as neural network and statistical models, which rely primarily on observed data rather than analytical equations derived from first principles [Twomey *et al* 1998].

4.12.1 Validation Techniques:

Many researchers working on mathematical models have tried out various techniques of validation of their developed models. A combination of enhanced and higher order auto correlation functions (ACF) and cross correlation functions (CCF) tests were conducted by Zhang *et al.* 2009, for validating their model. A new methodology proposed by Wang *et al*. 2008, used a density function for expanded training in the validation process. This method was said to be having improved generalization ability. Hirooka *et al.* 1996, used the numeric simulation methodology for validating the data. The second derivative of validation error based regularization algorithm (SDVR) was derived by Chen *et al*. 1999, based on Gauss-Newton approximation. One of the validation techniques developed by Hull *et al*. 2002, have the capability of identifying the problematic space in the database when the output obtained is not of desired level.

The following four types of data validation techniques have been employed in the selection of optimized neural model:

4.12.1.1 Resubstitution Validation Technique:

Resubstitution validation technique is a validation process in which the entire data set which has undergone training has also to undergo the testing process. This means that in such validation type, since network has "seen" all the data, so the mean square error obtained after the validation of network may not give the true picture of the network performance and the results obtained may be showing quiet low values of

MSE, are just “apparent”. This may be because the network has learned or memorized the data. So this type of validation process cannot be alone relied upon and one must validate the network models using other validation techniques also.

4.12.1.2 Holdout Validation Technique:

Holdout validation technique is a validation process in which the two-third data has to undergo training process. The validation of network is done using the remaining one-third data. This type of validation technique is much better for real world problems. As the data used for training and testing are different and also the network model has not seen the testing data while undergoing training, so the results obtained in terms of MSE are more realistic. The results from this validation technique may be showing somewhat higher values of MSE, but will definitely lead to the proper validation of network model. Validation with this type of technique can thus be relied upon. A similar validation technique was used by Larsen et al. 1996, in which he divided the entire database into two parts, keeping one part for training the data and other part for testing the data. Tsai et al. 2010 also used the same technique with statistical validation of neural networks.

4.12.1.3 Three Way Data Split Validation Technique

Three Way Data Split Validation technique is a validation process in which the entire data is divided into three equal parts. First part is used for training the network model, while the second part is used for validating the network models. The third part is kept for testing the selected neural network model. In this type of validation process, the data used to train the network is quiet less, only one-third, as compared to the entire data set. So there may be a possibility that the neural network model may not be properly trained. When such network model undergoes validation, the variation in results will be considerable. It means that this type of validation technique, although validated and tested through unseen data sets, may not be fruitful for practical problems, owing to large variations in results. Similar validation technique has been employed by Mallet *et al*. 2006**,** in their study in which the entire data was divided into three parts. First part used for training was 40% of the entire database, second part for validation was of 30% and the third part for testing the network was again of 30% of the database.

4.12.1.4 K-Fold Cross Validation Technique

A somewhat standard statistical technique for coping with the generalization error is cross-validation, Wang *et al* 2008. Cross-validation has been widely used for estimating the performance of neural networks and early stopping of training, Liu Y. *et al* 2006. In the process of k-Fold Cross Validation technique, the entire data is divided into k (1, 2, 3, 4…, k) parts. From these divisions, one part is kept for testing and the remaining (k-1) part undergo training. In this manner, the neural network model undergoes training k times and each part, one to k, gets a chance to undergo testing. Out of these k validations, the network model showing the least mean square error is then selected as the desired neural network model. The value of k is taken as ten for obtaining optimized results. It means that each time, ninty percent data is being trained and ten percent data being tested. Technique of k-fold cross validation was employed for data validation by Mojarad *et. al* 2010, analysing the network performance for cancer progression. Pruning of neural networks was done using cross validation technique by Huynh *et al* 2005.

4.13 Summary

Methodology adopted for carrying out this research work have been presented in this chapter. Various configurations of PT slabs have been worked out on the basis of its span, depth, live load, column size and grade of concrete. These slabs have been analysed for the maximum deflection at the centre of slab panel and minimum weight of post tensioning steel required per metre area of the slab. One of the most distinguished techniques of artificial intelligence, ANNs have been employed as a conceptual design tool for the analysis of PT slabs. Several models of ANNs with varying architecture have been proposed and worked upon for selecting the best performing network. Here the number of hidden layers, number of hidden layer neurons, training function, transfer function and other parameters required for ANNs training are analysed. Important design parameters essential for the design of PT slabs have been. For validating the trained network, various validation techniques have been employed and compared. This chapter discusses all the parameters for developing the ANNs including its training and validation.

CHAPTER-5

Network Validation Results and Analysis

5.1 Introduction

In this research problem, efforts are made to model such a network, which is able to calculate the deflection and weight of PT steel required, very close to the target values. Modelling of PT-slab with drop panels is commenced using standard design software. Various parameters involved in the design and analysis of PT-slab are shown in Table 5.1:

Table 5.1: Design data for neural network

S.No.	PARAMETERS	VALIDATION RANGE	UNITS	No.	TOTAL DATA
1	Span	7, 8, 9, 10, 11 and 12	m	6	6x5x3x3x3 = 810 **Out of these, 11 combinations failed in design. So Total Data, 810 – 11 = 799**
2	Depth	170, 190, 210, 230 and 250	mm	5	
3	Load	3, 4, and 5	kN/m^2	3	
4	Column size	450x450, 600x600 and 750x750	mm^2	3	
5	Concrete grade	M35, M40 and M45	N/mm^2	3	

For selecting the desired neural network model, the data must be validated. In this process, the data is trained with increasing number of epochs or iterations. If the performance of the network (Goal), in terms of MSE is less than the previous goal, then this value is accepted and the network can be now trained at next higher level of epochs. This process continues till the performance becomes constant at two consecutive iteration levels or till the goal is reached. Four different validation techniques are used in selecting the model giving the best performance. These are as follows:

1. Resubstitution validation technique
    - 799 dataunits for training, same 799 for testing
2. Holdout validation technique
    - 533 dataunits for training, 266 for testing
3. Three-way data split validation technique
    - 267 dataunits for training, 266 for validation, 266 for testing
4. K-fold cross validation technique
    - 719 dataunits for training, 80 for testing

Training, validation and testing of each network is done using neural network toolbox in MATLAB software supported by an excel program. Using each of these validation techniques, various neural networks have been modelled. Twelve different combinations have been considered, out of which eight combinations are with two hidden layers and four are with single hidden layer. In each combination, three sets with different numbers of neurons have been created. Each neural network developed have been named according to its architecture. For eg. a single layered network with training function as trainLM and transfer function as logsigmoid is named as 1L_LM_Log.

5.2 Networks for Resubstitution validation technique:

The single layer networks are shown in Table 5.2 and the double layer networks are shown in Table 5.3 for resubstitution validation technique. Total four network architectures are developed for single layer networks and eight are developed for double layer networks.

Table 5.2: Single layer networks for resubstitution validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
1.	NET1	1	20	Resilient	Tan sigmoid
2.	NET2	1	20	Resilient	Log sigmoid
3.	NET3	1	20	Levenberg Marquardt	Tan sigmoid
4.	NET4	1	20	Levenberg Marquardt	Log sigmoid

Table 5.3: Double layer networks for resubstitution validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
5.	NET5	2	9	Levenberg Marquardt	Tan sigmoid Tan sigmoid
6.	NET6	2	9	Levenberg Marquardt	Log sigmoid Log sigmoid
7.	NET7	2	9	Levenberg Marquardt	Tan sigmoid Log sigmoid
8.	NET8	2	9	Levenberg Marquardt	Log sigmoid Tan sigmoid
9.	NET9	2	9	Resilient	Tan sigmoid Tan sigmoid
10.	NET10	2	9	Resilient	Log sigmoid Log sigmoid
11.	NET11	2	9	Resilient	Tan sigmoid Log sigmoid
12.	NET12	2	9	Resilient	Log sigmoid Tan sigmoid

Single layer networks:

Variation of deflection of PT slab and weight of PT steel, with the number of training epochs for single layered neural networks have been discussed here. Single layer networks have been created and resubstitution validation technique have been employed for the selection of optimized neural network model. Each network has been validated with different number of neurons, i.e., (5, 10, 15 and 20) in the hidden layer. It is seen that the network performance is best when there are 20 number of neurons in the hidden layer. Mean Square Error (MSE) has been recorded at an interval of hundred epochs upto a maximum of 1000 epochs.

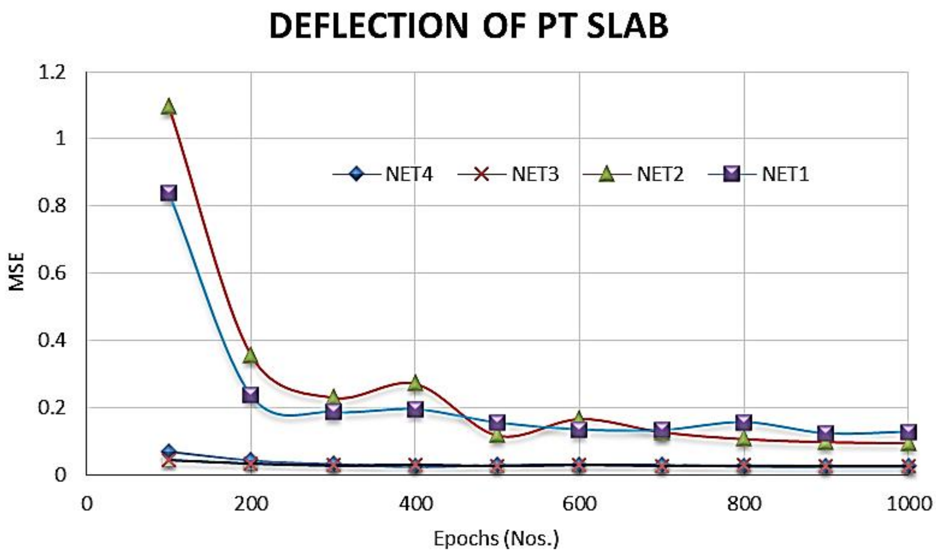


Fig 5.1: Deflection models (4 nets)

In Fig 5.1, four such network models, NET1 to NET4, with single layer have been compared against the minimum MSE for deflection of PT slab. The variation in average MSE for deflection, of network models NET1 and NET2, lies in the range approximately from 0.2 to 0.1, when seen in 500 to 1000 epochs. On the other hand network models NET3 and NET4 shows average MSE for deflection under the range of 0.07 for all epochs, which can be seen as a horizontal line. It is quiet evident from the graph that network models, NET3 and NET4, stands reasonably better as compared to network models, NET1 and NET2.

In Fig 5.2, network models NET3 and NET4 are ploted in the scale of 0 to 0.07 for more better comparison, so as to select the optimized network. In this range, although, the networks takes off in the range of 0.07 to 0.04, but eventually, both the networks tend to merge towards the mark of 0.025 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.025 at 1000 epochs.

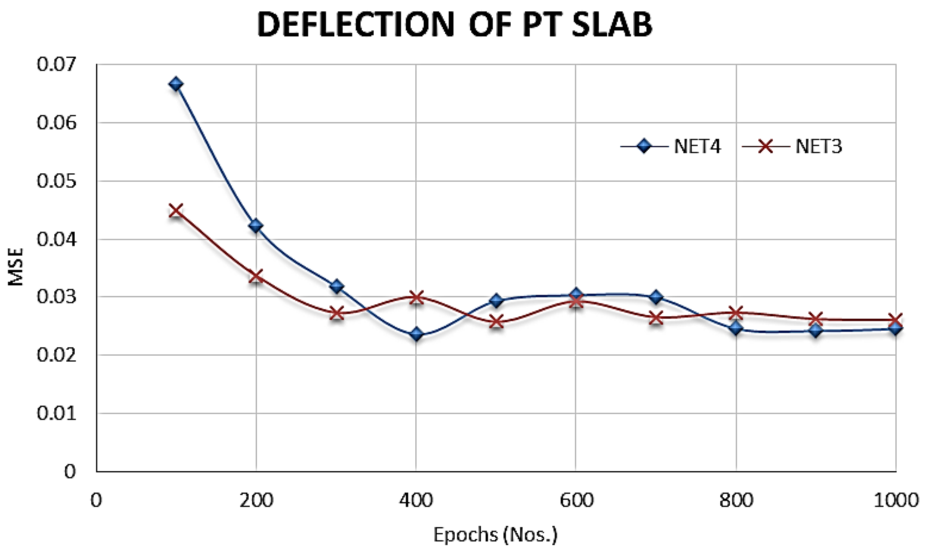


Fig 5.2: Optimized deflection models (2 nets)

Training MSE for deflection of PT slab of network model, NET4: 1L_LM_Log with twenty neurons in the hidden layer, comes out to be 0.024548 at 1000 epochs. When the entire data undergoes testing , the **MSE comes out to be 0.02484**.

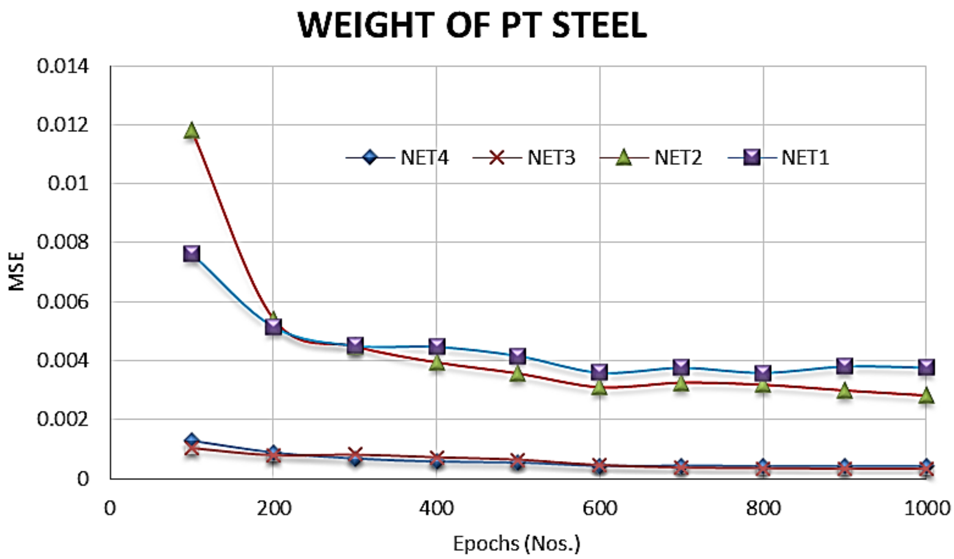


Fig 5.3: Weight models (4 nets)

Fig 5.3, shows the same network models as above, NET1 to NET4, for comparison against the minimum MSE for weight of PT steel. The variation in average MSE for weight of PT steel, of network models NET1 and NET2, lies in the range approximately from 0.004 to 0.002, when seen in 500 to 1000 epochs. Also the network models NET3 and NET4 shows average MSE for weight of PT steel under the range of 0.0014 for all epochs. It can be seen again that the same network models, NET3 and NET4, proves to be better as compared to network models, NET1 and NET2.

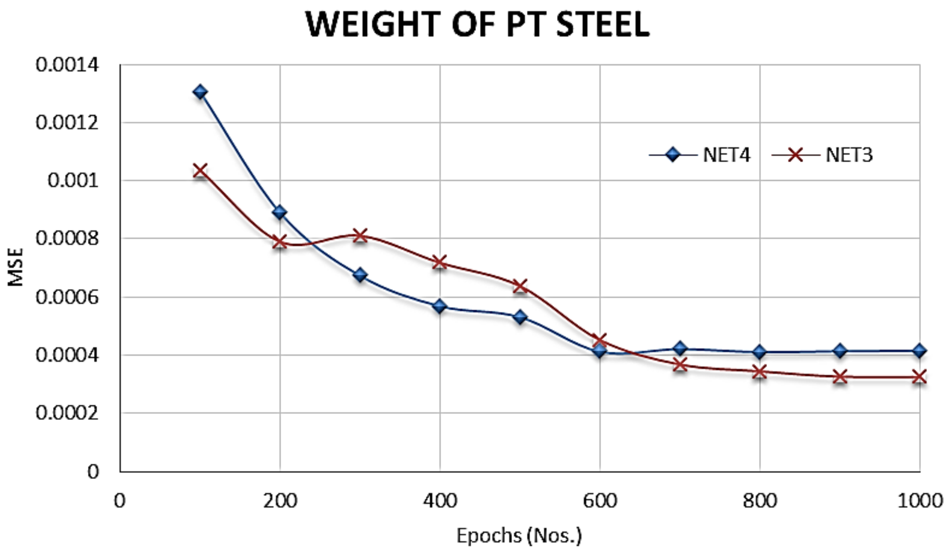


Fig 5.4: Optimized weight models (2 nets)

In Fig 5.4, network models NET3 and NET4 are ploted in the scale of 0 to 0.0014 for more better judgement. In this range, the networks takes off from the range of 0.0014 to 0.001, but eventually, both the networks tend to merge towards the mark of 0.00035 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.00035 at 1000 epochs.

Training MSE for weight of PT steel of network model, NET3: 1L_LM_Tan with twenty neurons in the hidden layer, comes out to be 0.0003264 at 1000 epochs. When the entire data undergoes testing , the **MSE comes out to be 0.00033** which is reasonably good as compared to other network models for single layer.

Double layer networks:

Double layer networks having variation of deflection with the number of iterations or epochs have been discussed here. Resubstitution validation technique have been employed for the selection of optimized neural network model. Each network have been validated with different sets of neurons, i.e., (5,5), (7,7) and (9,9) in first and second layers respectively. It is seen that the networks performance is best when the number of neurons in first and the second layers are (9,9) respectively. Mean Square Error (MSE) has been recorded at an interval of hundred epochs upto a maximum of 1000 epochs.

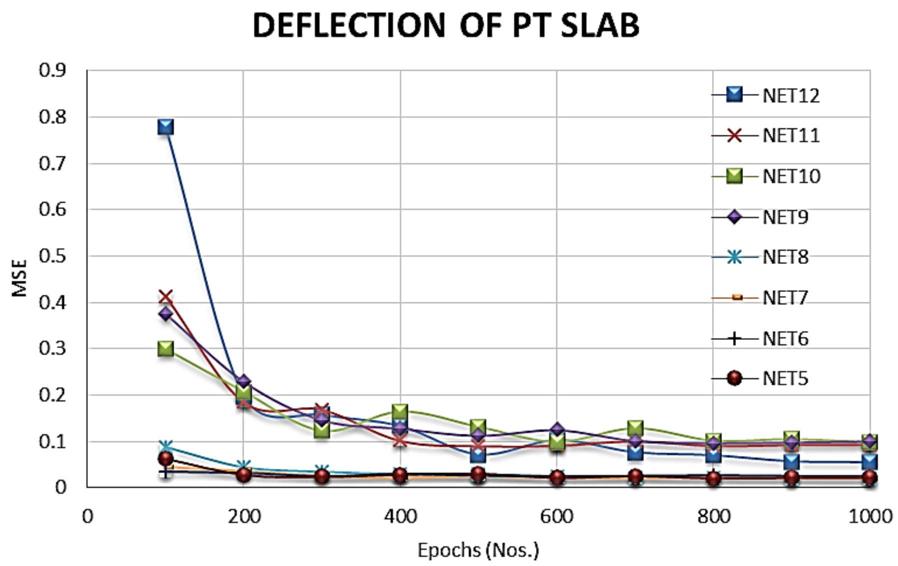


Fig 5.5: Deflection models (8 nets)

In Fig 5.5, eight such network models, NET5 to NET12, with double layers have been compared. The variation in average MSE for deflection of PT slab of network models NET9 to NET12 lies in the range approximately from 0.2 to 0.1, when seen in 400 to 1000 epochs, whereas network models NET5 to NET8 shows average MSE under the range of 0.09 for all epochs, which can be seen as a straight line. It is quiet evident from the graph that network models, NET5 to NET8, stands reasonably better as compared to network models, NET9 to NET12.

In Fig 5.6, network models NET5 to NET8 are ploted in the scale of 0 to 0.1 for more better comparision of the network models, so as to select the optimized network. In this range, although, the networks takes off in the range of 0.09 to 0.04, but eventually, all the networks tend to merge towards the mark of 0.02 from 600 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.02 at 1000 epochs.

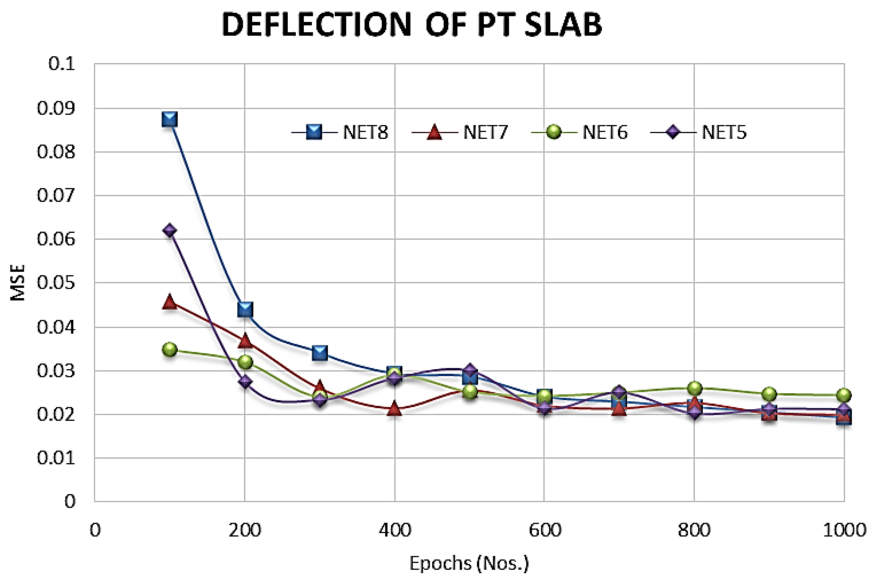


Fig 5.6: Optimized deflection models (4 nets)

Training MSE for deflection of PT slab of network model NET7: 2L_LM_Tan_Log with nine neurons in each layer comes out to be 0.0200615 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to be 0.01998** which is even less than the MSE for deflection of PT slab of single layer model, NET4: 1L_LM_Log.

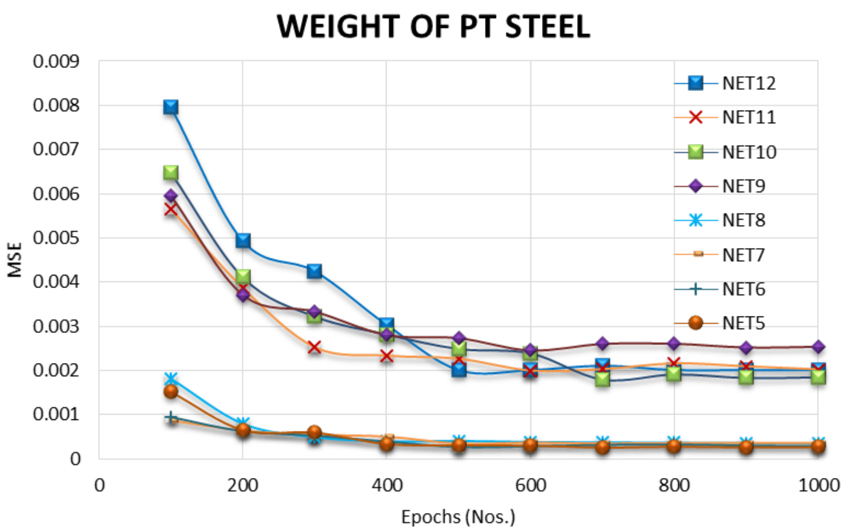


Fig 5.7: Weight models (8 nets)

In Fig 5.7, same eight network models, NET5 to NET12, have been adopted. The variation in average MSE for the weight of PT steel of network models NET9 to NET12 lies in the range approximately from 0.003 to 0.002, when seen in 500 to 1000 epochs, whereas network models NET5 to NET8 shows average MSE under the range of 0.001 for 500 to 1000 epochs, which can be seen as a straight line. It is quiet evident from the graph that network models, NET5 to NET8, are far better as compared to network models, NET9 to NET12.

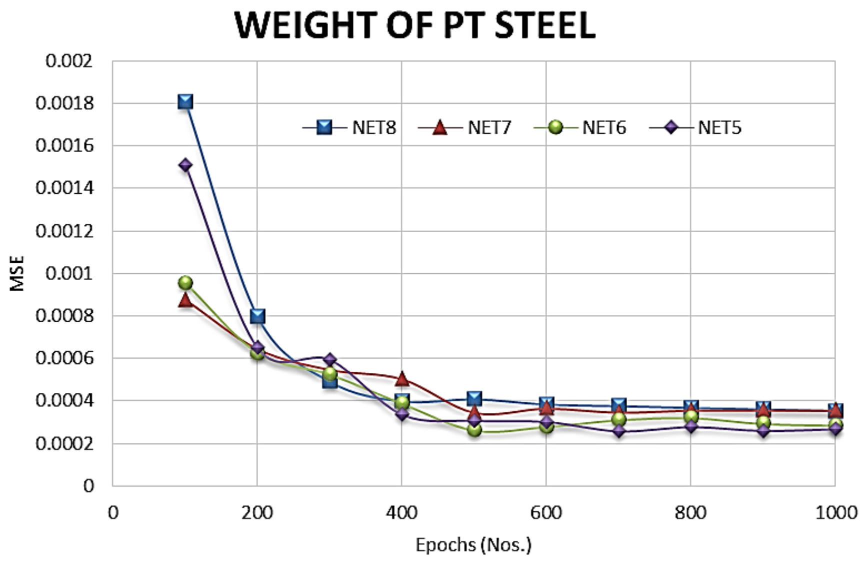


Fig 5.8: Optimized weight models (4 nets)

In Fig 5.8, network models NET5 to NET8 are ploted in the scale of 0 to 0.002 for better selection of optimized network. In this range, the networks takes off in the range of 0.008 to 0.0008, but eventually, all the networks tend to merge towards the mark of 0.0003 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.0003 at 1000 epochs.

Training MSE for weight of PT steel of network model NET5: 2L_LM_Tan_Tan with nine neurons in each layer comes out to be 0.0002679 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to be 0.00025** which is even less than MSE for weight of PT steel of single layer model, NET3: 1L_LM_Tan. The summary of results obtained by applying resubstitution validation tecnique are represented in Table 5.4 for deflection of PT-slabs and for weight of PT-steel.

Table 5.4: Summary of results of resubstitution validation technique

S.No.	Network Model	No. of layers	No. of neurons	Training MSE at 1000 epochs	Testing MSE
Deflection of PT slabs					
1.	NET4: 1L_LM_Log	1	20	0.024548	0.02484
2.	NET7: 2L_LM_Tan_Log	2	9,9	0.0200615	0.01998
Weight of PT steel					
3.	NET3: 1L_LM_Tan	1	20	0.0003264	0.00033
4.	NET5: 2L_LM_Tan_Tan	2	9,9	0.0002679	0.00025

The various selected network shows reasonably good results when compared with various other network models with single and double layers. Finally,from the results, it can be concluded that network model NET7: 2L_LM_Tan_Log and NET5: 2L_LM_Tan_Tan can be adopted as the optimized network models for Deflection of PT slabs and Weight of PT steel respectively in case of resubstitution validation technique.

5.3 Network models for holdout validation technique:

The single layer networks are shown in Table 5.5 and the double layer networks are shown in Table 5.6 for holdout validation technique. Total four network architectures are developed for single layer networks and eight are developed for double layer networks.

Table 5.5: Single layer networks for holdout validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
1.	NET1	1	20	Resilient	Tan sigmoid
2.	NET2	1	20	Resilient	Log sigmoid
3.	NET3	1	20	Levenberg Marquardt	Tan sigmoid
4.	NET4	1	20	Levenberg Marquardt	Log sigmoid

Table 5.6: Double layer networks for holdout validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
5.	NET5	2	9	Levenberg Marquardt	Tan sigmoid Tan sigmoid
6.	NET6	2	9	Levenberg Marquardt	Log sigmoid Log sigmoid
7.	NET7	2	9	Levenberg Marquardt	Tan sigmoid Log sigmoid
8.	NET8	2	9	Levenberg Marquardt	Log sigmoid Tan sigmoid
9.	NET9	2	9	Resilient	Tan sigmoid Tan sigmoid
10.	NET10	2	9	Resilient	Log sigmoid Log sigmoid
11.	NET11	2	9	Resilient	Tan sigmoid Log sigmoid
12.	NET12	2	9	Resilient	Log sigmoid Tan sigmoid

Single layer networks:

The variation of deflection of PT slab and weight of PT steel, with the number of epochs have been discussed here. Same single layer network models have been used, as taken in the previous case. Holdout validation technique have been employed for the selection of optimized neural network model.

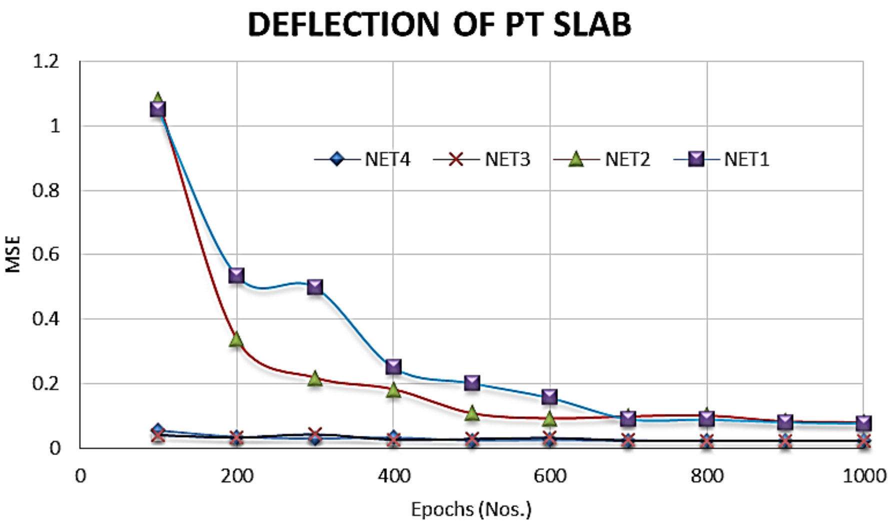


Fig 5.9: Deflection models (4 nets)

In Fig 5.9, four such network models, NET1 to NET4, have been validated to achieve minimum MSE for deflection of PT slab. The variation in average MSE for deflection, of network models NET1 and NET2, merges to 0.1, when seen in 500 to 1000 epochs. On the other hand network models NET3 and NET4 shows average MSE for deflection under the range of 0.06 for all epochs. It is clear from the graph that network models, NET3 and NET4, stands reasonably better as compared to network models, NET1 and NET2.

In Fig 5.10, we have plotted network models NET3 and NET4 in the scale of 0 to 0.06 for more better comparision. In this range, although, the networks takes off in the range of 0.06 to 0.04, but eventually, both the networks tend to merge towards the mark of 0.025 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.025 at 1000 epochs.

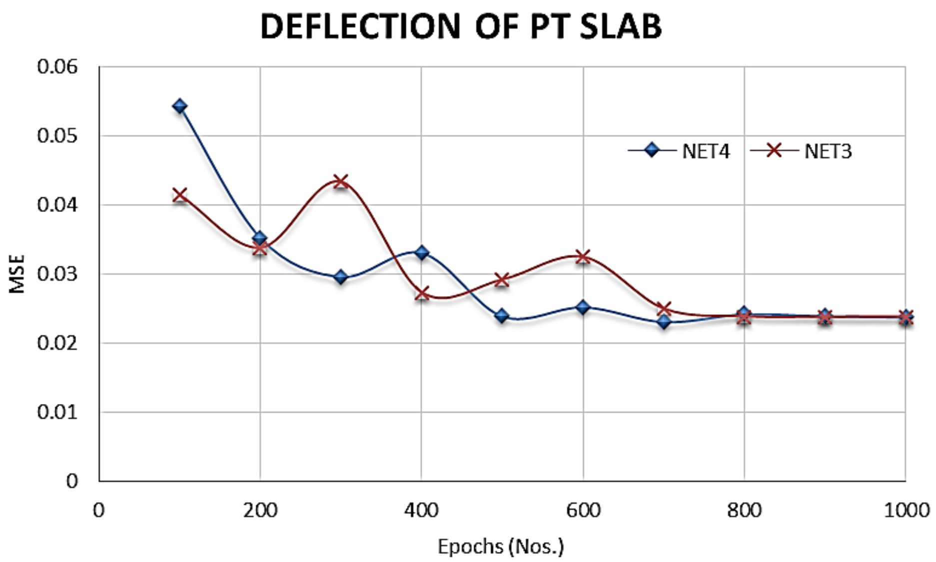


Fig 5.10: Optimized deflection models (2 nets)

Training MSE for deflection of PT slab of network model, NET3: 1L_LM_Tan with twenty neurons in the hidden layer, comes out to be 0.0238361 at 1000 epochs. When the entire data undergoes testing , the **MSE comes out to be 0.02048**.

Fig 5.11, shows the network models, NET1 to NET4, for comparision against the minimum MSE for weight of PT steel. The variation in average MSE for weight of PT steel, of network models NET1 and NET2, lies in the range approximately from 0.004 to 0.002, when seen in 500 to 1000 epochs. Also the network models NET3 and NET4 shows average MSE for weight of PT steel under the range of 0.0014 for all epochs. It can be seen again that the same network models, NET3 and NET4, proves to be better as compared to network models, NET1 and NET2.

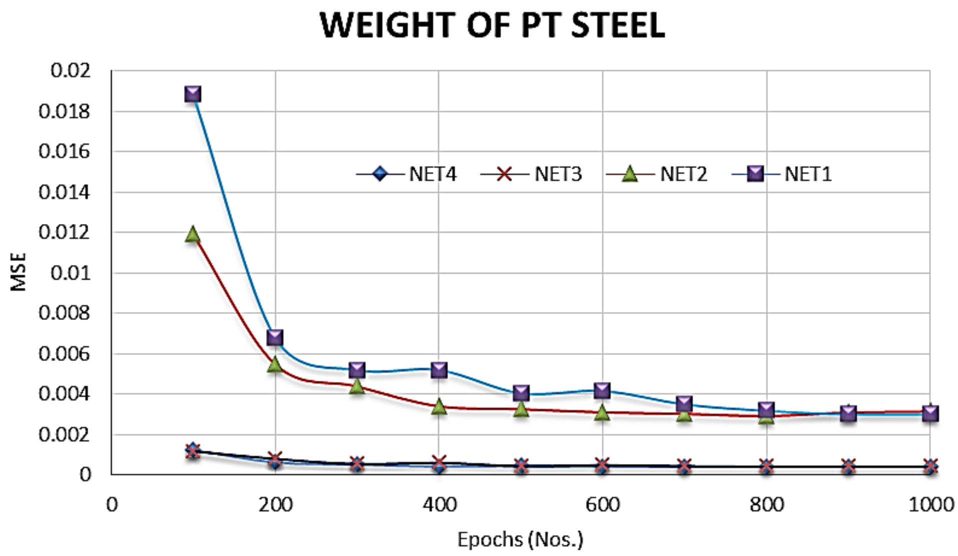


Fig 5.11: Weight models (4 nets)

In Fig 5.12, network models NET3 and NET4 are ploted in the scale of 0 to 0.0014 for more better judgement. In this range, the networks takes off from 0.0014, but eventually, both the networks tend to merge towards the mark of 0.0004 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.0004 at 1000 epochs.

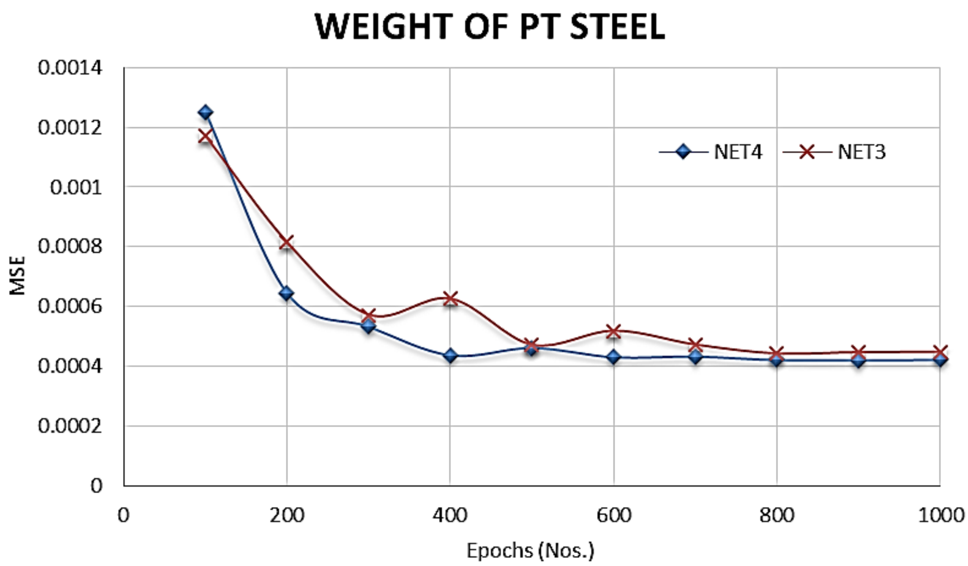


Fig 5.12: Optimized weight models (2 nets)

Training MSE for weight of PT steel of network model, NET4: 1L_LM_Log with twenty neurons in the hidden layer, comes out to be 0.0004208 at 1000 epochs.

When the entire data undergoes testing , **the MSE comes out to be 0.00042** which is reasonably good as compared to other network models for single layer.

Double layer networks:

Variation of deflection with the number of epochs have been discussed here. Holdout validation technique have been employed. Each network is validated with different number of neurons, i.e., (5,5), (7,7) and (9,9) in first and second layers respectively. It is seen that the networks are performing the best when the number of neurons in first and the second layers are (9,9) respectively. Mean Square Error (MSE) has been recorded at an interval of hundred epochs upto a maximum of 1000 epochs.

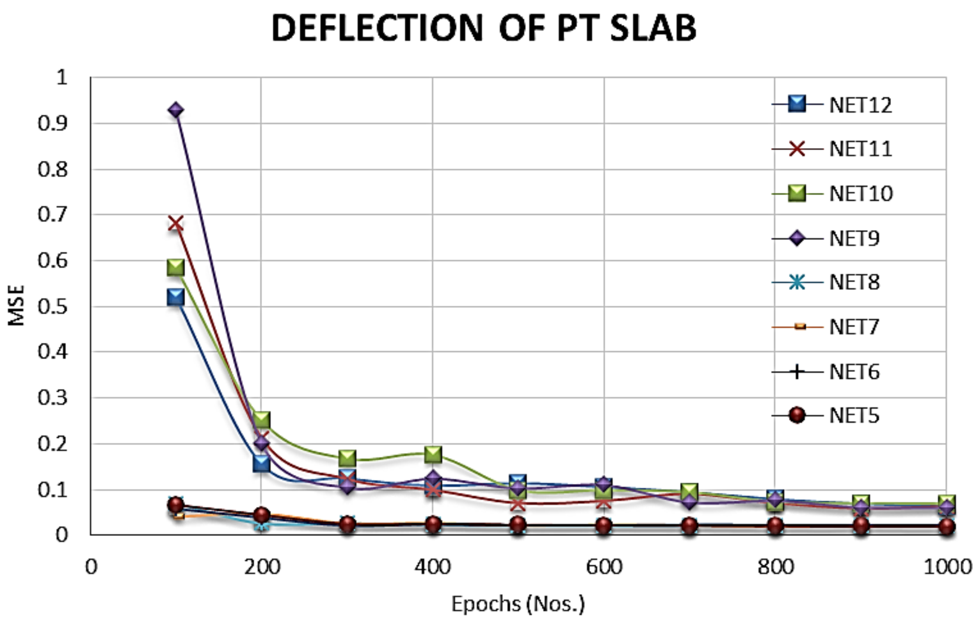


Fig 5.13: Deflection models (8 nets)

In Fig 5.13, eight network models, NET5 to NET12, have been compared. The variation in average MSE for deflection of PT slab of network models NET9 to NET12 lies in the range below 0.1, when seen in 400 to 1000 epochs, whereas network models NET5 to NET8 shows average MSE under the range of 0.03 for 400 to 1000 epochs. Network models, NET5 to NET8, stands far better, as compared to network models, NET9 to NET12.

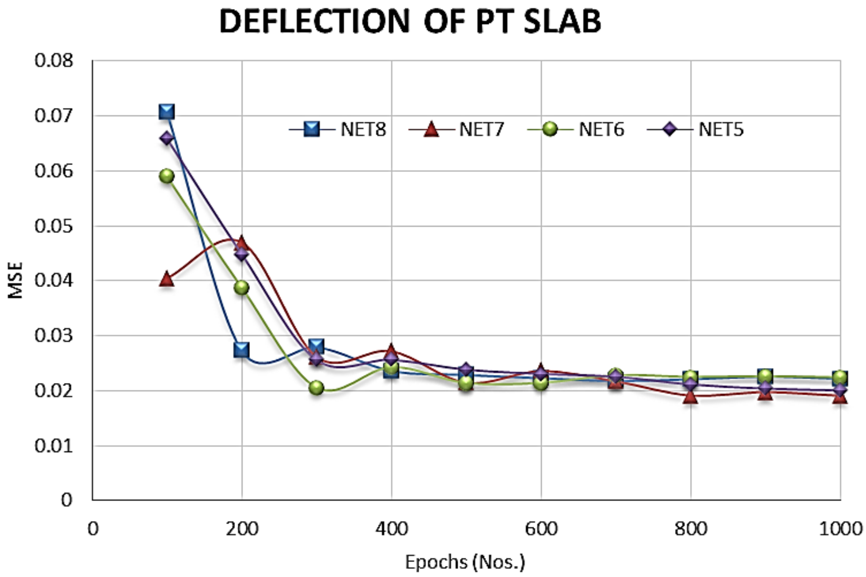


Fig 5.14: Optimized deflection models (4 nets)

In Fig 5.14, network models NET5 to NET8 are ploted in the scale of 0 to 0.08, so as to select the optimized network. In this range, although, the networks takes off in the range of 0.07 to 0.04, but eventually, all the networks tend to merge towards the mark of 0.02 from 400 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.02 at 1000 epochs.

Training MSE for deflection of PT slab of network model NET7:2L_LM_Tan_Log with nine neurons in each layer comes out to be 0.0191037 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to be 0.01599** which is even less than the MSE for deflection of PT slab of single layer model, NET4: 1L_LM_Log.

In Fig 5.15, eight network models, NET5 to NET12, have been adopted. The variation in average MSE for the weight of PT steel of network models NET9 to NET12 lies in the range approximately from 0.003 to 0.002, when seen in 500 to 1000 epochs, whereas network models NET5 to NET8 shows average MSE under the range of 0.001 for 300 to 1000 epochs. Network models, NET5 to NET8, are far better as compared to network models, NET9 to NET12.

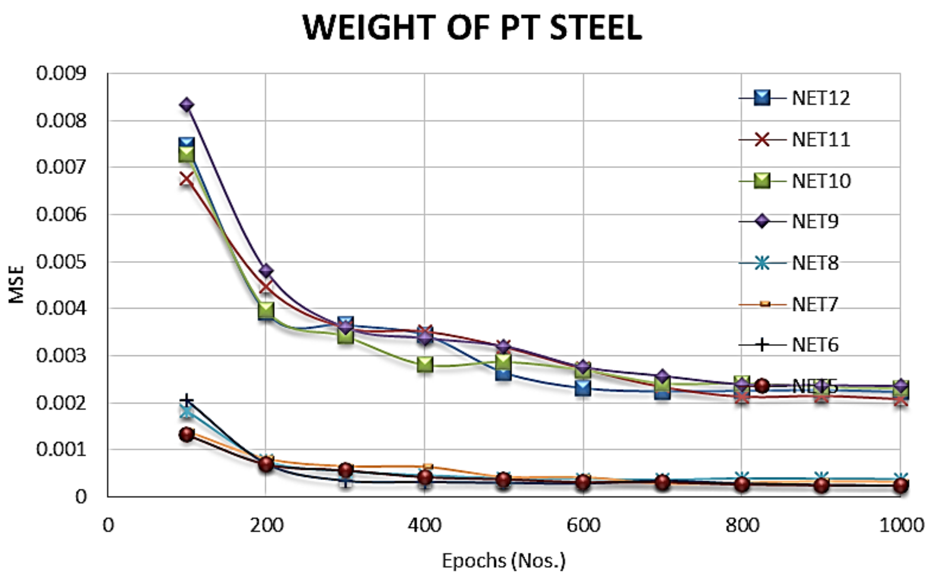


Fig 5.15: Weight models (8 nets)

In Fig 5.16, network models NET5 to NET8 are ploted in the scale of 0 to 0.0025. In this range, the networks takes off in the range of 0.002 to 0.0015, but eventually, all the networks tend to merge towards the mark of 0.0003 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.0003 at 1000 epochs.

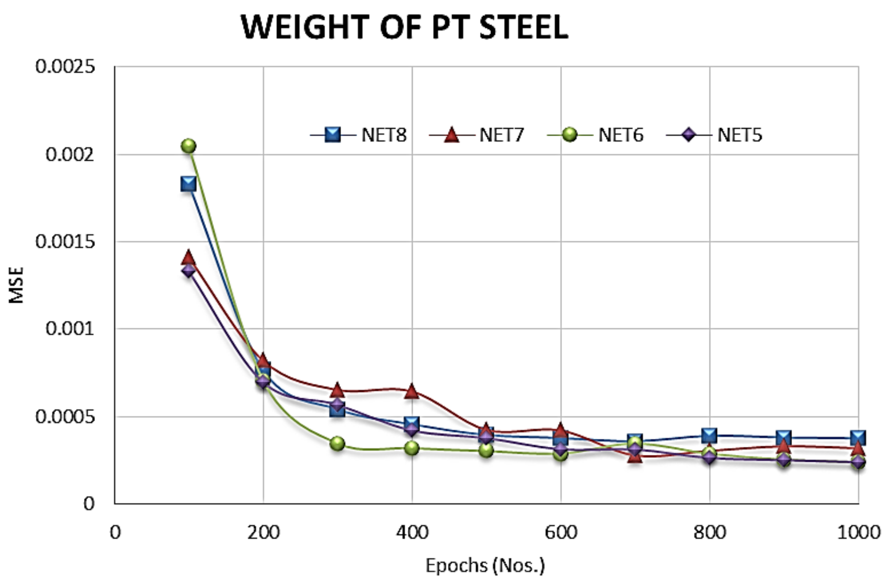


Fig 5.16: Optimized weight models (4 nets)

Training MSE for weight of PT steel of network model NET6: 2L_LM_Log_Log with nine neurons in each layer comes out to be 0.0002385 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to**

be 0.00021 which is even less than MSE for weight of PT steel of single layer model, NET4: 1L_LM_Log. The summary of results obtained by applying holdout validation tecnique are represented in Table 5.7 for deflection of PT-slabs and Table 5.8 for weight of PT-steel.

Table 5.7: Summary of results of holdout validation technique

S.No.	Network Model	No. of layers	No. of neurons	Training MSE at 1000 epochs	Testing MSE
Deflection of PT slabs					
1.	NET3: 1L_LM_Tan	1	20	0.0238361	0. 02048
2.	NET7: 2L_LM_Tan_Log	2	9,9	0. 0191037	0. 01599
Weight of PT steel					
3.	NET4: 1L_LM_Log	1	20	0. 0004208	0. 00042
4.	NET6: 2L_LM_Log_Log	2	9,9	0. 0002385	0. 00021

The various selected network shows reasonably good results when compared with various other network models with single and double layers. Finally, looking at the results, it can be concluded that network model NET7: 2L_LM_Tan_Log and NET6: 2L_LM_Log_Log can be adopted as the optimized network models for Deflection of PT slabs and Weight of PT steel respectively in case of holdout validation technique.

5.4 Networks for Three-Way data split validation technique:

The single layer networks are shown in Table 5.8 and the double layer networks are shown in Table 5.9 for three-way data split validation technique. Total four network architectures are developed for single layer networks and eight are developed for double layer networks.

Table 5.8: Single layer networks for three-way data split validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
1.	NET1	1	20	Resilient	Tan sigmoid
2.	NET2	1	20	Resilient	Log sigmoid
3.	NET3	1	20	Levenberg Marquardt	Tan sigmoid
4.	NET4	1	20	Levenberg Marquardt	Log sigmoid

Table 5.9: Double layer networks for three-way data split validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
5.	NET5	2	9	Levenberg Marquardt	Tan sigmoid Tan sigmoid
6.	NET6	2	9	Levenberg Marquardt	Log sigmoid Log sigmoid
7.	NET7	2	9	Levenberg Marquardt	Tan sigmoid Log sigmoid
8.	NET8	2	9	Levenberg Marquardt	Log sigmoid Tan sigmoid
9.	NET9	2	9	Resilient	Tan sigmoid Tan sigmoid
10.	NET10	2	9	Resilient	Log sigmoid Log sigmoid
11.	NET11	2	9	Resilient	Tan sigmoid Log sigmoid
12.	NET12	2	9	Resilient	Log sigmoid Tan sigmoid

Single layer networks:

The variation of deflection of PT slab and weight of PT steel, with the number of epochs have been discussed here. Three-way data split validation technique have been employed for the selection of optimized neural network model.

In Fig 5.17, four such network models, NET1 to NET4, have been validated to achieve minimum MSE for deflection of PT slab. Maximum number of epochs in NET1 and NET2 is 20 whereas in NET3 and NET4 it has been taken as 15. It is so because the selected goal is achieved in only 15 epochs in case of NET3 and NET4.

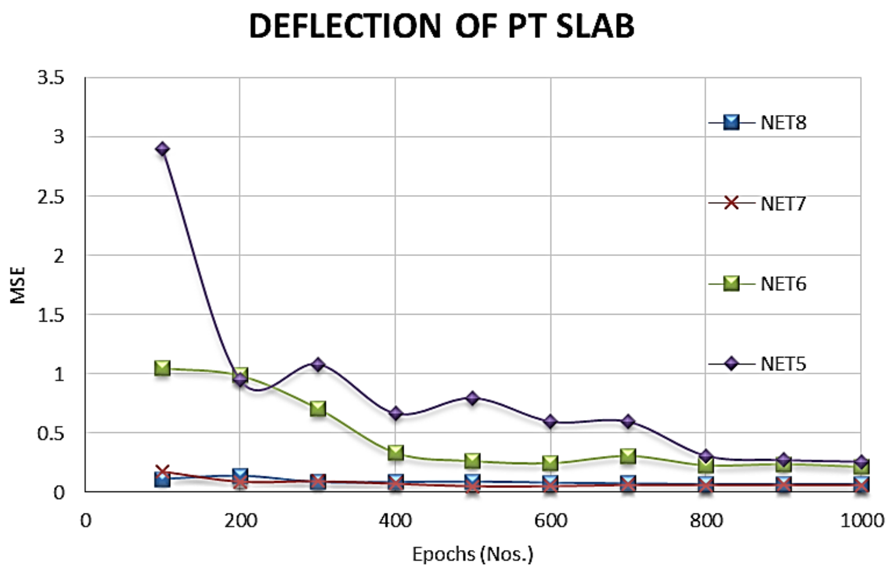


Fig 5.17: Deflection models (4 nets)

The variation in average MSE for deflection, of network models NET1 and NET2, merges to 0.25, when seen in 800 to 1000 epochs. On the other hand network models NET3 and NET4 shows average MSE for deflection under the range of 0.18 for all epochs.

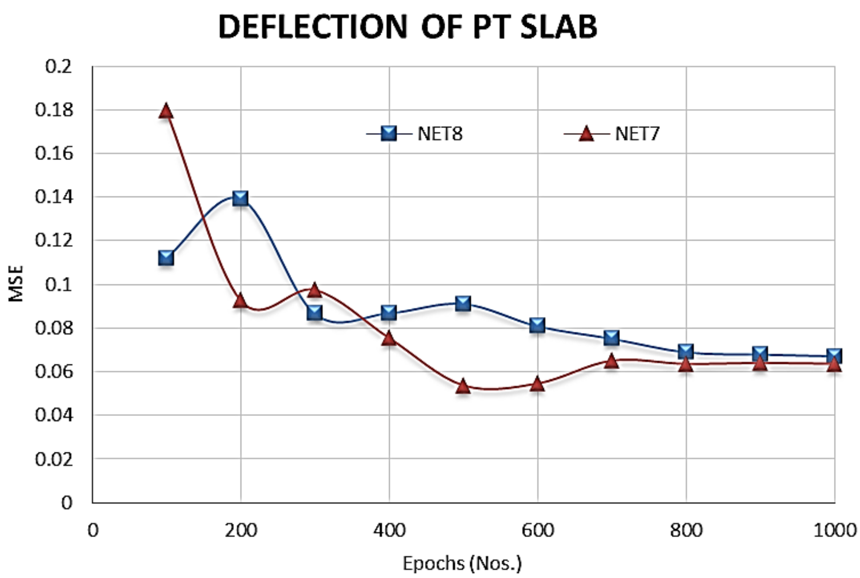


Fig 5.18: Optimized deflection models (2 nets)

In Fig 5.18, we have plotted network models NET3 and NET4 in the scale of 0 to 0.18 for more better comparision. In this range, although, the networks takes off in the range of 0.18 to 0.12, but eventually, both the networks tend to merge towards the mark of 0.07 from 600 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.07 at 1000 epochs.

Validation MSE for deflection of PT slab of network model, NET3: 1L_LM_Tan with fifteen neurons in the hidden layer, comes out to be 0.06353 at 1000 epochs. When the entire data undergoes testing the **MSE comes out to be 0.06197**.

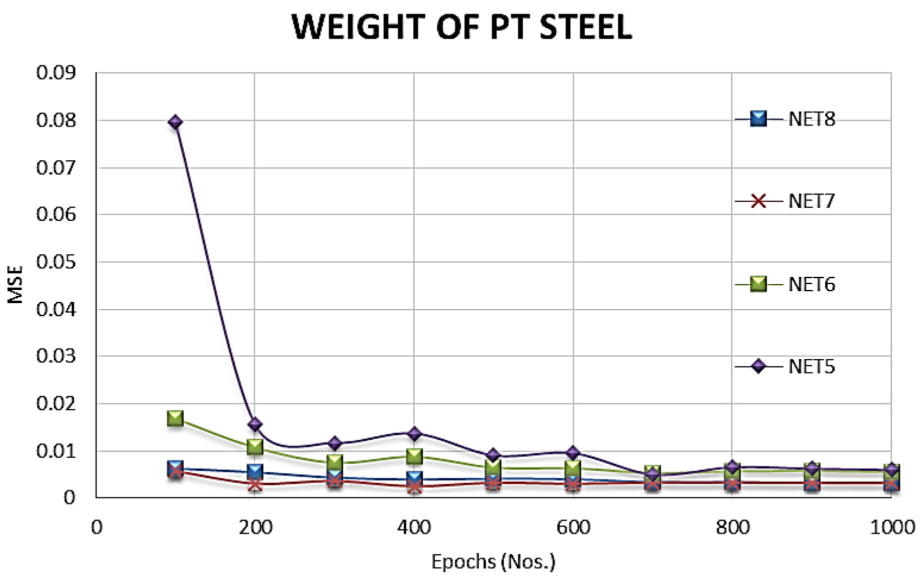


Fig 5.19: Weight models (4 nets)

Fig 5.19, shows the network models, NET1 to NET4, for comparision against the minimum MSE for weight of PT steel. Maximum number of epochs in NET1 and NET2 is 20 whereas in NET3 and NET3 it has been taken as 15. The variation in average MSE for weight of PT steel, of network models NET1 and NET2, lies in the range approximately under 0.01, when seen in 500 to 1000 epochs. Also the network models NET3 and NET4 shows average MSE under the range of 0.007 for all epochs. It can be seen again that the same network models, NET3 and NET4, proves to be better as compared to network models, NET1 and NET2.

In Fig 5.20, network models NET3 and NET4 are ploted in the scale of 0 to 0.007 for more better judgement. In this range, the networks takes off from 0.007, but eventually, both the networks tend to merge towards the mark of 0.0035 from 700 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.0035 at 1000 epochs.

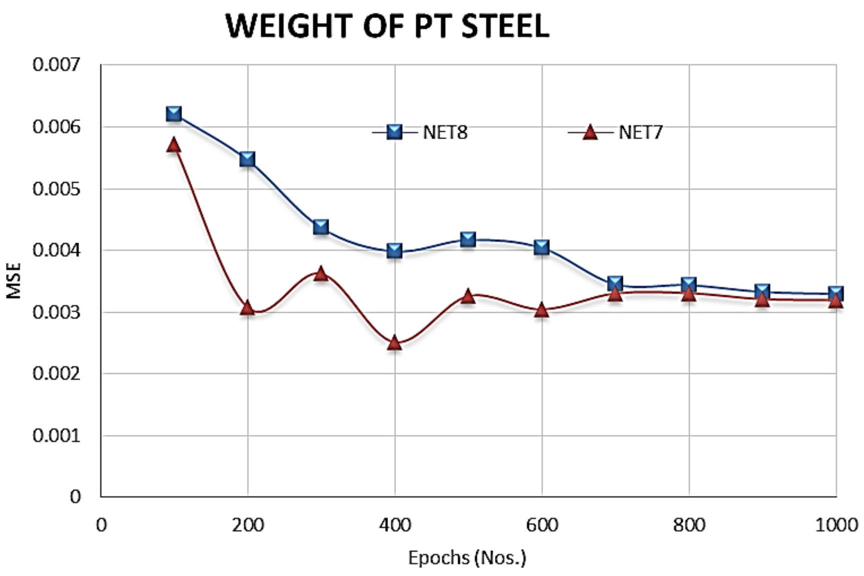


Fig 5.20: Optimized weight models (2 nets)

Training MSE for weight of PT steel of network model, NET4: 1L_LM_Log with fifteen neurons in the hidden layer, comes out to be 0.0031879 at 1000 epochs. When the entire data undergoes testing , the **MSE comes out to be 0.00201** which is reasonably good as compared to other network models for single layer.

Double layer networks:

Variation of deflection with the number of epochs is discussed here, when three way data split validation technique is employed. Each network is validated with different number of neurons, i.e., (5,5), (7,7) and (9,9) in first and second layers respectively. Mean Square Error (MSE) has been recorded at an interval of hundred epochs upto a maximum of 1000 epochs.

In Fig 5.21, eight network models, NET5 to NET12, have been compared. Maximum number of neurons in NET9 to NET12 are (9,9) whereas in NET5 to NET8 it has been taken as (7,7). The variation in average MSE for deflection of PT slab of network models NET9 to NET12 lies in the range below 0.25, when seen in 400 to 1000 epochs, whereas network models NET5 to NET8 shows average MSE under the range of 0.075 for 500 to 1000 epochs. Network models, NET5 to NET8, inspite of having less number of neurons, stands far better as compared to network models NET9 to NET12.

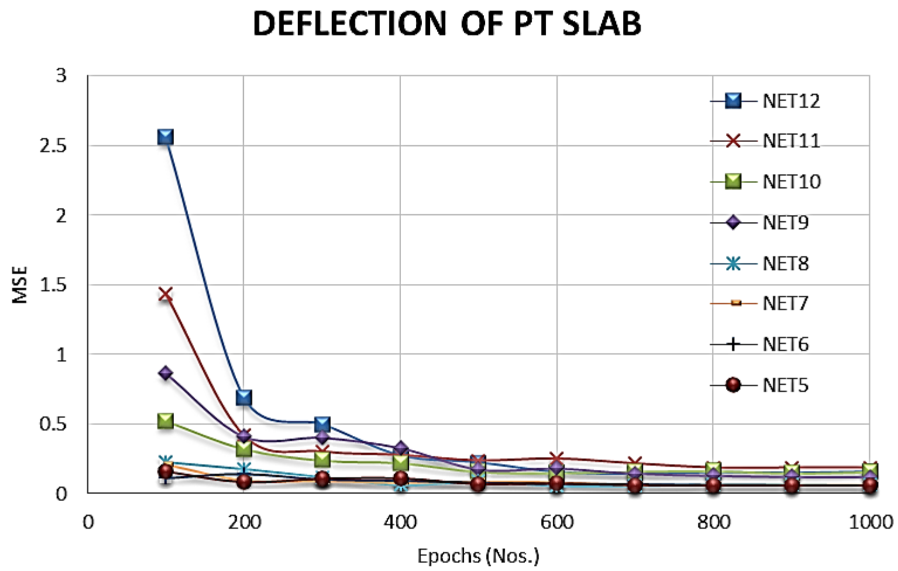


Fig 5.21: Deflection models (8 nets)

In Fig 5.22, network models NET5 to NET8 are ploted in the scale of 0 to 0.25, so as to select the optimized network. In this range, although, the networks takes off in the range of 0.225 to 0.125, but eventually, all the networks tend to merge towards the mark of 0.07 from 400 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.07 at 1000 epochs.

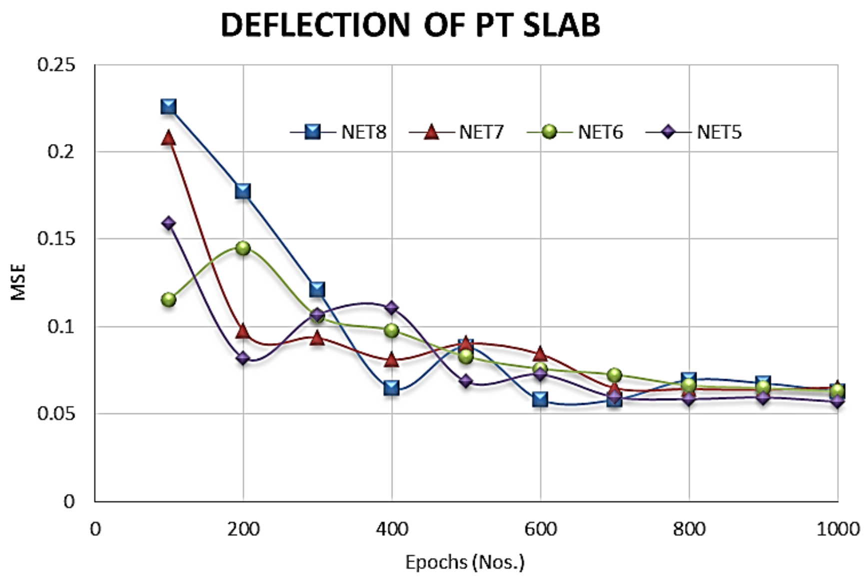


Fig 5.22: Optimized deflection models (4 nets)

Training MSE for deflection of PT slab of network model NET5: 2L_LM_Tan_Tan with seven neurons in each layer comes out to be 0.05679 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to be 0.05384** which is even less than the MSE for deflection of PT slab of single layer model, NET3: 1L_LM_Tan.

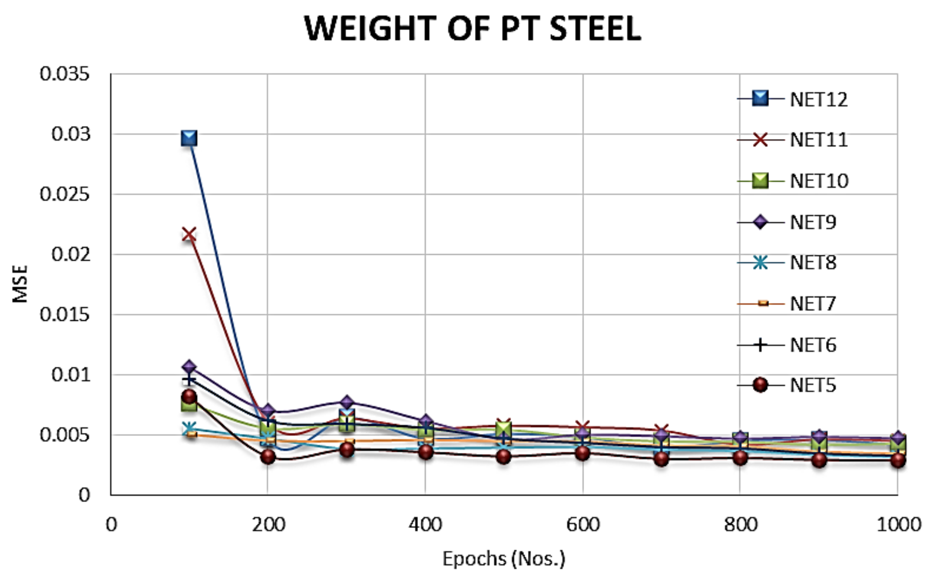


Fig 5.23: Weight models (8 nets)

In Fig 5.23, eight network models, NET5 to NET12, have been adopted. Maximum number of neurons in NET9 to NET12 are (9,9) whereas in NET5 to NET8 it has been taken as (7,7). The variation in average MSE for the weight of PT steel of network models NET9 to NET12 merges to 0.005, when seen in 400 to 1000 epochs, whereas network models NET5 to NET8 shows average MSE under the range of 0.004 for 500 to 1000 epochs. Network models, NET5 to NET8, prove to be better as compared to network models, NET9 to NET12.

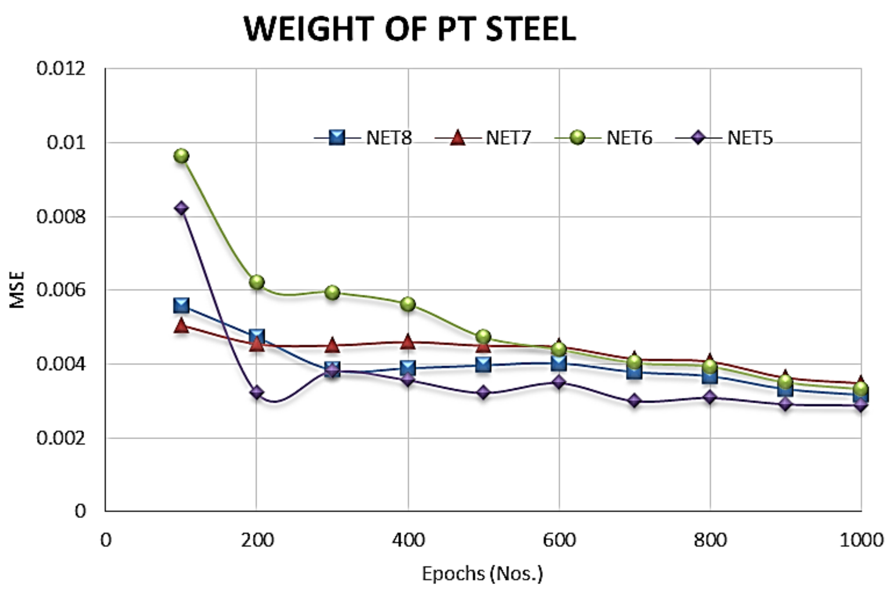


Fig 5.24: Optimized weight models (4 nets)

In Fig 5.24, network models NET5 to NET8 are ploted in the scale of 0 to 0.012. In this range, the networks takes off in the range of 0.01 to 0.005, but eventually, all the networks tend to merge towards the mark of 0.003 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.003 at 1000 epochs.

Training MSE of weight of PT steel of network model NET5: 2L_LM_Tan_Tan with seven neurons in each layer comes out to be 0.002897 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to be 0.00178** which is even less than MSE for weight of PT steel of single layer model, NET3: 1L_LM_Tan. The summary of results obtained by applying three-way data split validation tecnique are represented in Table 5.10 for deflection of PT-slabs and for weight of PT-steel.

Table 5.10: Summary of results of three-way data split validation technique

S.No.	**Network Model**	**No. of layers**	**No. of neurons**	**Training MSE at 1000 epochs**	**Testing MSE**
Deflection of PT slabs					
1.	NET3: 1L_LM_Tan	1	15	0.06353	0. 06197
2.	NET5: 2L_LM_Tan_Tan	2	7,7	0. 05679	0. 05384
Weight of PT steel					
3.	NET4: 1L_LM_Log	1	15	0. 0031879	0. 00201
4.	NET5: 2L_LM_Tan_Tan	2	7,7	0. 002897	0. 00178

The various selected network shows reasonably good results when compared with various other network models with single and double layers. Finally, looking at the results, it can be concluded that network model NET5: 2L_LM_Tan_Tan can be adopted as the optimized network model for both Deflection of PT slabs and Weight of PT steel in case of three way data split validation technique.

5.5 Networks for k-fold cross validation technique:

The single layer networks are shown in Table 5.11 and the double layer networks are shown in Table 5.12 for k-fold cross validation technique. Total four network architectures are developed for single layer networks and eight are developed for double layer networks.

Table 5.11: Single layer networks for k-fold cross validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
1.	NET1	1	20	Resilient	Tan sigmoid
2.	NET2	1	20	Resilient	Log sigmoid
3.	NET3	1	20	Levenberg Marquardt	Tan sigmoid
4.	NET4	1	20	Levenberg Marquardt	Log sigmoid

Table 5.12: Double layer networks for k-fold cross validation technique

S.No.	Network	Layers	Max. Hidden Nodes	Training Function	Transfer function
5.	NET5	2	9	Levenberg Marquardt	Tan sigmoid Tan sigmoid
6.	NET6	2	9	Levenberg Marquardt	Log sigmoid Log sigmoid
7.	NET7	2	9	Levenberg Marquardt	Tan sigmoid Log sigmoid
8.	NET8	2	9	Levenberg Marquardt	Log sigmoid Tan sigmoid
9.	NET9	2	9	Resilient	Tan sigmoid Tan sigmoid
10.	NET10	2	9	Resilient	Log sigmoid Log sigmoid
11.	NET11	2	9	Resilient	Tan sigmoid Log sigmoid
12.	NET12	2	9	Resilient	Log sigmoid Tan sigmoid

Single layer networks:

Variation of PT slab deflection and weight of PT steel, with the number of training epochs have been discussed here. Single layer networks have been created and k- fold cross validation technique have been employed for the selection of optimized neural network model. value of k has been taken as 10. Each network has been validated with different number of neurons, i.e., (5, 10, 15 and 20) in the hidden layer. It is seen that the networks are performing the best when there are 20 number of neurons in the hidden layer. Mean Square Error (MSE) has been recorded at an interval of hundred epochs upto a maximum of 1000 epochs.

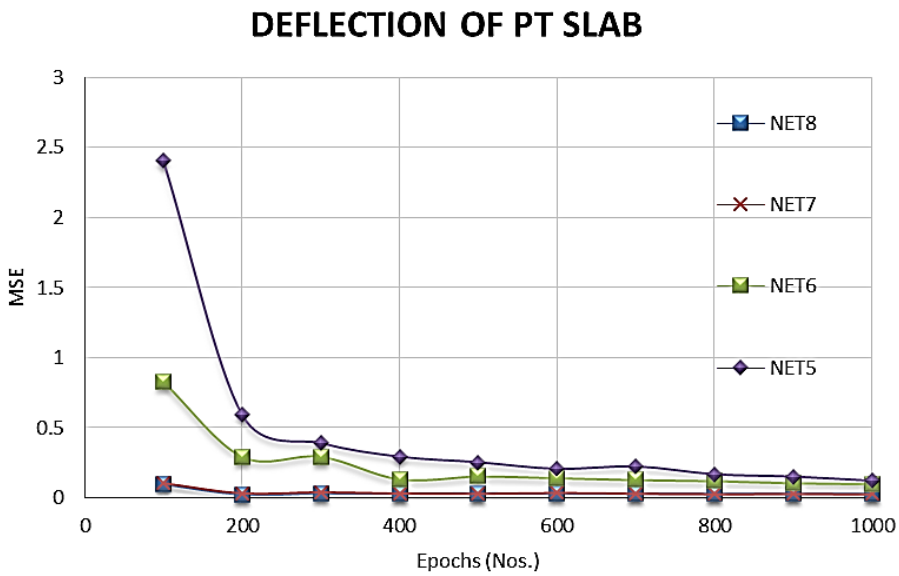


Fig 5.25: Deflection models (4 nets)

In Fig 5.25, four neural network models, NET1 to NET4, with single hidden layer have been compared to arrive at the network model giving the best result for the deflection of PT slab. The variation in average MSE for deflection, of network models NET1 and NET2, lies in the range approximately from 0.5 to 0.20, when seen in 400 to 1000 epochs. On the other hand network models NET3 and NET4 shows average MSE for deflection under the range of 0.1 for all epochs, which can be seen as a straight line. It is quiet evident from the graph that network models, NET3 and NET4, stands reasonably better as compared to network models, NET1 and NET2.

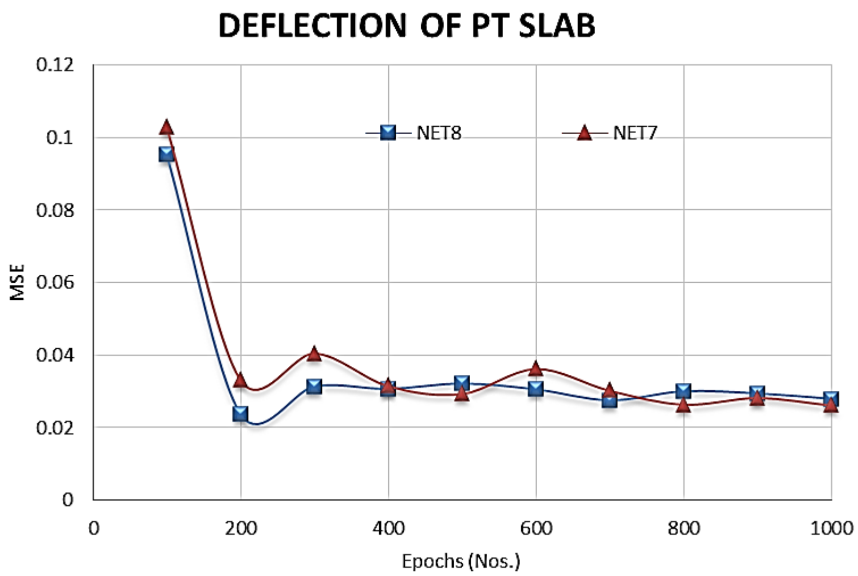


Fig 5.26: Optimized deflection models (2 nets)

In Fig 5.26, network models NET3 and NET4 are ploted in the scale of 0 to 0.1 for more better comparision with the other networks, so as to select the optimized network. In this range, although, the networks takes off in the range of 0.1, but eventually, both the networks tend to merge towards the mark of 0.03 from 200 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.03 at 1000 epochs.

Training MSE for deflection of PT slab of network model, NET3: 1L_LM_Tan with twenty neurons in the hidden layer, comes out to be 0.0261579 at 1000 epochs. When the entire data undergoes testing , the **MSE comes out to be 0.03010**.

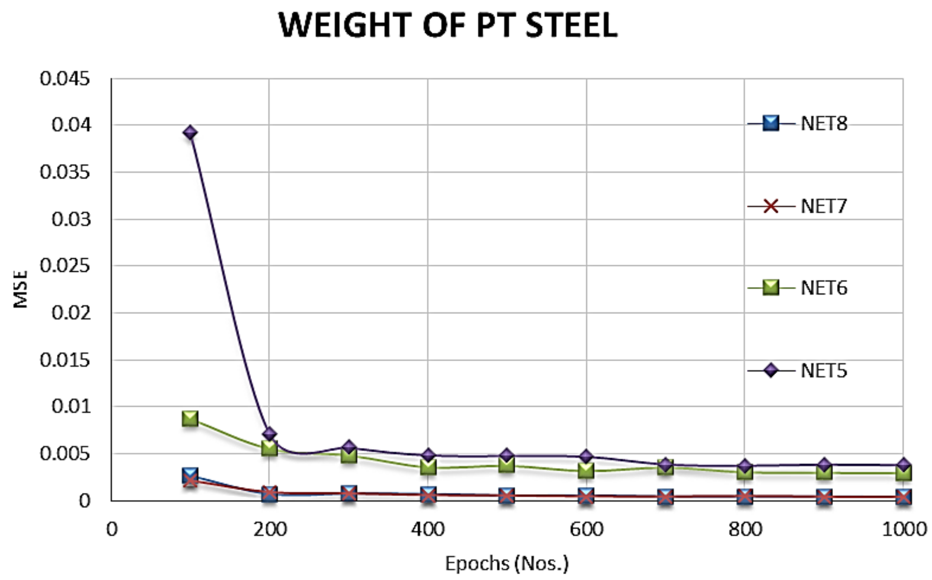


Fig 5.27: Weight models (4 nets)

Fig 5.27, shows the same network models as above, NET1 to NET4, for comparision against the minimum MSE for weight of PT steel. The variation in average MSE for weight of PT steel, of network models NET1 and NET2, lies in the range approximately from 0.005 to 0.003, when seen in 200 to 1000 epochs. Also the network models NET3 and NET4 shows average MSE for weight of PT steel under the range of 0.0024 for all epochs. It can be seen again that the same network models, NET3 and NET4, proves to be better as compared to network models, NET1 and NET2.

In Fig 5.28, network models NET3 and NET4 are ploted in the scale of 0 to 0.0025 for more better judgement. In this range, the networks takes off approximately from the range of 0.0025, but eventually, both the networks tend to merge towards the mark of 0.00035 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.00035 at 1000 epochs.

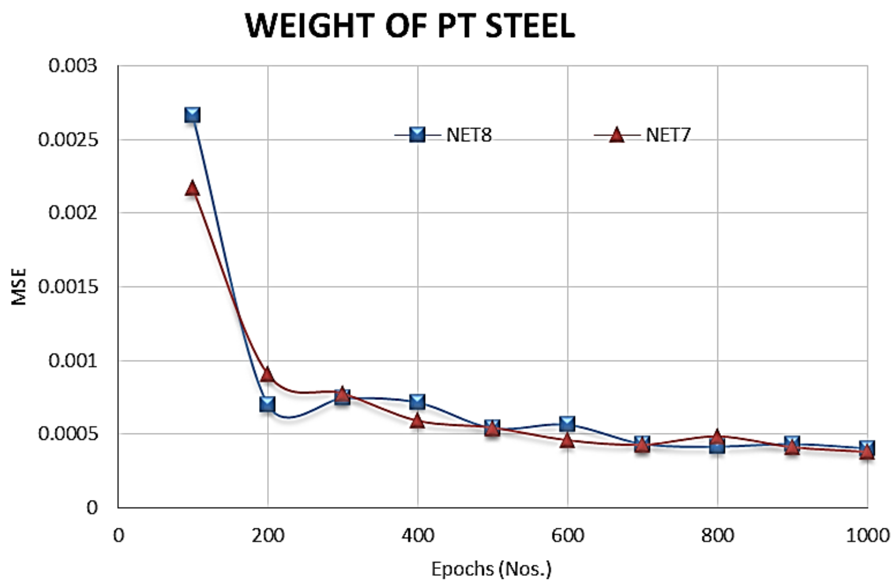


Fig 5.28: Optimized weight models (2 nets)

Training MSE for weight of PT steel of network model, NET3: 1L_LM_Tan with twenty neurons in the hidden layer, comes out to be 0.0003765 at 1000 epochs. When the entire data undergoes testing , the **MSE comes out to be 0.00044** which is reasonably good as compared to other network models for single layer.

Double layer networks:

Variation of deflection with the number of iterations or epochs by using k-fold cross validation technique have been discussed here which is employed for the selection of optimized neural network model. Each network have been validated with different number of neurons, i.e., (5,5), (7,7) and (9,9) in first and second layers respectively. It is seen that the networks are performing the best when the number of neurons in first and the second layers are (9,9) respectively. Mean Square Error (MSE) has been recorded at an interval of hundred epochs upto a maximum of 1000 epochs.

In Fig 5.29, eight such network models, NET5 to NET12, with double layers have been compared so as to reach out at the optimized network. The variation in average MSE for deflection of PT slab of network models NET9 to NET12 lies in the range approximately from 0.2 to a little below 0.1, when seen in 200 to 1000 epochs. At the same time, network models NET5 to NET8 shows average MSE under the range of 0.08 for all epochs, which can be seen as a straight line. It is quiet evident from the graph that network models, NET5 to NET8, stands reasonably better as compared to network models, NET9 to NET12.

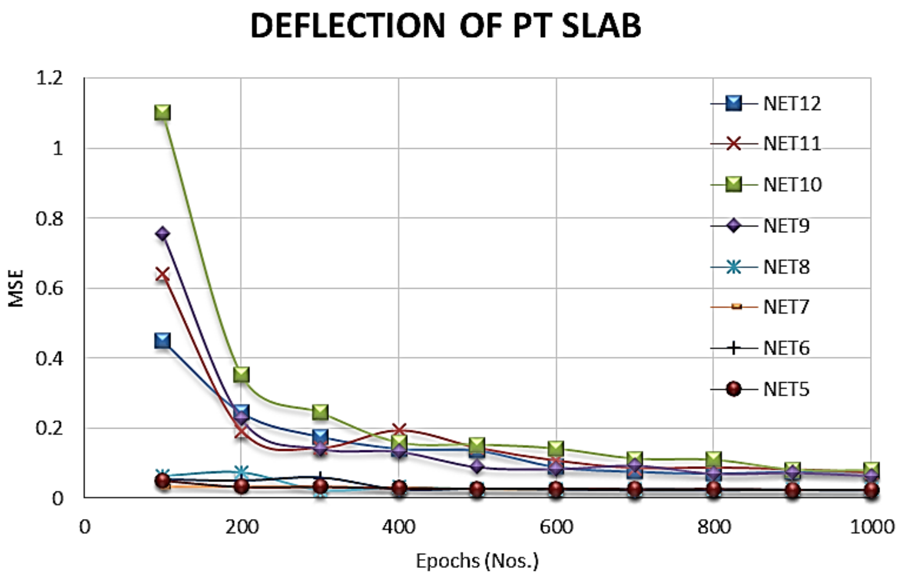


Fig 5.29: Deflection models (8 nets)

In Fig 5.30, network models NET5 to NET8 are ploted in the scale of 0 to 0.08 for more better comparision of the network models, so as to select the optimized network. In this range, although, the networks takes off in the range of 0.08 to 0.035, but eventually, all the networks tend to merge towards the mark of 0.02 from 400 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.02 at 1000 epochs.

Training MSE for deflection of PT slab of network model NET7: 2L_LM_Tan_Log with nine neurons in each layer comes out to be 0.0214092 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to be 0.02708** which is even less than the MSE for deflection of PT slab of single layer model, NET3: 1L_LM_Tan.

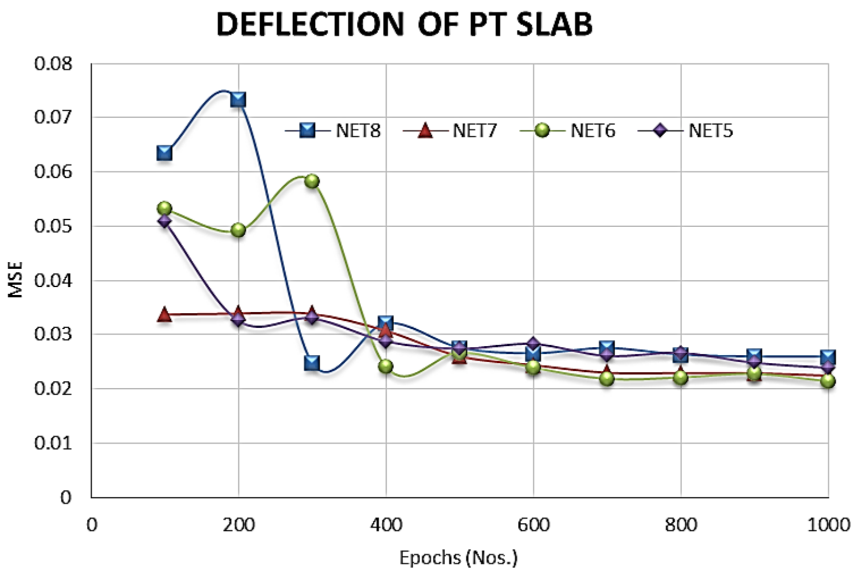


Fig 5.30: Optimized deflection models (4 nets)

In Fig 5.31, same eight network models, NET5 to NET12, have been adopted. The variation in average MSE for the weight of PT steel of network models NET9 to NET12 lies in the range approximately from 0.004 to 0.002, when seen in 300 to 1000 epochs, whereas network models NET5 to NET8 shows average MSE under the range of 0.001 for 200 to 1000 epochs, which can be seen as a straight line. It is quiet evident from the graph that network models, NET5 to NET8, are far better as compared to network models, NET9 to NET12.

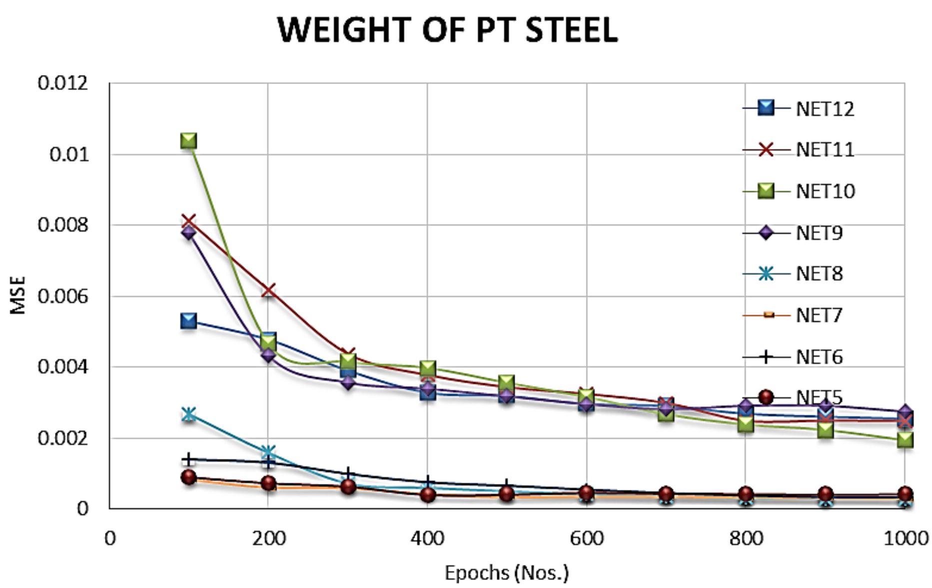


Fig 5.31: Weight models (8 nets)

In Fig 5.32, network models NET5 to NET8 are ploted in the scale of 0 to 0.003 for better selection of optimized network. In this range, the networks takes off in the range of 0.0025 to under 0.001, but eventually, all the networks tend to merge towards the mark of 0.0003 from 500 to 1000 epochs. It implies that the final network should be the one having training MSE very close to 0.0003 at 1000 epochs.

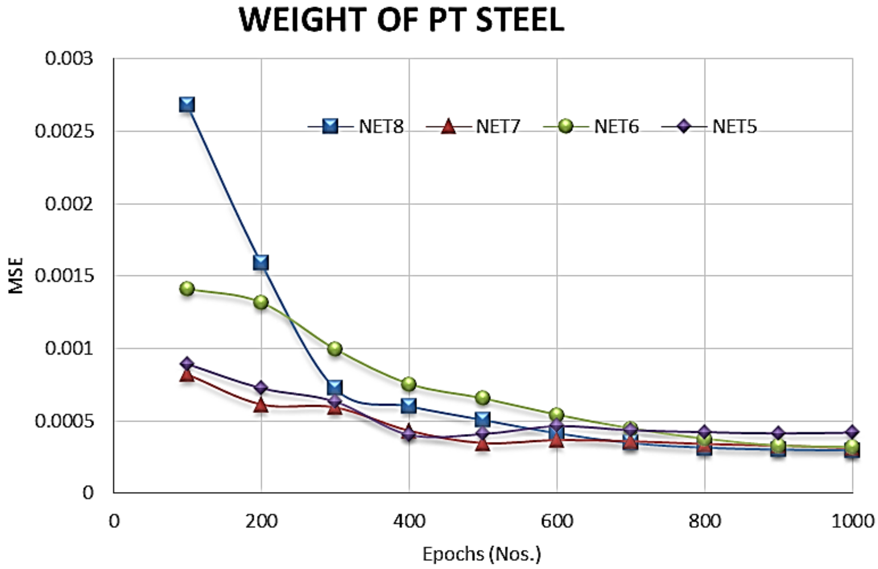


Fig 5.32: Optimized weight models (4 nets)

Training MSE for weight of PT steel of network model NET7: 2L_LM_Tan_Log with nine neurons in each layer comes out to be 0.0002983 at 1000 epochs. When the entire data undergoes testing , the **MSE of this model comes out to be 0.00040** which is even less than MSE for weight of PT steel of single layer model, NET3: 1L_LM_Tan. The summary of results obtained by applying k-fold cross validation tecnique is represented in Table 5.13 for deflection of PT-slabs and for weight of PT-steel.

Table 5.13: Summary of results of k-fold cross validation technique

S.No.	Network Model	No. of layers	No. of neurons	Training MSE at 1000 epochs	Testing MSE
Deflection of PT slabs					
1.	NET3: 1L_LM_Tan	1	20	0.0261579	0.03010
2.	NET7: 2L_LM_Tan_Log	2	9,9	0.0224037	0.02708

Weight of PT steel					
3.	NET3: 1L_LM_Tan	1	20	0.0003765	0.00044
4.	NET7: 2L_LM_ Tan_Log	2	9,9	0.000310	0.00040

The various selected network shows reasonably good results when compared with various other network models with single and double layers. Finally, from the results, it can be concluded that network model NET7: 2L_LM_Tan_Log can be adopted as the optimized network models for both, Deflection of PT slabs and Weight of PT steel in case of k-fold cross validation validation technique.

5.6 Analysis of results

In this chapter, the results obtained by various artificial neural network models, for deflection of PT slab and for the weight of PT steel required, have been analysed and discussed. Various validation techniques are applied to the developed neural networks and their performances have been evaluated by comparing the error rate in terms of Mean Squared Error (MSE). The MSE is a measure of how close a fitted line is to data points. The smaller the Mean Squared Error, the closer the fit is to the data. The summary of results for various validation techniques have been shown in Table 5.14 for deflection of PT slabs and in Table 5.15 for weight of PT steel.

Table 5.14: Summary of results for deflection of PT slab

Validation Technique	Network	Training MSE	Testing MSE
Resubstitution	NET7: 2L_LM_Tan_Log	0.0200615	0.01998
Holdout	NET7: 2L_LM_Tan_Log	**0. 0191037**	**0. 01599**
Three-way data split	NET5: 2L_LM_Tan_Tan	0. 05679	0. 05384
K-fold Cross validation	NET7: 2L_LM_Tan_Log	0.0224037	0.02708

Table 5.15: Summary of results for weight of PT steel

Validation Technique	Network	Training MSE	Testing MSE
Resubstitution	NET5: 2L_LM_Tan_Tan	0.0002679	0.00025
Holdout	NET6: 2L_LM_Log_Log	**0. 0002385**	**0. 00021**
Three-way data split	NET5: 2L_LM_Tan_Tan	0. 002897	0. 00178
K-fold Cross validation	NET7: 2L_LM_Tan_Log	0.000310	0.00040

The results shows that of all the validation techniques, maximum testing MSE value for deflection of PT slab corresponds to Three way data split (T) validation technique. The variation in the testing MSE value for T was found close to 63%, 70% and 50% when compared to the values obtained by Resubstitution (R), Holdout (H) and k-fold Cross validation (C) techniques respectively as shown in Fig.5.33. The variation in corrosponding value of training MSE was found close to 65%, 66% and 60% when compared to R, H and C validation techniques respectively. Here we can note that there is not much difference between the percentage variation of testing MSE and training MSE values for R and H validations. For C validation, there is a 10% variation between the results of training and testing MSE over the T validation results.

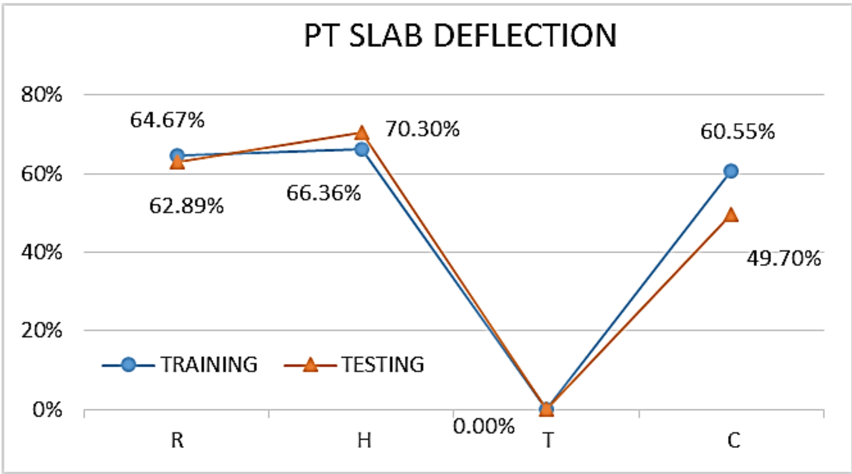


Fig 5.33: Relative comparison of MSE for PT slab deflection

Similarly, in case of the weight of PT steel, the results shows that again maximum testing MSE value corresponds to T validation of all the other validation techniques. The variation in the testing MSE value for T validation was found close to 91%, 92% and 89% when compared to the values obtained by R, H and C validation techniques respectively as shown in Fig.5.34. The variation in corrosponding value of training MSE was found close to 86%, 88% and 77% when compared to R, H and C validation techniques respectively. Here again the difference between the percentage variation of testing MSE and training MSE values for R and H validations is not much. For C validation, there is a 12% variation between the results of training and testing MSE over the T validation results.

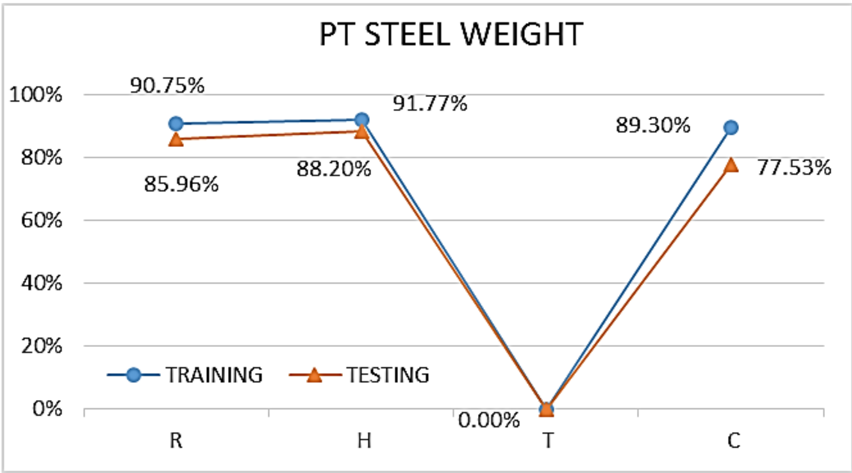


Fig 5.34: Relative comparison of MSE for PT steel weight

The above discussion indicates that the results of T validation are very pessimistic as compared to other validation techniques. The non performance of T validation can be attributed to two reasons. First, the networks have been trained with minimum data units. Secondly, for smaller data set, random sub sampling is to be performed (Reich and Barai 1999). For training of the networks, of the total data of 799 data units, 799 data units were provided for R, 534 for H, 267 for T and 720 for C validation. The architecture of ANN consisted of five input nodes and two output nodes. In these five input nodes, high variability have been introduced to form a database of various slab configurations. Now, since only a part (one third) data from the database was used for training in T validation, there is maximum possibility that, this one third part may not be able to absorb all the variability present in the database. Hence, such

ANNs will not perform well for the test data for which they have not been trained. This explains the higher value of training and testing MSE for T validation.

Next, from Table 5.14 and Table 5.15 it can be seen that the remaining three validation techniques viz. R, H and C are giving the MSE values for training and testing very close to each other. As such, comparision of these validation techniques on the basis of MSE alone cannot be justified. Hence we have performed an analysis for the error distribution and the graphs are plotted for the error values for all the validation techniques.

Graphs shown in Fig. 5.35 (a) and Fig. 5.35 (b) represents the training and testing error distribution for PT slab deflection as obtained from R validation technique. Here, x axis represents the number of data units and the corresponding values of deflection as obtained after training/testing the data, is represented on y axis. It is seen in Fig. 5.35 (a) that error is in the range of ±0.5 mm. Training error for some of the data points are exceeding upto ±1.0 mm also. Whereas error plot of R validation for training is showing a error distribution in somewhat a fixed range, the testing error plot for the same validation, Fig. 5.35 (b), is of diverging nature. Upto 300 data point, R validation is giving a highly optimistic testing results and beyond 300 data units the error is diverging upto 800 data points. This divergence is between ±0.3 to ±1.5mm. Although, the same data base is used for testing as used for training, the testing results were far away as expected. Analysis of data base have shown that the difference in testing errors are more for the the larger spans, spans of 11 and 12 meters. In larger spans deflection ranges between 30 to 40 mm and as such even a small variation of 1.5 mm appears very large as compared to deflections in spans less than 10 meters. In R validation, only those data are tested, which have undergone training by the same network and so any new data which the network have noot ‘seen’, cannot be tested. Hence R validation cannot be recommended as an appropriate validation technique.

In case of H validation, for training error distribution, as shown in fig. 5.35 (c), it can be seen that the errors are well within the limits of ±0.3mm for maximum data units used for training the network and a very few errors are coming upto a mark of ±0.5mm. As far as testing results are concerned, again their distribution is highly directional as shown in fig. 5.34 (d), although a very few error are approaching the range

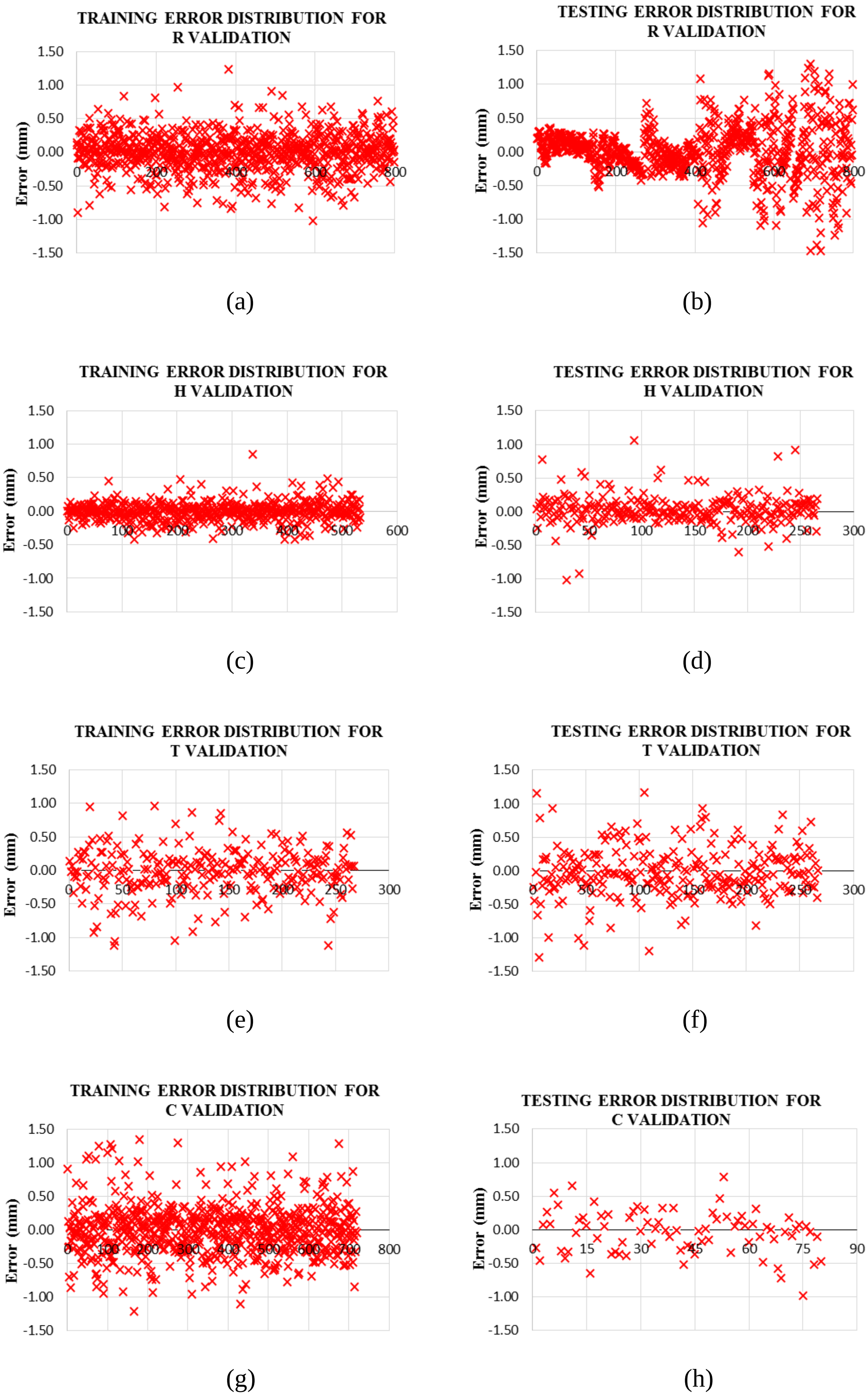


Fig 5.35: Comparison of validation techniques for PT slab deflection

between +0.5 to +1.0mm. The network undergone H validation, was trained using two-third (67%) data units and hence it is expected that this network has absorbed reasonable variations of PT slab configurations. Hence when this network is tested with remaining one-third (33%) data units, the optimistic results were obtained. Therefore, H validation can be considered as one of the optimistic validation technique.

For T validation, as shown in fig 5.35 (e), there is a high variation in the training error distribution. Error for maximum data points is in the range between +0.5mm and reaching upto a level of +1.0mm. Such high variation of training error implies that the network training was insufficient. The data units used for training the network were only one-third of the data base and as such the network didn't experienced the entire variation of the PT slab database. In can be seen in fig 5.35 (f), the results for testing error distribution are as expected, very high and dispersed, attaining a range between +1.0 to +1.5mm. As such this technique is giving very pessimistic results and not recommended as a decision support system.

Error distribution for training outcomes in case of C validation is represented in fig 5.35(g). Here the network has been trained using 719 data units (90% of the database) and tested with remaining 80 data units (10% of the database). Although, majourity of the training error are well within the range of +0.5mm, at the same time considerable number of training error are attaining values between +0.5mm to +1.5mm. Here, even though the network was trained with sufficient data units, still the results are not fully conforming to what was expected. The testing error distribution as shown in fig 5.35(h) indicates that almost all the testing data is in the range of +0.5mm except a very few reaching +1.0mm error mark. Performing validation of a network with too few data units may not give the true picture of the network performance as these data units may not consist of all sort of database variation. Hence, this validation is not recommended for the present case.

The results for the deflection of PT slab as obtained from R, H, T and C validation techniques have been discussed above in detail. The discussion reveals the fact that H validation is the only technique having sufficient dataunits for training the network as well as for its testing. Secondly the range of training and testing errors is minimum as compared to other validation techniques. Thirdly, also MSE value for H validation comes out to be a minimum. Hence, for prediction of deflection of PT slabs, H validation can be recommended over other validation techniques.

Graphical representation of error distribution is shown in Fig. 5.36 (a) and Fig. 5.36 (b) for the training and testing errors for weight of PT steel as obtained from R validation technique. Here, x axis represents the number of data units and the y axis represents the corresponding values of weight of PT steel per square meter as obtained after training/testing the data. It can be seen from Fig. 5.36 (a) that training error is in the range of $\pm$0.1 kg/m^2 for maximum number of data units and very few of these are exceeding the of $\pm$0.15 kg/m^2.

The testing error plot for the same validation, Fig. 5.36 (b), shows the formation of localized clusters. Also the distribution of error is not uniform and follows a dumbbell type distribution. The main divergence in error is in the range of 0 to 200 dataunits and then between 600 to 800 dataunits. The dumbbell shape distribution is just due to the the sequence of data units as they are randamized. The cluster formation may be due to same value of weight for several configurations of the PT slab. However, this shows that the network was not able to adapt the experience from the data base provided. The main thing here, is the testing error which is ranging from $\pm$0.05 kg/m^2 for maximum number of data units and very few of these are exceeding the of $\pm$0.15 kg/m^2. Although the error is not very much, but since the distribution is not uniform, R validation cannot be recommended as a decision support system for determination of weight of PT steel.

Training error distribution as obtained by H validation is shown in fig. 5.36 (c) for the weight of PT slabs. The distribution shows a very good training of the network as the maximum number of error lies between $\pm$0.03 kg/m^2 with very few reaching up to a $\pm$0.05 kg/m^2 level mark. As far as testing results are concerned, again their distribution is highly directional as shown in fig. 5.36 (d). The distribution of testing error shows the error range below $\pm$0.03 kg/m^2 for maximum number of dataunits and a quite few in the over this range. The error distribution for both training and testing results are showing very optimistic results. The error range is very small as compared to other errors obtained by other validation techniques. As discussed previously, in H validation, number of dataunits for training (two-third) as well as for testing (one-third) are sufficient and hence the networks are well trained by this technique. As such, the

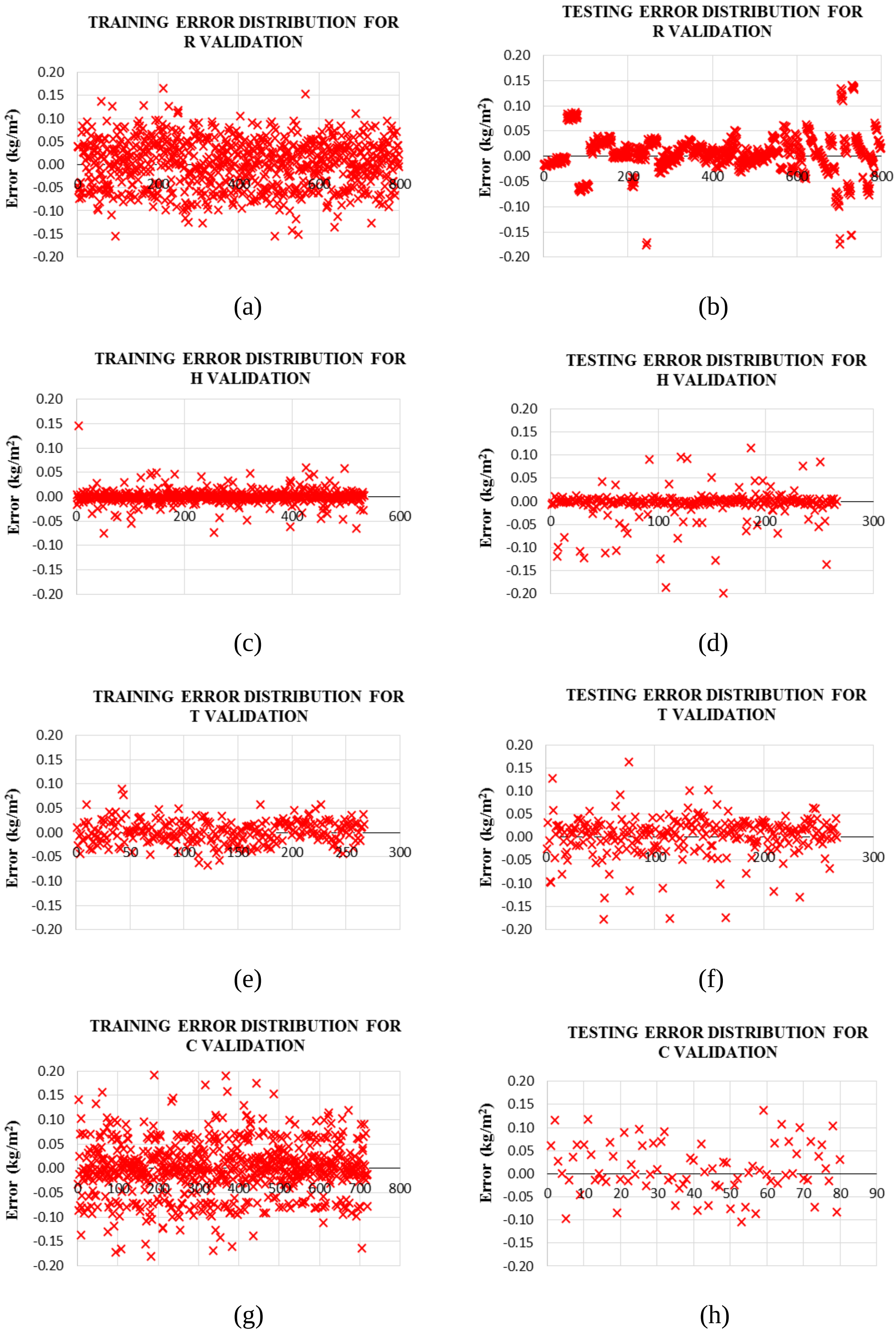


Fig 5.36: Comparison of validation techniques for PT steel weight

optimistic results can be expected. Owing to highly directional nature and minimum error range, H validation can be recommended as a technique for determining the weight of PT steel and an excellent support system.

For T validation, as shown in fig 5.36 (e), there is a uniform and controlled variation in the training error distribution. Error for maximum data points is in the range between $\pm$0.05 kg/m^2 and quiet a few reaching upto a level of $\pm$0.10 kg/m^2. Such distribution of training error implies that the network has very well adapted the variations in the database. In can be seen in fig 5.36 (f), the results for testing error distribution are very high and dispersed, attaining a range between $\pm$1.0 to $\pm$1.5 kg/m^2. Since the database for training and testing is only one-third of the entire database, a high variation is expected in both cases respectively. As such this technique cannot be recommended as a decision support system.

Fig 5.36(g) shows the training error distribution for C validation for the weight of PT steel. The figure represents a very high variation in the error distribution, conforming with Fig 5.36(a) of R validation for training error distribution. Here, although the network have been trained with large number of data units (90% of the database), still the results are not following the expectations. The errors are mainly concentrated in the range of $\pm$0.1 kg/m^2 and some are reaching to a maximum extent of $\pm$0.2 kg/m^2 also. This shows a broad spectrum of error distribution which may be because, in large database, the variations are also large and the network may not be able to absorb the variations in the database efficiently. The testing error distribution shown in fig 5.36 (h) represents the high variabilty in predicted errors by C validation. The range of error is between $\pm$0.1 kg/m^2 and some errors are touching a level of 0.15 kg/m^2 also. Only 10% database has been used for testing the network performance and which may not present the actual performance of the network. Hence C validation is not recommended for the present case.

The results for the weight of PT steel as obtained from R, H, T and C validation techniques have been discussed above in detail. The discussion reveals the fact that H validation is the only technique having sufficient dataunits for training the network as well as for its testing. Secondly the range of training and testing errors is minimum as compared to other validation techniques. Thirdly, also MSE value for H validation comes out to be a minimum. Hence, for prediction of weight of PT slabs, H validation can be recommended over other validation techniques.

Finally, it can be concluded that:

1) Network NET7, with 9 neurons in each layer, can be adopted as the optimized network for deflection of PT slab in case of Holdout validation technique.

2) Network NET6, with 9 neurons in each layer, can be adopted as the optimized network for weight of PT steel in case of Holdout validation technique.

5.7 Summary

In the present research work, both single and double layered neural networks have been employed and validated for evaluating their performance. This section deals with the application of developed ANN modals to the problem of determination of deflection of PT slabs and to the weight of post tesioning steel in PT slabs. Performance of ANNs have been represented graphically by forming a plot between number of epochs (taken on X axis) and the MSE (taken on Y axis) attained. The maximum number of epochs are taken as 1000 and the MSE is determined at an increment of every 100 epochs. The number of hidden neurons for single layered networks have been taken as 20 and in case of double layered networks it is taken as 9 in each of the two layers. The networks have been validated using, four validation techniques namely resubstitution, holdout, three way data split and kfold cross validation. A detailed discussion on the results obtained by employing these validtaion techniques have been made in this section. This discussion supports the H validation technique for predicting the deflecion of PT slabs and weight of PT steel.

Chapter – 6

Benchmarking of Artificial Neural Networks

6.1 Introduction

Benchmarking can be defined as a process of checking and evaluating certain phenomena with some standard technique. This is a method in which the best performing methodology can be identified. Also the degree of effectiveness of the applied technique can be evaluated.

The basic steps of benchmarking may be as follows:

1. Determine what to benchmark.
    - Here, largely the area to be benchmarked is to be identified.

2. Form a group.
    - Once the area to be benchmarked is identified, some specific data is collected for testing.

3. Identifying top performers.
    - This data is analyzed and compared with the standard data and hence the best performing technique is identified.

4. Implementation of the chosen technique.
    - The identified technique is implemented to the available data sets.

The main objective of this research work is to find the most suitable technique at conceptual stage of design, using which the deflection and requirement of Post Tensioning steel in Post Tensioned slabs can be predicted. This is done by approximating a function indication the relation between the input and the output parameters by using Artificial Neural Networks. Finally to prove that the results obtained by using Artificial Neural Networks are much precise than any other statistical technique and can be conveniently applied to problems related structural designing, study of statistical methods of function approximation is a must.

6.2 Regression Analysis

Regression analysis is a technique which is widely used for prediction and forecasting. It is a computational tool which determines the relationship between dependent and independent variables. The use of one or more than one variable to determine the expected value of one's own variable is known as regression. In this technique we have one or more independent variable also known as explanatory variable and a dependent variable which is also known as response variable. Explanatory variable gives a logical relationship with a problem for which a response will be generated. In this manner explanatory variable is used for modeling of response variable through a linear relationship. If there are 'n' numbers of independent variables which are mathematically related to a dependent variable, then this mathematical relationship is termed as regression analysis. When the data base is provided to the regression model, it develops a quantitative function between the explanatory and the response variable. It also shows that how the dependent variables respond to the independent variables. This methodology is much suited for the processes where the results are not precise but vary in a particular range. Observations of almost all the experimental programs vary to some extent and hence can be easily accessed using this technique. In such experimental programs which are even discontinuous at certain point or where the result values vary widely, regression techniques can be applied appreciably.

6.2.1 Regression Model Building

Every experiment in a research work consists of a large database of several independent variables. Various subsets of these variables determine the efficiency of the regression model. Hence, if the subset of independent variables is the good predictor, the regression model will be more precise. So, it is quiet evident that a methodology must be selected for the determination of such a subset of independent variables which gives acceptable results. Best subset is one having such independent variables which directly relates to or which some weight towards the characteristic property of the dependent variable. Regression determines the curve which fits the given independent variables in such a way that the sum of square of the vertical distance from the curve to the independent variable is the least.

6.2.2 Linear Regression Models

Linear regression model is the simplest model amongst all the other regression models. This model shows the dependency of dependent variable 'y' on the subset of independent variables (x). The mathematical expression for this is given in eq-6.1:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \qquad \text{------------ 6.1}$$

The equation above represents a two dimensional plane, as a linear expression with two regressors also called independent variables or predictors. Here, β_0 is the absolute value of the intersection of the plane, β_1 is the change in the value of y per unit change in value of x_1 keeping x_2 constant and β_2 is the change in the value of y per unit change in value of x_2 keeping x_1 constant.

Number of regressors is not limited and these can vary for some finite value. The dependent variable y is related to k independent variables as shown in eq-6.2.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots\ldots\ldots + \beta_k x_k + \varepsilon \qquad \text{----------- 6.2}$$

The equation 6.2 shown above is the modified form of equation 5.1 with k number of independent variables and hence, referred as a multiple linear regression model.

6.2.3 Linear Regression Model, Parameters and their Estimation

Regression coefficients are estimated using the method of least squares in models of multiple linear regressions. Model expression with dependent variables y_1, y_2, y_3,………,y_n can be represented as shown in eq-6.3:

$$y_i = \beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\ldots\ldots\ldots\ldots\ldots\ldots+\beta_k x_{ik} + \varepsilon_i$$

$$= \beta_0 + \Sigma\, \beta_j\, x_{ij} + \varepsilon_i\ ; \quad i=1, 2, 3, 4\ldots\ldots\ldots\ldots, n \qquad \text{----------- 6.3}$$

where, y_i, is the response variable at i^{th} level and x_{ij} is the j^{th} regression variable at variable x_i.

The regression coefficients are chosen by the least square methodology in such a way so that the sum of squares of the error (ε_i) comes out to be a minimum.

The least square function may be represented as shown below and gives the least value with respect to $\beta_0, \beta_1, \ldots\ldots\ldots\ldots\ldots \beta_k$. as shown in eq-6.4

$$L = \Sigma\, \varepsilon_i{}^2$$

$$= \Sigma\, (y_i - \beta_0 - \Sigma\, \beta_j\, x_{ij})^2 \qquad \text{----------- 6.4}$$

The regression model equation can also be represented in form of a matrix structure as shown in eq-6.5

$$y = X\beta + \varepsilon \qquad \text{----------- 6.5}$$

where,

$$y = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{bmatrix}; \quad X = \begin{pmatrix} 1 & x_{11} & x_{12} . & . & x_{1k} \\ 1 & x_{21} & x_{22} . & . & x_{2k} \\ . & . & . \; . & . & . \\ . & . & . \; . & . & . \\ . & . & . \; . & . & . \\ 1 & x_{n1} & x_{n2} . & . & x_{nk} \end{pmatrix}$$

$$\beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ . \\ . \\ . \\ \beta_n \end{bmatrix} \quad \text{and} \quad \varepsilon = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ . \\ . \\ . \\ \varepsilon_n \end{bmatrix}$$

Now, for evaluating the least square function, expression given in eq-6.6 is used:

$$L = \sum_{i=1}^{n} e_i^2 = \varepsilon'\varepsilon = (y - X\beta)'(y - X\beta) \qquad \text{----------- 6.6}$$

The solution for β in eq-6.6 is given by the least square estimator which is represented as $\hat{\beta}$ in the eq-6.7.

$$\frac{\partial L}{\partial \beta} = 0 \qquad \text{----------- } 6.7$$

Finally, the expression obtained by differentiating the least square function with respect to β is expressed as shown in eq-6.8 and it represents the matrix form of normal equation of least square.

$$X'X\hat{\beta} = X'y \qquad \text{----------- } 6.8$$

In order to solve this equation, both side of this equation is multiplied by the inverse of $(X'X)$. Hence the least square estimator $\hat{\beta}$ is given by the eq-6.9:

$$\hat{\beta} = (X'X)^{-1}X'\,y \qquad \text{----------- } 6.9$$

The matrix representation of the resultant equation is as shown below:

$$\begin{pmatrix} n & \sum_{i=1}^{n} x_{i1} & \cdot\ \cdot\ \ \cdot & \sum_{i=1}^{n} x_{ik} \\ \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i1}^{2} & \cdot\ \cdot\ \ \cdot & \sum_{i=1}^{n} x_{i1}x_{ik} \\ \cdot & \cdot & \cdot\ \cdot\ \ \cdot & \cdot \\ \cdot & \cdot & \cdot\ \cdot\ \ \cdot & \cdot \\ \cdot & \cdot & \cdot\ \cdot\ \ \cdot & \cdot \\ \sum_{i=1}^{n} x_{ik} & \sum_{i=1}^{n} x_{ik}x_{i1} & \cdot\ \cdot\ \ \cdot & \sum_{i=1}^{n} x_{ik}^{2} \end{pmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \\ \\ \hat{\beta}_k \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} x_{i1} \\ \sum_{i=1}^{n} x_{i1}y_i \\ \cdot \\ \cdot \\ \\ \sum_{i=1}^{n} x_{ik}y_i \end{bmatrix}$$

Hence the regression model which can fit this expression is given by eq-6.10:

$$\hat{y} = X\hat{\beta} \qquad \text{----------- } 6.10$$

And the same model with scalar expression is given by eq-6.11:

$$\hat{y}_X = \hat{\beta}_0 + \sum_{i=1}^{k} \hat{\beta}_j x_{ij} - \qquad \text{---------- } 6.11$$

where, $i = 1, 2, \ldots\ldots, n$

Now, the residual error, that is the difference between the values obtained from actual observation y_i and the corresponding values obtained from the fitted curve i.e., $\hat{y}_i$ is given by eq-6.12:

$$e_i = y_i - \hat{y}_i \qquad \text{----------- } 6.12$$

6.2.4 Properties of Estimators

In the methodology adopted of least squares, it is seen that the expected value of $\hat{\beta}$ is none other than β itself. An unbiased estimator is produced in least square method for linear regression model and it can be shown as expressed in eq-6.13:

$$E(\hat{\beta}) = \beta \qquad \text{----------- } 6.13$$

Thus, we can say that $\hat{\beta}$ is an unbiased estimator of β.

The variances of unbiased estimator $\hat{\beta}$ are expressed in terms of the inverse of $(X'X)$ matrix. The covariance matrix of the regression coefficients $(\hat{\beta})$ is given by the variance σ^2 multiplied by the inverse of $(X'X)$.

The variances of $\hat{\beta}_0, \hat{\beta}_1, \ldots\ldots, \hat{\beta}_k$ are the diagonal element of $\sigma^2 \, (X'X)^{-1}$. Also the off-diagonal elements of this matrix represent the covariance as shown in eq 6.14.

$(\hat{\sigma}^2$ = variance of error term)

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{(n-p)} = \frac{SS_E}{(n-p)} \qquad \text{----------- 6.14}$$

6.2.5 Polynomial Regression Models

One independent variable may also depend on the characteristic of any other independent variable in order to obtain the dependent variable. In such cases the linear model, $y = x\beta + \varepsilon$ may not give the best fitting curve and hence a polynomial regression models are to be used. For example, the second order polynomial in one variable may be given by eq-6.15:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 \qquad \text{----------- 6.15}$$

Also the second degree polynomial in two variables is given by eq-6.16:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon \qquad \text{----------- 6.16}$$

In the polynomial regression model with two variables, all the possible combinations of the variables are made and hence the error is measured. This can be seen in the expression shown above.

6.3 Correlation Analysis

Once the relationship is known amongst the variables, correlation analysis is performed on the variables. Correlation analysis is a statistical method of determining the closeness of the actual value of the variables and the values of variables obtained from the relationship established.

Goodness of relationship between the variables is measured in terms of correlation coefficient. When there is not much difference between the values of variables, the correlation coefficient is large and when the values have a considerable difference the correlation coefficient is small. The range of correlation coefficient is between +1 to -1. When the relationship between the variables is perfectly linear and positive, i.e. increasing, correlation coefficient is +1. On the other hand if the relationship between the variables is perfectly linear but negative, i.e. decreasing, correlation coefficient is -1.

In order to evaluate the correlation coefficient, the total corrected sum of squares (SS_{TC}), the sum of the squares due to regression (SS_R) and the sum of squares due to error (SS_E) is to be determined. The above expressions for sum of squares may be related as expressed in eq-6.17:

$$SS_{TC} = SS_E + SS_R \qquad \text{----------- } 6.17$$

The condition for the regression curve to fit perfectly with the database is when $SS_E = 0$ or when $SS_R = SS_{TC}$. In this condition the value of correlation coefficient will be 1. On the other hand, there will be zero correlation, if $SS_R = 0$. So correlation coefficient is given in terms of the sum of squares.

The eq-6.18 gives the desired expression:

$$R^2 = \frac{SS_R}{SS_{TC}} = \frac{SS_{TC} - SS_E}{SS_{TC}} = 1 - SS_E/SS_{TC} \qquad \text{---------- } 6.18$$

When a new variable is added to the regression model, it is seen that the R^2 value always increases. Hence correlation analysis cannot be used as a measure of assurance for the goodness of new variable added.

In this case, adjusted R^2 statistic may be used as represented in eq-6.19:

$$R^2_{adj} = \frac{SS_E/(n-p)}{SS_{TC}/(n-1)} \qquad \text{----------- } 6.19$$

where, $SS_E/(n\text{-}p)$ is the residual mean square error and $SS_{TC}/(n\text{-}1)$ is a constant,

Now, when a variable is added to the regression model, R^2_{adj} will increase only when this added variable reduces the mean square error.

6.3.1 Regression Model Building and Parameter Estimation

The entire database consisting of 799 data units was divided into two parts. Out of these, one part with two-third data (533 data units) were used for building regression model and the remaining one-third data (266 data units) were used for testing the goodness of fit of the regression model.

Both, linear regression model and polynomial regression model are used for the evaluation of the model.

6.3.1.1 Linear Regression Model

There are five independent variables viz:

- Span of PT slab,
- Depth/thickness of PT slab,
- Live load intensity,
- Column size (square cross section), and
- Grade of concrete

Corresponding to the above mentioned input parameters, there are two outputs for the various configuration of the slab. Hence, the expression for multiple linear regression model is given as been shown in eq-6.20:

$$y = A + Bx_1 + Cx_2 + Dx_3 + Ex_4 + Fx_5 \quad \text{----------- 6.20}$$

Where, y = dependent or the response variable,

A, B, C, D, E and F = regression parameters and

x_1, x_2, x_3, x_4 and x_5 are independent variables corresponding to span, depth, live load intensity, column size and grade of concrete respectively.

The regression analysis is performed using standard statistical software package. Regression analysis is performed taking 75 percent of the data set values, i.e. on 600 data units, for deflection and weight of post tensioning steel of the PT slab. The regression parameters were evaluated from the standard software. The Table 6.1 indicates the values of regression parameters and coefficient of determination (R^2) as follows:

Table 6.1: Regression parameters for Linear Regression

Parameters	Deflection	Weight
A	-7.698	0.687
B	5.453	0.164
C	-0.150	0.007
D	0.756	0.061
E	0.000	0.000
F	-0.089	-0.006
R^2	0.921	0.307

Here,

Deflection = deflection for the PT slab

Weight = weight for PT steel

6.3.1.2 Polynomial Regression Model

As been already stated previously, the effect of characteristics of one variable influences the other variable; hence the interaction between the variables is to be considered in case of polynomial regression model. The expression for the same may be expressed as shown in eq-6.21:

$$y = A+Bx_1 + Cx_2 + Dx_3 +Ex_4 +Fx_5 +Gx_1^2 + Hx_2^2 + Ix_3^2 + Jx_4^2 + Kx_5^2 + Lx_1x_2 + Mx_1x_3 + N x_1x_4 + Ox_1x_5 + Px_2x_3 + Qx_2x_4 + Rx_2x_5 + Sx_3x_4 + Tx_3x_5 +Ux_4x_5 \text{----------} \quad 6.21$$

Where, A, B, C, D, ………… and U = Regression parameters.

The Table 6.2 indicates the values of regression parameters and coefficient of determination (R^2) for polynomial regression as follows:

Table 6.2: Regression parameters for Polynomial Regression

Parameter	Deflection	Weight
A	-7.378	0.913
B	2.501	-0.225
C	-0.080	0.026
D	-0.049	0.247
E	-3.974E-05	0.002
F	0.212	-0.090
G	0.725	0.112
H	0.001	0.000
I	-0.041	-0.007
J	-1.890E-06	2.906E-08
K	-0.001	0.002
L	-0.048	-0.007
M	0.220	0.054
N	0.000	0.000
O	-0.047	-0.007
P	-0.005	-0.002
Q	-1.561E-05	4.146E-06
R	.001	0.000
S	6.706E-05	-7.169E-05
T	0.002	-0.003
U	4.052E-05	-1.598E-05
R^2	0.996	0.815

Here,

Deflection = deflection for the PT slab

Weight = weight for PT steel

6.4 Comparison of Outputs of ANN, Linear Regression and Polynomial Regression

Here, the results obtained from linear multiple regression models and polynomial multiple regression models have been compared with the results of artificial neural networks and standard experimental values. By this comparison, best suited technique for the evaluation of stated problem will be chosen. For the regression equation building, 75% data of the total database (i.e. 600 data units) have been utilized.

The basis of statistical analysis is the values of mean, standard deviation and paired t-test for the outputs obtained through ANN, Linear Regression and polynomial Regression. The results obtained from ANNs outputs with linear regression outputs and ANNs outputs with polynomial regression outputs for the deflection of PT slabs are tabulated in Table 6.3.

Table 6.3: Outputs for Artificial Neural Network (ANN), Linear and Polynomial Regression models for Deflection of Post Tensioned slabs.

Description	Detailed Analysis Output	ANN (Holdout) Output (ANNsH)	Linear Regression Output (R_L)	Polynomial Regression Output (R_P)
Mean	11.727	11.614	11.944	11.944
Standard Deviation	10.588	10.491	10.609	10.511
Mean of Difference		0.0037	0.006	0.005
Standard Deviation of Difference		0.1408	2.9611	0.6894
Value of t		0.1928	-0.3926	-0.3840
R square value		0.99959	0.921	0.996

Mean is obtained by dividing the sum of entire database by the total number of units present in a database. It gives the average value of all the observations in a population. In the present case, the mean value of deflection is very close to each other for all the prediction techniques. To be more precise, mean value of ANNs, H validation (ANNsH) comes out to be 11.614, which is somewhat more close to the output mean value of 11.727, as compared to R_L and R_P mean value of 11.944. The results of mean values as shown in table 6.3, implies that, ANNsH is giving values in conformance with the values obtained by all the other techniques. However, mean alone has no meaning, as it does not give any information about the spread or distribution of the observations. For evaluation of the spread of distribution, mean value is to be accompanied with standard deviation (SD). For the detailed analysis output, SD for deflection comes out to be 10.588, whereas SD for ANNsH, R_L and R_P comes out to be 10.491, 10.609 and 10.511 respectively. Here, the SD for all the techniques are in close proximity with each other. Hence the entire output obtained by different techniques falls within the same range of mean $\pm$ SD and as such all the techniques can be considered as equally good.

Further, mean of difference between detailed analysis output and outputs from other techniques are calculated and their SD have been determined. This is required for analyzing the distribution of errors from various analysis techniques. The mean of difference for ANNsH, R_L and R_P comes out to be 0.0037, 0.006 and 0.005, respectively, whereas SD for mean of difference comes out to be 0.1408, 2.961 and 0.6894 respectively. The above presented results for the SD shows that the errors are quiet close to the standard output values for ANNsH technique. SD for R_L is very high and indicates that in linear regression technique, the distribution of errors is very broad. For R_P, although the SD is less than R_L, but still higher than ANNsH value of SD. From this discussion, it can be said, that ANNsH proves to be a better prediction technique than the regression modals for determining the deflection in PT slabs.

A “t” test is the statistical examination of two population mean. Its common application is to test if a new process or treatment is superior to a current process or treatment. Two-tailed t-test, with a level of significance as 95% (α value) was performed on the outputs obtained from various data analysis techniques. The sample size was detailed analysis output was 799 data units whereas sample size for ANNsH, was 533 and for R_L & R_P was 600 data units. A range of ‘t’ between +1.96 to -1.96 implies that the sample mean complies with the population mean. It can be seen, from table 6.3 that the value of t for all the three techniques, viz; ANNsH, R_L and R_P, lays within the said limits. It implies that the population mean for all the three analysis techniques lies within the same range. Further, it can be noted that the ‘t’ value of 0.1928 for ANNsH is more closer to the center of normal distribution curve as compared to the values, -0.3926 and -0.3840 for R_L and R_P techniques respectively.

Coefficient of determination (R^2) is a statistic that gives information about the goodness of fit of a model. In regression, the R^2 is a statistical measure of how well the regression line approximates the real data points. As can be seen from table 6.3, R^2 for ANNsH comes out to be 0.9996, whereas for R_L and R_P techniques, the value comes out to be 0.921 and 0.996 respectively. A R^2 value of one indicates that the regression line perfectly fits the data. For the present case, R^2 value for ANNsH is quite close to 1.0 as compared to other regression techniques. Hence, it shows that ANNsH predicted output values are able to fit the regression curve more efficiently than any other regression techniques.

Table 6.4: Outputs for Artificial Neural Network (ANN), Linear and Polynomial Regression models for Weight of Post Tensioned Steel.

Description	**Detailed Analysis Output**	**ANN (Holdout) Output (ANNsH)**	**Linear Regression Output (R_L)**	**Polynomial Regression Output (R_P)**
Mean	3.9267	3.5952	3.5946	3.5944
Standard Deviation	0.817	0.620	0.647	0.564
Mean of Difference		0.0000	0.002	0.001
Standard Deviation of Difference		0.0153	0.5201	0.2691
Value of t		0.84012	1.03182	0.89845
R square value		0.99989	0.307	0.815

The results obtained from ANNs outputs with linear regression outputs and ANNs outputs with polynomial regression outputs for the weight of PT steel are tabulated in table 6.4 for a comparative study. The mean value of weight for detailed analysis output as well as other analysis techniques are in close proximity with each other. The mean value for ANNsH, R_L and R_P comes out to be 3.5952, 3.5946 and 3.5944 respectively, which are very close to the output mean value of 3.9267. The results of mean values as shown in table 6.4, implies that, ANNsH is giving values in conformance with the values obtained by all the other techniques. For the detailed analysis output, SD for weight of PT steel comes out to be 0.817, whereas SD for ANNsH, R_L and R_P comes out to be 0.620, 0.647 and 0.564 respectively. Here, the SD for all the techniques is quite close to each other. Hence the entire output obtained by different techniques falls within the same range of mean $\pm$ SD and as such all the techniques can be considered as equally good.

Further, the mean of difference for ANNsH, R_L and R_P comes out to be 0.0000, 0.002 and 0.001 respectively, whereas SD for mean of difference comes out to be 0.0153, 0.5201 and 0.2691 respectively. It can be seen that, the SD is minimum for ANNsH technique, which implies that the distribution of errors are quiet close to the standard output values. SD for R_L is very high and indicates that in linear regression technique, the distribution of errors is very broad. For R_P, although the SD is less than R_L, but still higher than ANNsH value of SD. From this discussion, it can be said, that ANNsH proves to be a better prediction technique than the regression modals for determining the deflection in PT slabs.

For, two-tailed t-test, with a level of significance as 95%, table 6.4 shows that the value of 't' for all the three techniques, viz; ANNsH, R_L and R_P, lays within the said limits (between +1.96 to -1.96). It implies that the population mean for all the three analysis techniques lies within the same range. Further, it can be noted that the 't' value of 0.84012 for ANNsH is more closer to the center of normal distribution curve as compared to the values, 1.03182 and 0.89845 for RL and RP techniques respectively.

As can be seen from table 6.4, R^2 for ANNsH comes out to be 0.99989, whereas for R_L and R_P techniques, the value comes out to be 0.307 and 0.815 respectively. For the present case, R^2 value for ANNsH is quite close to 1.0 as compared to other regression techniques. Hence, it shows that ANNsH predicted output values are able to fit the regression curve more efficiently than any other regression techniques.

6.5 Experimental Validation of the study:

In the presented research work, ANNs were used to develop a decision support modal to assist the designers of PT slabs. This ANN modal is comprehensively analyzed and validated using various validation techniques. The detailed analysis have shown that the developed ANN modal with Holdout validation technique gives results in conformance with the results as generated from the sophisticated standard design software. ANNs have also proved their superiority over other regression (linear and polynomial) methodologies.

However, all these analysis is based on the database generated using a standard design software. Practically, the design results as obtained from a software, most of the times differs from the results taken from the field. Hence, to touch the practical aspect of the developed ANN modal, so that the expert designers can use them, the experimental validation of ANN modal was commenced. Deflection and PT steel weight data are collected from the expert PT design consultants for various configurations of PT slabs. This collected data was tested by the ANN modal trained and validated by Holdout validation technique.

Table 6.5 shows the five inputs in terms of span, depth, load, column size & grade of concrete including the outputs as deflection & weight of PT steel obtained theoretically, by ANN and from the field data. The variation in PT slab span is taken from a minimum of 7 m to a maximum of 12 m whereas the depth of slab has been taken in a range of 170 mm to 250 mm. A constant live load on 3 kN/m2 is considered on the slab. A square cross section of columns is considered with variation from 450 mm side to 750 mm side. A comparative presentation of the outputs as obtained from theoretical, ANN and field analysis is made here.

Table 6.5: Experimental Validation of Research Work

S. No	INPUTS					THEORITICAL OUTPUT		ANN OUTPUT		FIELD OUTPUT	
	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	CONC. GRADE	DEF (mm)	WT (kg/m^2)	DEF (mm)	WT (kg/m^2)	DEF (mm)	WT (kg/m^2)
1	7	170	3	450	35	2.6	3.25	2.7	3.25	3.5	4.32
2	8	190	3	500	35	4.2	3.61	4.1	3.61	5.0	4.80
3	9	230	3	550	35	4.3	3.44	4.3	3.43	6.5	5.05
4	10	250	3	600	35	6.1	3.89	6.0	3.89	7.5	5.44
5	11	250	3	650	35	11.3	4.24	11.2	4.23	10.0	5.92
6	12	250	3	750	35	17.4	4.51	17.3	4.50	16.5	6.56

Source: M/s Vijaytech associates and M/s Post Tension Services India Pvt. Ltd.

The results for the outputs from theoretical output, ANN analysis and from Field Data are presented graphically in Fig 6.1 for PT slab deflection. It can be seen here that theoretical and ANN outputs are exactly in conformance with each other, however field outputs are deviating somewhat from these values. The maximum variation can be seen in nine meter span, where the difference in deflection is more than 2 mm. For all other spans, this difference in deflection ranges in between 0.5 to 1.0 mm. This range of deflection is quite less while considering a span upto twelve meters

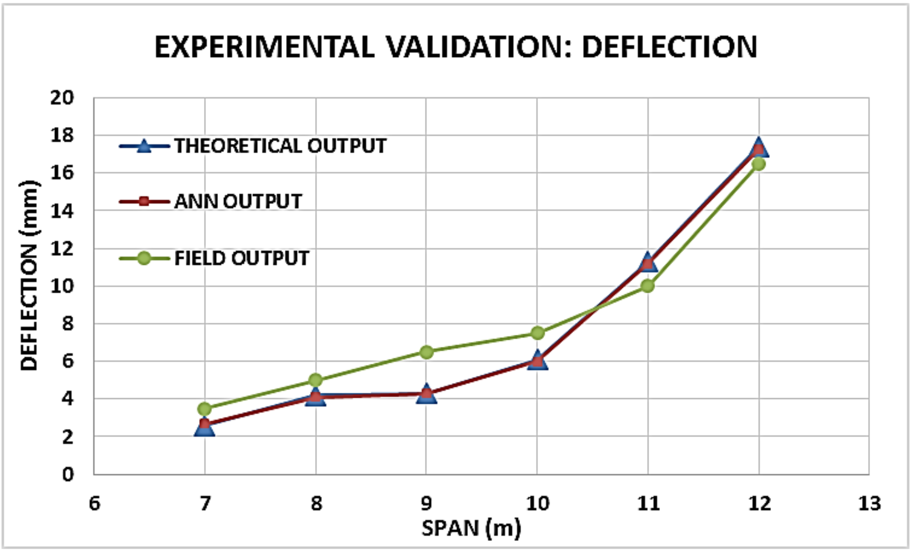


Fig 6.1: Experimental Validation for PT slab deflection

The similar results are presented graphically in Fig 6.2, for PT steel weight. It can be seen here that theoretical and ANN outputs are exactly in conformance with each other, however field outputs are deviating somewhat from these values. The maximum variation can be seen between nine to twelve meter span, where the difference in weight is varying from 0.5 to 1.0 kgs per meter square. This range of variation is considerable for spans upto twelve meters in length.

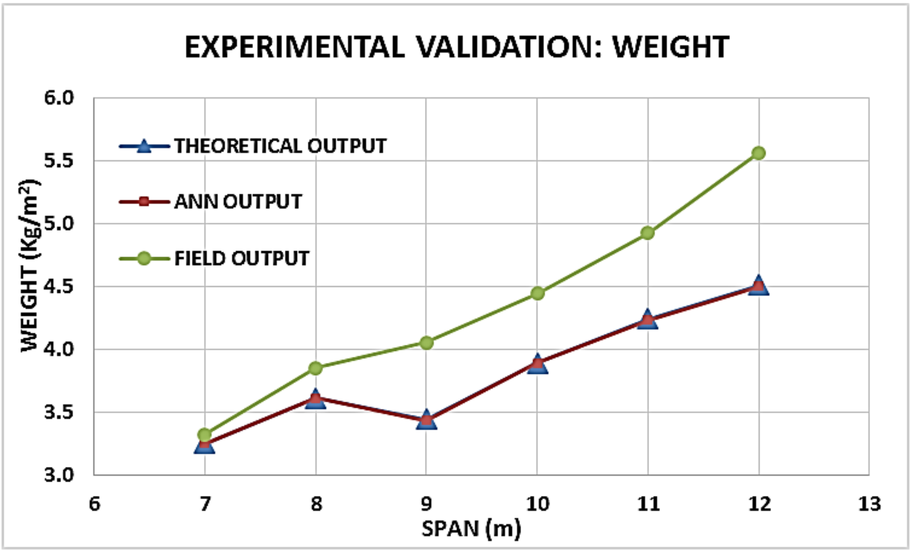


Fig 6.2: Experimental Validation for PT steel weight

The above discussion indicates towards the reliability of using the developed ANN modal for real world problems. The community of expert structural designers will definitely get benefited, having a tool for supporting their design decision.

Table 6.6: Analysis of Experimental Validation for Deflection of PT slab

	TRAINING OUTPUT	TEST OUTPUT	FIELD OUTPUT
MSE	0.0125	0.0260	1.8187
R	0.99994	0.9999	0.9854

Table 6.6, represents the MSE and correlation values for training, testing and field outputs for deflection of PT slabs. MSE value of 1.8187 for field output is higher than the MSE values of 0.0125 and 0.0260 for training and testing outputs respectively. However, the correlation value of 0.9854 is quiet promising for the field data as compared to correlation value of 0.9999 for both training and testing outputs. Also, the MSE and correlation values for training, testing and field outputs for weight of PT steel has been presented in Table 6.7. MSE value of 0.3852 for field output is higher than the MSE values of 0.00002 and 0.00003 for training and testing outputs respectively. However, the correlation value of 0.9740 is quiet promising for the field data as compared to correlation value of 0.9999 for both training and testing outputs.

Table 6.7: Analysis of Experimental Validation for Weight of PT steel

	TRAINING OUTPUT	TEST OUTPUT	FIELD OUTPUT
MSE	0.00002	0.00003	0.3852
R	0.99998	0.99998	0.9740

This may be due to the fact that the field data collected for the input, column, consists of variations which is not considered in the entire database. For eg., data base consisted of only three sizes of column as, 450 mm by 450 mm, 600 mm by 600 mm and 750 mm by 750 mm whereas field data consisted of 500 mm by 500 mm and 650 mm by 650 mm column sizes also. Secondly, the value of outputs for the same inputs varies somewhat for theoretical output and field data. The network has undergone training for input parameters in a particular range only. The network cannot determine the variations out of these ranges, as it has never experienced it during training. Hence, the results for experimental validation are varying to some extent.

However, since the theoretical and field data are correlating to a good extent of over 0.97, the expert designers of PT slabs can use the developed network as a decision support tool for their engineering designs.

6.6 Sensitivity Analysis:

Sensitivity analysis is performed to see the response of the outputs when some small change is brought about in the input design parameters. The more sensitive outputs will vary to a large extent for a particular input parameter even with a very small change. Fig 6.3 shows the percentage variation in the outputs namely, Deflection of PT slab (DEF) and weight of PT steel (WT) when a small variation of 1% is brought about in the input design parameter namely, Span, Depth, Load, Column size and Concrete grade.

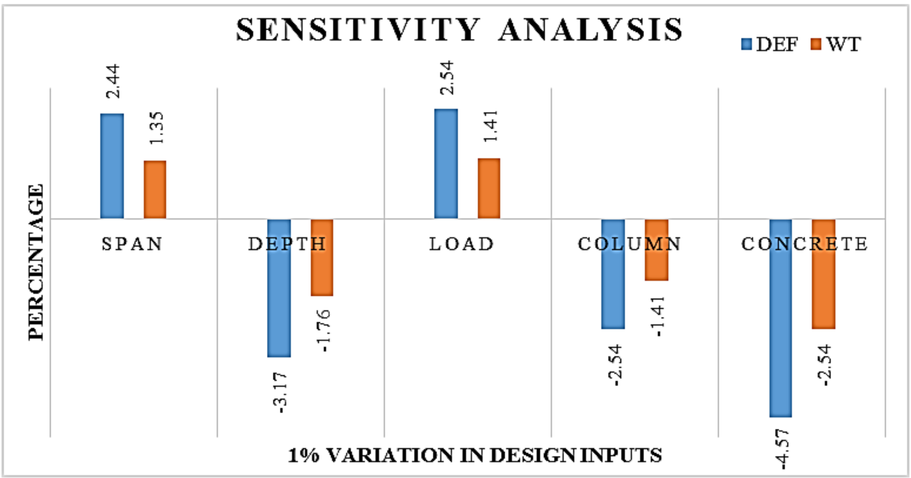


Fig 6.3: Sensitivity Analysis of Outputs

The results of sensitivity analysis shows that the grade of concrete is influencing the output parameters to a larger extent. With 1% variation in the grade of concrete, deflection of PT slab is changed by 4.57% whereas the variation of 2.54% can be noted in case of weight of PT steel. The grade of concrete is parameter for the strength of concrete. The higher the grade of concrete, the more will be the load taking capacity and hence the deflection will be compensated, and less steel will be required.

Rest, all other input parameters, are influencing the deflection and weight of PT steel in a range of one to three percent, which is not very significant.

6.7 Comparative study with respect to the reported papers:

The developed neural network has been validated and analyzed thoroughly to be used to support the expert designers of PT slab in making design decisions. At the same time, a comparative study of the developed ANN modal have been conducted and represented in Table 6.8, with respect to some of the papers reported in the Literature review.

Table 6.8: Comparative study of research work with respect to the reported papers

S.No	Author	Network	Training data	Testing data	Hidden Layer	Hidden Neurons	Algorithm	Testing R
1.	Paper Title: Application of ANN in conceptual design of PT slabs							
	Present Research work	Holdout	533	266	2	9,9	Backpropagation	0.999
2.	Paper Title: Neural network based approach for determining the shear strength of circular reinforced concrete columns							
	Caglar N. 2009	Scaled conjugate gradient	31	16	2	6,3	Backpropagation	0.833
3.	Paper Title: Artificial neural network model for steel-concrete bond prediction							
	Dahou Z. et al. 2009	Cross validation	90	22	1	10	Backpropagation	0.89
4.	Paper Title: Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete							
	Ozcan F. et al. 2009	Multilayer feed forward network	135	50v* 55t+	1	11	Backpropagation	0.9767
5.	Paper Title: Investigation of flow resistance in smooth open channels using ANN							
	Bilgil A. and Altun H. 2008.	Multi-Layered Perception NN	48	47	2	7,15	Backpropagation	0.9926
6.	Paper Title: Appraisal of long-term effects of fly ash and silica fume on compressive strength of concrete by neural networks							
	Pala M. et al. 2007	Scaled conjugate gradient	130	13	1	9	Backpropagation	0.9990
7.	Paper Title: Modeling Confinement Efficiency of Reinforced Concrete Columns with Rectilinear Transverse Steel Using ANNs							
	Tang C.W. et al. 2003	Multilayer-functional-link NN	45	10	1	14	Backpropagation	0.9217

Today several ANN techniques are available and still for setting the neural network architecture, we largely depends on the trial methods. This comparative presentation will help the researchers, to get a review of various architectures of neural network being used in various structural problems. Researchers will get to know about the size of database for training the network and will also have an idea for the hidden layer neurons. Also they will be benefitted by having a comparison of R2 values obtained by using different techniques.

6.8 Guidelines to the designers of PT slabs:

The ANN tool developed in this research work conforms to the objective of its utility for the structural experts of PT slabs who will conceptually arrive at a most reliable solution. Also for using this tool, designer need not to be an expert from the field of ANN. [Ferreira IML and Gil PJS 2012].

Following guidelines are proposed for using the developed tool by the structural experts and researchers:

1. *Number of hidden layers*: as per the experimental investigations, it is suggested to go for two number of hidden layers while estimating for both, the deflection of PT slab and weight of post tensioning steel.
2. *Number of hidden layer nodes*: Several combinations for the number of hidden layer nodes have been experimented with and have been reported. Results show that using nine neurons in each of the two layers gives precise results.
3. *Training function*: Researchers are strongly recommended to consider Levenberg-Marquardt backpropagation algorithm with ANN out of the several algorithm available. Several researches as well the presented work indicates the advantage of Levenberg-Marquardt backpropagation algorithm over other algorithms.
4. *Transfer function*: Using sigmoidal functions empowers ANN, because of their advantage of being differentiable and bringing the error to a minimum. Log sigmoid transfer function should be used for data range between 0 to +1 and tan sigmoid transfer function is recommended for data range from -1 to +1.
5. *Validation techniques*: four validation techniques have been presented and compared in this research work. In the present case, for the analysis of deflection

and weight of post tensioning steel in PT slabs, Holdout technique comes out with the more precise results as compared to other validation techniques. Hence researchers are advised to consider Holdout technique for the validation of their data along with other validation techniques.

This study aims at providing ANN tool at conceptual stage of design and its application is limited to the determination of deflection of PT slab and weight of post tensioning steel. Hence, the architecture of the ANN proposed is valid for PT slabs only. Also the valid range of structural parameters for the developed ANN modal is as tabulated in Table: 5.1 under column title ‘validation range’. The researchers are advised to go through the references and current studies on ANN for the design and analysis of other structural forms.

6.9 Summary

A conceptual design tool is developed and the results as obtained are quiet encouraging. Here, a platform is created for the ANN tool for proving its reliability over other prevailing statistical techniques. Various regression modals were also developed for testing the goodness of fit and analyzed with respect to the results of the developed ANN modal. Values of mean, standard deviation, t value and correlation coefficients have been determined and compared for holdout, linear regression and polynomial regression modals. The results have shown ANNs to be more reliable to the problems related to engineering domain, particularly for the design of PT slabs.

CHAPTER-7

Conclusions

7.1 Introduction:

Research work carried out in this thesis presents a novel idea of developing a decision support tool using ANNs technique which, have an experience of a sophisticated design software, for the design of PT slabs at conceptual stage of design. It is indicative from the detailed analysis presented in the thesis, that such decision support tool can be used for the prediction of deflection and weight of post tensioning steel in PT slabs at conceptual stage of design. Here we have also presented a set of guidelines for analyzing the PT slabs using developed ANN model. By using the suggested model, structural design engineers would be able to determine the design parameters at conceptual stage of design. These parameters would keep deflection under the limits and would cater for the economy of the project by suggesting the optimal quantity of post tensioning steel.

In the present research work, a detailed study has been commenced for providing a base for the conceptual design of post-tensioned slabs. Artificial Neural Networks have been used as a tool for serving the purpose from the field of Artificial Intelligence. In this work, observation is made on the variation in the weight of PT steel and on the deflection of PT slab. A large number of data set was generated with different slab configurations to feed into the network as input. Various ANN models with both single layer and double layers were created with different architectures. Levenberg-Marquardt training algorithm and Resilient Backpropagation algorithm are used for training the networks. Log-Sigmoid and Tan-Sigmoid are used as the activation functions. The results obtained have been validated using four different validation techniques. The number of epochs is taken upto 1000 cycles.

Conclusions drawn out from the detailed analysis of the results is summarised below:

1. The research shows that, Artificial Neural Networks can be used for conceptual design of post-tensioned slabs. These networks are robust and give results that are more precise than obtained by regression analysis.

2. For making the networks learn, Levenberg-Marquardt training algorithm and Resilient backpropagation algorithm were employed. The research indicates that training by Levenberg-Marquardt algorithm gives the best result.
3. Twelve neural networks models were developed, out of which four were single layered networks and eight were double layer networks. Results indicated that networks with double layer have given far better results as compared to single layer networks.
4. The research shows that one unique neural network may not be best for prediction of all the output variables. Different models may have to be used for different output variables. In the present case, the best performance in terms of deflection of PT slabs, has been given by the network NET7 with two hidden layers, having nine neurons in each layer with Levenberg-Marquardt backpropagation algorithm and with Tan-sigmoid activation function in first layer and Log-sigmoid in the second. The training MSE came out to be 0.0191 and the testing MSE for this network came out to be 0.0159. Also, the best performing network for the weight of post tensioning steel came out to be the network NET6, with two hidden layers, having nine neurons in each layer with Levenberg-Marquardt backpropagation algorithm and with Log-sigmoid activation function in both the layers. The training MSE came out to be 0.000238 and the testing MSE for this network came out to be 0.00021.

5. Validation of artificial neural networks is important before it can be used for any practical purpose. Validation ensures the robustness on any decision making tool. In the present research work resubstitution, holdout, three-way data split and k-fold cross validation techniques were used. The research concludes that holdout validation technique is best for ensuring robustness of the artificial neural network model.

6. Sensitivity analysis shows that deflection is most affected by the variation in the grade of concrete. Rest, other design parameters does not influence the deflection and weight of PT steel largely.

7. Experimental validation indicated to the fact that the developed ANN modal can be reliably used as a decision support system by the experts designers of PT slabs.

8. Comparative study clears the scope for the researchers for deciding the neural network architecture taking up the structural design problem.

7.2 Use of developed ANN Modal:

The developed ANN modal can be used very easily as a decision support tool for testing and evaluating the real world problems related to design and analysis of PT slabs. The developed model is in a form of a excel file which can be put up on the server and anybody can download it for its usage. The only requirement is that the user's computer must have MATLAB software installed. This excel file consists of macros, that calls up neural network toolbox commands from MATLAB program. Once the program runs, all the data are tabulated in an excel sheet itself.

7.3 Further Research

1) In this research work three span continuous flat post tensioned slab with drop panels have been analysed for design at conceptual stage. This research can be further extended for different type of prestress slab and beam systems.

2) Although Levenberg-Marquardt training algorithm has provided the best training results but still various different training functions may be applied for choosing the best-trained network.

3) Various other types of neural networks such as Radial basis networks, Genetic Algorithms, Recurrent networks, Self-Organizing Maps, etc., may also be applied to the same slab configurations and then compared with backpropagation neural networks.

4) Network validation is one of the most important parts of network selection. It is the foundation on which the reliability of the performance of neural networks is supported. For this, more work should be done on validation techniques to be applied on the networks.

5) The range of design parameters for which the decision support tool is developed, may have close variations so that the experimental validation can be more fruitful.

7.4 Summary

The research presented in this thesis strongly recommends the use of ANNs in the civil and structural engineering domain. Several researchers round the globe are experimenting on the modelling of architecture of ANNs and their performance evaluation and here we have presented a particular case of deigning of PT slabs for its deflection and weight of post tensioning steel using ANN. The novelty of the research is based on the fact that a conceptual design tool have been developed to help the designers for determining the major design constraints at the preliminary stage of design. For the same, guidelines have also been proposed to the researchers for the architecture of ANN, training algorithms, transfer functions and the number of layers and nodes to be considered. Also using this tool would let the new designers to have a reasonable idea for the design outcomes instead of vague or no outcomes.

REFERENCES

1. **Agrawal V., Nagar R., and Sancheti G. (2011)**,“ Application of Artificial Neural Network in Conceptual Design of Communication Towers”, Proceedings of the International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2011), Planetary Scientific Research Centre Conferences, Pattaya, Thailand.

2. **Aktan A.E. and Bertero F. (1984)**, “Conceptual Seismic Design of Frame-Wall Structures”, Journal of Structural Engineering, Vol.110, No.11, pp. 2778–2797.

3. **Ashour A.F. and Alqedra M.A. (2005)**, “Concrete breakout strength of single anchors in tension using neural networks”, Advances in Engineering Software, Vol.36, pp. 87–97.

4. **Asmar M.E., Hanna A.S. and Whited G.C. (2011)**, “New Approach to Developing Conceptual Cost Estimates for Highway Projects”, Journal of Construction Engineering and Management, Vol.137, No.11, pp. 942–949.

5. **Attoh-Okine N. O. (2002)**, “Combining Use of Rough Set and Artificial Neural Networks in Doweled-Pavement-Performance Modelling—A Hybrid Approach”, Journal of Transportation Engineering, Vol.128, No.3, pp. 270–275.

6. **Bilgil A. and Altun H. (2008),** “Investigation of flow resistance in smooth open channels using artificial neural networks”, Flow Measurement and Instrumentation, Vol. 19, pp. 404 - 408.

7. **Caglar N. (2009)**, “Neural network based approach for determining the shear strength of circular reinforced concrete columns”, Construction and Building Materials, Vol.23, pp. 3225–3232.

8. **Calvi G.M., Sullivan T.J. and Villani A. (2010)**, “Conceptual Seismic Design of Cable-Stayed Bridges”, Journal of Earthquake Engineering, Vol.14, pp. 1139–1171.

9. **Chao L.C. and Skibniewski M.J. (1994)**, “Estimating Construction Productivity: Neural-Network-Based Approach”, Journal of Computing in Civil Engineering, Vol. 8, No. 2, pp. 234–251.

10. **Chen D. and Hagan M.T. (1999)**, “Optimal Use of Regularization and Cross-Validation in Neural Network Modeling”, IEEE 1999, pp. 1275- 1280.

11. **Dahou Z., Sbartaï Z.M., Castel A., and Ghomarid F. (2009),** “Artificial neural network model for steel-concrete bond prediction”, Engineering Structures, Vol.31, pp. 1724-1733.

12. **De Paoli D.W., Wilson J.H. and Thomas C.O. (1996)**, “Conceptual Design of Soil Venting Systems”, Journal of Environmental Engineering, Vol.122, No.5, pp. 0399–0406.

13. **Elazouni A.M., Nosair I.A., Mohieldin Y.A. and Mohamed A.G. (1997)**, “Estimating Resource Requirements at Conceptual Design Stage Using Neural Networks”, Journal of Computing in Civil Engineering, Vol.11, No.4, pp. 217–223.

14. **Eldin N.N. (1988)**, “Constructability Improvement of Project Designs”, Journal of Construction Engineering and Management, Vol.114, No.4, pp. 631–640.

15. **Feng M.Q., Kim D.K., Yi J.H. and Chen Y. (2004)**, “Baseline Models for Bridge Performance Monitoring”, Journal of Engineering Mechanics, Vol.130, No.5, pp. 562–569.

16. **Ferreira I.M.L. and Gil P.J.S. (2012),** “Application and Performance Analysis of Neural Networks for Decision Support in Conceptual Design”, Expert Systems with Applications, Vol. 39, pp. 7701-7708.

17. **Gilbert R.I. and Mickleborough N.C. (2005),** “Design of Prestressed Concrete”, Book published by Academic devision of Unwin Hyman Ltd, London.

18. **Grierson D.E. and Khajehpour S. (2002)**, “Method for Conceptual Design Applied to Office Buildings”, Journal of Computing in Civil Engineering, Vol.16, No.2, pp. 083–103.

19. **Hirooka S., Hattori K., and Takeda T (2011)**, “Development and Validation of Neural Network Based Ionospheric Tomography”, IEEE 2011.

20. **Honigmann C. and Billington D.P. (2003)**, “Conceptual Design for the Sunniberg Bridge”, Journal of Bridge Engineering, Vol.8, No.3, pp. 0122–0130.

21. **Hull J., Ward D. and Zakrzewski R.R. (2002)**,“Verification and Validation of Neural Networks for Safety-Critical Applications”, Proceedings of the American Control Conference Anchorage, IEEE 2002, pp. 4789- 4794.

22. **Huynh T.Q. and Setiono R. (2005)**, “Effective neural network pruning using cross-validation”, Proceedings of International Joint Conference on Neural Networks, Montreal, Canada, IEEE 2005, pp. 972- 977.

23. **Inel M. (2007)**, “Modeling ultimate deformation capacity of RC columns using artificial neural networks”, Engineering Structures, Vol.29, pp. 329–335.

24. **IS:1343 (1980)**,“Code of Practice For Prestressed Concrete (first revision)”.

25. **IS:6006 (1983)**,“ Specification for uncoated stress relieved strand for prestressed concrete (first revision)”.

26. **Jeyasehar C.A. and Sumangala K. (2006)**, “Damage assessment of prestressed concrete beams using artificial neural network (ANN) approach”, Computers and Structures, Vol.84, pp. 1709–1718.

27. **Jingling Y., Luo Z., Hongfu D. and Haizheng T. (2012),** “Prediction of Free Lime Content in Cement Clinker Based on RBF Neural Network”, Journal of Wuhan University of Technology, Vol.27, No.1, pp. 187-190.

28. **Joghataie A. and Farrokh M. (2008)**, “Dynamic Analysis of Nonlinear Frames by Prandtl Neural Networks”, Journal of Engineering Mechanics, Vol.134, No.11, pp. 961–969.

29. **Jung S. and Kim K.S. (2008),** “Knowledge-based prediction of shear strength of concrete beams without shear reinforcement”, Engineering Structures, Vol. 30, pp. 1515–1525.

30. **Jung S., Ghaboussi J., and Kwon S.D. (2004)**, "Estimation of Aeroelastic Parameters of Bridge Decks Using Neural Networks", Journal of Engineering Mechanics, Vol.130, No.11, pp. 1356–1364.

31. **Kao C.Y. and Hung S.L. (2003)**, "Detection of structural damage via free vibration responses generated by approximating artificial neural networks", Computers and Structures, Vol.81, pp. 2631–2644.

32. **Keller R., Eckert C.M. and Clarkson P.J. (2009)**, "New Approach to Developing Conceptual Cost Estimates for Highway Projects", Journal of Construction Engineering and Management, Vol.20, No.6, pp. 571–587.

33. **Kerh T. and Yee Y.C. (2000)**, "Analysis of a deformed three-dimensional culvert structure using neural networks", Advances in Engineering Software, Vol.31, No.2, pp. 367–375.

34. **Kim J.I., Kim D.K., Feng M.Q. and Yazdani F., (2004)**, "Application of Neural Networks for Estimation of Concrete Strength", Journal of Materials in Civil Engineering, Vol.16, No.3, pp. 0899-1561.

35. **Lai S. and Mauro S. (1997)**, "Concrete strength prediction by means of neural network", Construction and Building Materials, Vol.11, No. 2, pp. 93-98.

36. **Larsen J., Hansen L.K., Svarer C. and Ohlsson M. (1996)**, "Design and Regulation of Neural Networks: The Optimal use of a Validation Set", IEEE 1996, pp. 62 - 71.

37. **Li Q.S., Liu D.K., Fang J.Q., Jeary A.P. and Wong C.K. (2000)**, "Damping in buildings: its neural network model and AR model", Engineering Structures, Vol.22, pp. 1216–1223.

38. **Liu Y. and Wakamatsu A. (2006)**, "Create Stable Neural Networks by Cross-Validation", International Joint Conference on Neural Networks, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, IEEE 2006, pp. 3925 - 3928.

39. **Mallet C., Barthes L. and Marsault T. (2006)**,"Validation of a Neural Network Model for the Separation of Atmospheric Effects on Attenuation", European Conference on Antennas and Propagation.

40. **Md. Azamathulla H., Deo M.C., and Deolalikarv P.B. (2008)**, "Alternative neural networks to estimate the scour below spillways", Advances in Engineering Software, Vol.39, No.2, pp. 689–698.

41. **Mehrjoo M., Khaji N., Moharrami H., and Bahreininejad A. (2008)**, "Damage detection of truss bridge joints using Artificial Neural Networks", Expert Systems with Applications, Vol.35, pp. 1122–1131.

42. **Mohamed S. K., Zhen-Ping L. and Stephen G. R. (1994)**, "Comparison of Traditional and Neural Classifiers for Pavement-Crack Detection", Journal of Transportation Engineering, Vol.120, No.4, pp. 552–569.

43. **Mojarad S.A., Dlay S.S., Woo W.L. and Sherbet G.V. (2010)**, "Breast Cancer Prediction and Cross Validation Using Multilayer Perceptron Neural Networks", Communication Systems Networks and Digital Signal Processing, IEEE 2010, pp. 760 - 764.

44. **Mukherjee A. and Biswas S. N. (1997)**, "Artificial neural networks in prediction of mechanical behaviour of concrete at high temperature", Nuclear Engineering and Design, Vol.178, No.2, pp. 1–11.

45. **Özcan F., Cengiz D. Atis, Karahan O., Uncuoglu E., and Tanyildizi H. (2009)**, "Comparison of artificial neural network and fuzzy logic models for prediction of long-term compressive strength of silica fume concrete", Advances in Engineering Software, Vol.40, pp. 856–863.

46. **Pala M., Ozbay M., Oztas M. and Yuce M.I., (2007)**, "Appraisal of long-term effects of fly ash and silica fume on compressive strength of concrete by neural networks", Construction and Building Materials, Vol.21, pp. 384–394.

47. **Park K. and Medina R.A. (2007)**, "Conceptual Seismic Design of Regular Frames Based on the Concept of Uniform Damage", Journal of Structural Engineering, Vol.133, No.7, pp. 945–955.

48. **Quirant J., Cevaer F., Morterolle S., Maurin B. and Dube J.F. (2011)**, "Conceptual Design and Analysis of a Deployable Structure with Flexible Joints", Journal of Aerospace Engineering, Vol.24, No.3, pp. 0277–0284.

49. **Rafiq M.Y., Bugmann G. and Easterbrook D.J. (2001)**, "Neural network design for engineering applications", Journal of Computers & Structures, Vol.79, No.17, pp. 1541–1552.

50. **Rao H.S. and Babu B.R. (2007)**, "Hybrid neural network model for the design of beam subjected to bending and shear", Sadhana, Vol.32, Part 5, pp. 577–586.

51. **Reich Y. and Barai S.V. (1999),** "Evaluating Machine Learning Models for Engineering Problems", Artificial Intelligence in Engineering, Vol. 13, pp. 257-272.

52. **Reich Y. and Barai S.V. (2000),** "A Methodology for Building Neural Networks Models from Empirical Engineering Data", Engineering Applications of Artificial Intelligence, Vol. 13, pp. 685–694.

53. **Rivard H. and Fenves S. J. (2000)**, "A Representation For Conceptual Design of Buildings", Journal of Computing in Civil Engineering, Vol.14, No.3, pp. 0151–0159.

54. **Sancheti G., Agrawal V. and Nagar R., (2009)**,"Neural Network Applications in Concrete Technology: A Review", Proceedings of National Conference on Sustainable Concrete Infrastructure Development, Indian Concrete Institute and Malaviya National Institute of Technology Jaipur.

55. **Sisk G.M., Miles J.C. and Moore C.J. (2003)**, "Designer Centered Development of GA-Based DSS for Conceptual Design of Buildings", Journal of Computing in Civil Engineering, Vol.17, No.3, pp. 159–166.

56. **Sofuoglu S.C. (2008)**,"Application of artificial neural networks to predict prevalence of building-related symptoms in office buildings", Building and Environment, Vol.43, No.2, pp. 1121–1126.

57. **Song H.W. and Kwon S.J. (2009)**, "Evaluation of chloride penetration in high performance concrete using neural network algorithm and micro pore structure", Cement and Concrete Research, Vol.39, pp. 814–824.

58. **Srinivas V. and Ramanjaneyulu K. (2007)**, "An integrated approach for optimum design of bridge decks using genetic algorithms and artificial neural networks", Advances in Engineering Software, Vol.38, No.2, pp. 475–487.

59. **Sunindijo R.Y. and Zou P.X.W. (2013)**, "Conceptualizing Safety Management in

60. **Swem F.L (1994)**, "A Validation Methodology for Neural Network Based Flight Control Systems", IEEE, pp. 348–352.

61. **Tang C.W., Chen H.J. and Yen T. (2003)**, "Modeling Confinement Efficiency of Reinforced Concrete Columns with Rectilinear Transverse Steel Using Artificial Neural Networks", Journal of Structural Engineering, Vol. 129, No. 6, pp. 775–783.

62. **Tsai C.Y., Wang Y.S., Hsu C.H. and Ou W.M. (2010)**, "The Statistical Validation for Artificial Neural Network", International Symposium on Computer, Communication, Control and Automation, pp. 121- 124.

63. **Twomey J.M. and Smith A.E. (1998)**, "Bias and Variance of Validation Methods for Function Approximation Neural Networks Under Conditions of Sparse Data", IEEE Transactions on system, man and cybernetics—Part C: Applications and reviews, Vol. 28, No.3,pp. 417–430.

64. **Wang K., Yang J., Shi G. and Wang Q. (2008)**, "An Expanded Training Set Based Validation Method to Avoid Overfitting for Neural Network Classifier", IEEE Fourth International Conference on Natural Computation, pp. 83–87.

65. **Wang W. and Gero J.S. (1997)**, "Sequence-Based Prediction in Conceptual Design of Bridges", Journal of Computing in Civil Engineering, Vol.11, No.1, pp. 37–43.

66. **William J.R. and Abudayyeh O.Y. (1992)**, "NIAM Conceptual Data-Base Design in Construction Management", Journal of Computing in Civil Engineering, Vol.6, No.1, pp. 41–62.

67. **Williams M.E. and Hoit M.I. (2004)**, "Bridge pier live load analysis using neural networks", Advances in Engineering Software, Vol.35, pp. 645–652.

68. **Xiang Y. and Tso S. K. (2002)**, "Detection and classification of flaws in concrete structure using bispectra and neural networks", NDT&E international, Vol.35, pp. 19–27.

69. **Ye J. and Campbell R.I. (2006)**, "Supporting conceptual design with multiple VR based interfaces", Journal of Virtual and Physical Prototyping, Vol.1, No.3, pp. 171–181.

70. **Yeung W.T. and Smith J.W. (2005)**, "Damage detection in bridges using neural networks for pattern recognition of vibration signatures", Engineering Structures, Vol.27, pp. 685–698.

71. **Zaid M., Gaydecki P., Quek S., Miller G. and Fernandes B. (2004)**, "Extracting dimensional information from steel reinforcing bars in concrete using neural networks trained on data from an inductive sensor", NDT&E International, Vol.37, pp. 551–558.

72. **Zhang L.F., Zhu Q.M. and Longden A. (2009),** "A Correlation-Test-Based Validation Procedure for Identified Neural Networks", IEEE Transactions on Neural Networks, Vol. 20, No.1, pp. 1–13.

BIO-DATA

The author obtained his Bachelor's degree in Civil Engineering from MNIT Jaipur in the year 2005 and Master's degree in Transportation Engineering from the same institute in 2008. Author is presently working as Associate Professor of Civil Engineering Department, Manipal University Jaipur (MUJ). Previously, the author has worked as a Reader, Swami Keshwanand Institute of Technology (SKIT) Jaipur, Dean academics/HoD, Baldevram Mirdha Institute of Technology Jaipur. Author have worked as a Guest Faculty in Civil Engineering Department, Malaviya National Institute of Technology Jaipur and as a field engineer taking up national level projects of Nuclear Power Plants in Rajasthan, Rail cum Road over Bridge in Bihar and River Interlinking Project in Andra Pradesh. He has also worked as a design engineer in Delhi Metro Rail Corporation taking up Jaipur Metro Rail Project.

List of publications are as follows:

1. Sancheti G., Nagar R., Agrawal V., "Prediction of Deflection in Post-Tensioned Slabs at Conceptual Stage of Design by Applying Resubstitution Validation Technique", Internatinal Journal of Civil Engineering, Vol.3, No.3, pp.55-62, May 2014.

2. Agrawal V., Nagar R., and Sancheti G.," Application of Artificial Neural Network in Conceptual Design of Communication Towers", Proceedings of the International Conference on Electrical, Electronics and Civil Engineering (ICEECE'2011), Planetary Scientific Research Centre Conferences, Pattaya, Thailand, 2011.

3. Sancheti G., Agrawal V., Nagar R., "Neural Network Applications in Concrete Technology: A Review", Proceedings of National Conference on Sustainable Concrete Infrastructure Development, Indian Concrete Institute and Malaviya National Institute of Technology Jaipur, May 19 – 20, 2009.

ANNEXURES

Annexure-I: Design Database

Input and output data provided to the network has been shown below. The deflection mentioned in output is calculated at the centre of the slab panel.

	INPUTS					OUTPUTS	
S.No	**SPAN (m)**	**DEPTH (mm)**	**LOAD (kN/m^2)**	**COLUMN (mm)**	**GRADE OF CONCRETE**	**DEFLECTION (mm)**	**STEEL (kg/m^2)**
1	7	170	3	450	35	2.6	3.25
2	7	170	3	450	40	2.4	3.25
3	7	170	3	450	45	2.2	3.25
4	7	170	3	600	35	2.5	3.25
5	7	170	3	600	40	2.3	3.25
6	7	170	3	600	45	2.2	3.25
7	7	170	3	750	35	2.5	3.25
8	7	170	3	750	40	2.3	3.25
9	7	170	3	750	45	2.2	3.25
10	7	170	4	450	35	3	3.25
11	7	170	4	450	40	2.8	3.25
12	7	170	4	450	45	2.7	3.25
13	7	170	4	600	35	3	3.25
14	7	170	4	600	40	2.8	3.25
15	7	170	4	600	45	2.6	3.25
16	7	170	4	750	35	2.9	3.25
17	7	170	4	750	40	2.7	3.25
18	7	170	4	750	45	2.6	3.25
19	7	170	5	450	35	3.5	3.25
20	7	170	5	450	40	3.3	3.25
21	7	170	5	450	45	3.1	3.25
22	7	170	5	600	35	3.5	3.25
23	7	170	5	600	40	3.2	3.25
24	7	170	5	600	45	3.1	3.25
25	7	170	5	750	35	3.4	3.25
26	7	170	5	750	40	3.2	3.25
27	7	170	5	750	45	3	3.25
28	7	190	3	450	35	1.3	2.99
29	7	190	3	450	40	1.2	2.99
30	7	190	3	450	45	1.1	2.99
31	7	190	3	600	35	1.2	2.99
32	7	190	3	600	40	1.1	2.99
33	7	190	3	600	45	1.1	2.99

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
34	7	190	3	750	35	1.2	2.99
35	7	190	3	750	40	1.1	2.99
36	7	190	3	750	45	1.1	2.99
37	7	190	4	450	35	1.6	2.99
38	7	190	4	450	40	1.5	2.99
39	7	190	4	450	45	1.4	2.99
40	7	190	4	600	35	1.6	2.99
41	7	190	4	600	40	1.5	2.99
42	7	190	4	600	45	1.4	2.99
43	7	190	4	750	35	1.5	2.99
44	7	190	4	750	40	1.4	2.99
45	7	190	4	750	45	1.4	2.99
46	7	190	5	450	35	2	2.99
47	7	190	5	450	40	1.8	2.99
48	7	190	5	450	45	1.7	2.99
49	7	190	5	600	35	1.9	2.99
50	7	190	5	600	40	1.8	2.99
51	7	190	5	600	45	1.7	2.99
52	7	190	5	750	35	1.9	2.99
53	7	190	5	750	40	1.8	2.99
54	7	190	5	750	45	1.7	2.99
55	7	210	3	450	35	0.4	2.70
56	7	210	3	450	40	0.4	2.70
57	7	210	3	450	45	0.3	2.70
58	7	210	3	600	35	0.4	2.70
59	7	210	3	600	40	0.4	2.70
60	7	210	3	600	45	0.3	2.70
61	7	210	3	750	35	0.4	2.70
62	7	210	3	750	40	0.4	2.70
63	7	210	3	750	45	0.3	2.70
64	7	210	4	450	35	0.7	2.70
65	7	210	4	450	40	0.6	2.70
66	7	210	4	450	45	0.6	2.70
67	7	210	4	600	35	0.6	2.70
68	7	210	4	600	40	0.6	2.70
69	7	210	4	600	45	0.6	2.70
70	7	210	4	750	35	0.6	2.70
71	7	210	4	750	40	0.6	2.70
72	7	210	4	750	45	0.6	2.70

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
73	7	210	5	450	35	0.9	2.70
74	7	210	5	450	40	0.9	2.70
75	7	210	5	450	45	0.8	2.70
76	7	210	5	600	35	0.8	2.70
77	7	210	5	600	40	0.8	2.70
78	7	210	5	600	45	0.8	2.70
79	7	210	5	750	35	0.9	2.70
80	7	210	5	750	40	0.8	2.70
81	7	210	5	750	45	0.8	2.70
82	7	230	3	450	35	-0.2	2.70
83	7	230	3	450	40	-0.2	2.70
84	7	230	3	450	45	-0.2	2.70
85	7	230	3	600	35	-0.2	2.70
86	7	230	3	600	40	-0.2	2.70
87	7	230	3	600	45	-0.2	2.70
88	7	230	3	750	35	-0.2	2.70
89	7	230	3	750	40	-0.2	2.70
90	7	230	3	750	45	-0.2	2.70
91	7	230	4	450	35	0	2.70
92	7	230	4	450	40	0	2.70
93	7	230	4	450	45	0	2.70
94	7	230	4	600	35	0	2.70
95	7	230	4	600	40	0	2.70
96	7	230	4	600	45	0	2.70
97	7	230	4	750	35	0	2.70
98	7	230	4	750	40	0	2.70
99	7	230	4	750	45	0	2.70
100	7	230	5	450	35	0.2	2.70
101	7	230	5	450	40	0.2	2.70
102	7	230	5	450	45	0.2	2.70
103	7	230	5	600	35	0.2	2.70
104	7	230	5	600	40	0.2	2.70
105	7	230	5	600	45	0.2	2.70
106	7	230	5	750	35	0.2	2.70
107	7	230	5	750	40	0.2	2.70
108	7	230	5	750	45	0.2	2.70
109	7	250	3	450	35	-0.6	2.56
110	7	250	3	450	40	-0.5	2.56
111	7	250	3	450	45	-0.5	2.56

	INPUTS					OUTPUTS	
S.No	**SPAN (m)**	**DEPTH (mm)**	**LOAD (kN/m^2)**	**COLUMN (mm)**	**GRADE OF CONCRETE**	**DEFLECTION (mm)**	**STEEL (kg/m^2)**
112	7	250	3	600	35	-0.6	2.56
113	7	250	3	600	40	-0.5	2.56
114	7	250	3	600	45	-0.5	2.56
115	7	250	3	750	35	-0.6	2.56
116	7	250	3	750	40	-0.5	2.56
117	7	250	3	750	45	-0.5	2.56
118	7	250	4	450	35	-0.4	2.56
119	7	250	4	450	40	-0.4	2.56
120	7	250	4	450	45	-0.4	2.56
121	7	250	4	600	35	-0.4	2.56
122	7	250	4	600	40	-0.4	2.56
123	7	250	4	600	45	-0.4	2.56
124	7	250	4	750	35	-0.4	2.56
125	7	250	4	750	40	-0.4	2.56
126	7	250	4	750	45	-0.4	2.56
127	7	250	5	450	35	-0.3	2.56
128	7	250	5	450	40	-0.3	2.56
129	7	250	5	450	45	-0.2	2.56
130	7	250	5	600	35	-0.3	2.56
131	7	250	5	600	40	-0.3	2.56
132	7	250	5	600	45	-0.2	2.56
133	7	250	5	750	35	-0.3	2.56
134	7	250	5	750	40	-0.2	2.56
135	7	250	5	750	45	-0.2	2.56
136	8	170	3	450	35	6.5	3.90
137	8	170	3	450	40	6.1	3.90
138	8	170	3	450	45	5.7	3.90
139	8	170	3	600	35	6.4	3.90
140	8	170	3	600	40	5.9	3.90
141	8	170	3	600	45	5.6	3.90
142	8	170	3	750	35	6.2	3.90
143	8	170	3	750	40	5.8	3.90
144	8	170	3	750	45	5.5	3.90
145	8	170	4	450	35	7.3	3.90
146	8	170	4	450	40	6.9	3.90
147	8	170	4	450	45	6.5	3.90
148	8	170	4	600	35	7.2	3.90
149	8	170	4	600	40	6.7	3.90
150	8	170	4	600	45	6.3	3.90

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
151	8	170	4	750	35	7.1	3.90
152	8	170	4	750	40	6.6	3.90
153	8	170	4	750	45	6.2	3.90
154	8	170	5	450	35	8.2	3.90
155	8	170	5	450	40	7.7	3.90
156	8	170	5	450	45	7.2	3.90
157	8	170	5	600	35	8	3.90
158	8	170	5	600	40	7.5	3.90
159	8	170	5	600	45	7.1	3.90
160	8	170	5	750	35	7.9	3.90
161	8	170	5	750	40	7.4	3.90
162	8	170	5	750	45	7	3.90
163	8	190	3	450	35	4.2	3.61
164	8	190	3	450	40	3.9	3.61
165	8	190	3	450	45	3.7	3.61
166	8	190	3	600	35	4.1	3.61
167	8	190	3	600	40	3.8	3.61
168	8	190	3	600	45	3.6	3.61
169	8	190	3	750	35	4	3.61
170	8	190	3	750	40	3.8	3.61
171	8	190	3	750	45	3.5	3.61
172	8	190	4	450	35	4.8	3.61
173	8	190	4	450	40	4.5	3.61
174	8	190	4	450	45	4.2	3.61
175	8	190	4	600	35	4.7	3.61
176	8	190	4	600	40	4.4	3.61
177	8	190	4	600	45	4.1	3.61
178	8	190	4	750	35	4.6	3.61
179	8	190	4	750	40	4.3	3.61
180	8	190	4	750	45	4.1	3.61
181	8	190	5	450	35	5.4	3.61
182	8	190	5	450	40	5	3.61
183	8	190	5	450	45	4.8	3.61
184	8	190	5	600	35	5.3	3.61
185	8	190	5	600	40	5	3.61
186	8	190	5	600	45	4.7	3.61
187	8	190	5	750	35	5.2	3.61
188	8	190	5	750	40	4.9	3.61
189	8	190	5	750	45	4.6	3.61

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
190	8	210	3	450	35	2.6	3.28
191	8	210	3	450	40	2.4	3.28
192	8	210	3	450	45	2.3	3.28
193	8	210	3	600	35	2.6	3.28
194	8	210	3	600	40	2.4	3.28
195	8	210	3	600	45	2.3	3.28
196	8	210	3	750	35	2.5	3.28
197	8	210	3	750	40	2.4	3.28
198	8	210	3	750	45	2.2	3.28
199	8	210	4	450	35	3.1	3.28
200	8	210	4	450	40	2.9	3.28
201	8	210	4	450	45	2.7	3.28
202	8	210	4	600	35	3	3.28
203	8	210	4	600	40	2.8	3.28
204	8	210	4	600	45	2.7	3.28
205	8	210	4	750	35	3	3.28
206	8	210	4	750	40	2.8	3.28
207	8	210	4	750	45	2.6	3.28
208	8	210	5	450	35	3.5	3.33
209	8	210	5	450	40	3.3	3.33
210	8	210	5	450	45	3.1	3.33
211	8	210	5	600	35	3.5	3.33
212	8	210	5	600	40	3.2	3.33
213	8	210	5	600	45	3.1	3.33
214	8	210	5	750	35	3.4	3.33
215	8	210	5	750	40	3.2	3.33
216	8	210	5	750	45	3	3.33
217	8	230	3	450	35	1.5	2.96
218	8	230	3	450	40	1.4	2.96
219	8	230	3	450	45	1.3	2.96
220	8	230	3	600	35	1.5	2.96
221	8	230	3	600	40	1.4	2.96
222	8	230	3	600	45	1.3	2.96
223	8	230	3	750	35	1.5	2.96
224	8	230	3	750	40	1.4	2.96
225	8	230	3	750	45	1.3	2.96
226	8	230	4	450	35	1.9	2.96
227	8	230	4	450	40	1.8	2.96
228	8	230	4	450	45	1.7	2.96

	INPUTS					OUTPUTS	
S.No	**SPAN (m)**	**DEPTH (mm)**	**LOAD (kN/m^2)**	**COLUMN (mm)**	**GRADE OF CONCRETE**	**DEFLECTION (mm)**	**STEEL (kg/m^2)**
229	8	230	4	600	35	1.9	2.96
230	8	230	4	600	40	1.7	2.96
231	8	230	4	600	45	1.6	2.96
232	8	230	4	750	35	1.8	2.96
233	8	230	4	750	40	1.7	2.96
234	8	230	4	750	45	1.6	2.96
235	8	230	5	450	35	2.2	2.96
236	8	230	5	450	40	2.1	2.96
237	8	230	5	450	45	2	2.96
238	8	230	5	600	35	2.2	2.96
239	8	230	5	600	40	2	2.96
240	8	230	5	600	45	1.9	2.96
241	8	230	5	750	35	2.2	2.96
242	8	230	5	750	40	2	2.96
243	8	230	5	750	45	1.9	2.96
244	8	250	3	450	35	0.8	2.89
245	8	250	3	450	40	0.7	2.89
246	8	250	3	450	45	0.7	2.69
247	8	250	3	600	35	0.8	2.69
248	8	250	3	600	40	0.7	2.69
249	8	250	3	600	45	0.7	2.69
250	8	250	3	750	35	0.8	2.69
251	8	250	3	750	40	0.7	2.69
252	8	250	3	750	45	0.7	2.69
253	8	250	4	450	35	1	2.69
254	8	250	4	450	40	1	2.69
255	8	250	4	450	45	0.9	2.69
256	8	250	4	600	35	1	2.69
257	8	250	4	600	40	1	2.69
258	8	250	4	600	45	0.9	2.69
259	8	250	4	750	35	1	2.69
260	8	250	4	750	40	1	2.69
261	8	250	4	750	45	0.9	2.69
262	8	250	5	450	35	1.3	2.69
263	8	250	5	450	40	1.2	2.69
264	8	250	5	450	45	1.2	2.69
265	8	250	5	600	35	1.3	2.69
266	8	250	5	600	40	1.2	2.69
267	8	250	5	600	45	1.1	2.69

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
268	8	250	5	750	35	1.3	2.69
269	8	250	5	750	40	1.2	2.69
270	8	250	5	750	45	1.1	2.69
271	9	170	3	450	35	12.5	4.35
272	9	170	3	450	40	11.7	4.35
273	9	170	3	450	45	11	4.35
274	9	170	3	600	35	12.3	4.35
275	9	170	3	600	40	11.5	4.35
276	9	170	3	600	45	10.8	4.35
277	9	170	3	750	35	12.1	4.35
278	9	170	3	750	40	11.3	4.35
279	9	170	3	750	45	10.7	4.35
280	9	170	4	450	35	13.8	4.35
281	9	170	4	450	40	12.9	4.35
282	9	170	4	450	45	12.2	4.35
283	9	170	4	600	35	13.6	4.35
284	9	170	4	600	40	12.7	4.35
285	9	170	4	600	45	12	4.35
286	9	170	4	750	35	13.4	4.35
287	9	170	4	750	40	12.6	4.35
288	9	170	4	750	45	11.9	4.35
289	9	170	5	450	35	15.6	4.35
290	9	170	5	450	40	14.2	4.35
291	9	170	5	450	45	13.4	4.35
292	9	170	5	600	35	15	4.35
293	9	170	5	600	40	14	4.35
294	9	170	5	600	45	13.2	4.35
295	9	170	5	750	35	14.8	4.35
296	9	170	5	750	40	13.8	4.35
297	9	170	5	750	45	13	4.35
298	9	190	3	450	35	8.7	4.04
299	9	190	3	450	40	8.2	4.04
300	9	190	3	450	45	7.7	4.04
301	9	190	3	600	35	8.6	4.04
302	9	190	3	600	40	8	4.04
303	9	190	3	600	45	7.6	4.04
304	9	190	3	750	35	8.5	4.04
305	9	190	3	750	40	7.9	4.04
306	9	190	3	750	45	7.5	4.04

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
307	9	190	4	450	35	9.7	4.04
308	9	190	4	450	40	9.1	4.04
309	9	190	4	450	45	8.5	4.04
310	9	190	4	600	35	9.6	4.04
311	9	190	4	600	40	8.9	4.04
312	9	190	4	600	45	8.4	4.04
313	9	190	4	750	35	9.4	4.04
314	9	190	4	750	40	8.8	4.04
315	9	190	4	750	45	8.3	4.04
316	9	190	5	450	35	10.7	4.04
317	9	190	5	450	40	10	4.04
318	9	190	5	450	45	9.4	4.04
319	9	190	5	600	35	10.5	4.04
320	9	190	5	600	40	9.8	4.04
321	9	190	5	600	45	9.3	4.04
322	9	190	5	750	35	10.4	4.04
323	9	190	5	750	40	9.7	4.04
324	9	190	5	750	45	9.1	4.04
325	9	210	3	450	35	6.2	3.73
326	9	210	3	450	40	5.8	3.73
327	9	210	3	450	45	5.4	3.73
328	9	210	3	600	35	6.1	3.73
329	9	210	3	600	40	5.7	3.73
330	9	210	3	600	45	5.4	3.73
331	9	210	3	750	35	6	3.73
332	9	210	3	750	40	5.6	3.73
333	9	210	3	750	45	5.3	3.73
334	9	210	4	450	35	6.9	3.73
335	9	210	4	450	40	6.4	3.73
336	9	210	4	450	45	6.1	3.73
337	9	210	4	600	35	6.8	3.73
338	9	210	4	600	40	6.3	3.73
339	9	210	4	600	45	6	3.73
340	9	210	4	750	35	6.7	3.73
341	9	210	4	750	40	6.3	3.73
342	9	210	4	750	45	5.9	3.73
343	9	210	5	450	35	7.6	3.73
344	9	210	5	450	40	7.1	3.73
345	9	210	5	450	45	6.7	3.73

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
346	9	210	5	600	35	7.5	3.73
347	9	210	5	600	40	7	3.73
348	9	210	5	600	45	6.6	3.73
349	9	210	5	750	35	7.4	3.73
350	9	210	5	750	40	6.9	3.73
351	9	210	5	750	45	6.5	3.73
352	9	230	3	450	35	4.4	3.44
353	9	230	3	450	40	4.1	3.44
354	9	230	3	450	45	3.9	3.44
355	9	230	3	600	35	4.3	3.44
356	9	230	3	600	40	4	3.44
357	9	230	3	600	45	3.8	3.44
358	9	230	3	750	35	4.2	3.44
359	9	230	3	750	40	4	3.44
360	9	230	3	750	45	3.7	3.44
361	9	230	4	450	35	5	3.44
362	9	230	4	450	40	4.7	3.44
363	9	230	4	450	45	4.4	3.44
364	9	230	4	600	35	4.9	3.44
365	9	230	4	600	40	4.6	3.44
366	9	230	4	600	45	4.3	3.44
367	9	230	4	750	35	4.8	3.44
368	9	230	4	750	40	4.5	3.44
369	9	230	4	750	45	4.2	3.44
370	9	230	5	450	35	5.5	3.44
371	9	230	5	450	40	5.2	3.44
372	9	230	5	450	45	4.9	3.44
373	9	230	5	600	35	5.4	3.44
374	9	230	5	600	40	5.1	3.44
375	9	230	5	600	45	4.8	3.44
376	9	230	5	750	35	5.3	3.44
377	9	230	5	750	40	5	3.44
378	9	230	5	750	45	4.7	3.44
379	9	250	3	450	35	2.6	3.13
380	9	250	3	450	40	2.5	3.13
381	9	250	3	450	45	2.3	3.13
382	9	250	3	600	35	2.6	3.13
383	9	250	3	600	40	2.4	3.13
384	9	250	3	600	45	2.3	3.13

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
385	9	250	3	750	35	2.6	3.13
386	9	250	3	750	40	2.4	3.13
387	9	250	3	750	45	2.3	3.13
388	9	250	4	450	35	3.1	3.13
389	9	250	4	450	40	2.9	3.13
390	9	250	4	450	45	2.7	3.13
391	9	250	4	600	35	3	3.13
392	9	250	4	600	40	2.8	3.13
393	9	250	4	600	45	2.7	3.13
394	9	250	4	750	35	3	3.13
395	9	250	4	750	40	2.8	3.13
396	9	250	4	750	45	2.6	3.13
397	9	250	5	450	35	3.5	3.13
398	9	250	5	450	40	3.3	3.13
399	9	250	5	450	45	3.1	3.13
400	9	250	5	600	35	3.4	3.13
401	9	250	5	600	40	3.2	3.13
402	9	250	5	600	45	3	3.13
403	9	250	5	750	35	3.4	3.13
404	9	250	5	750	40	3.2	3.13
405	9	250	5	750	45	3	3.13
406	10	170	3	450	35	22	4.73
407	10	170	3	450	40	21.8	4.73
408	10	170	3	450	45	20.1	4.73
409	10	170	3	600	35	22.3	4.73
410	10	170	3	600	40	21.1	4.73
411	10	170	3	600	45	19.5	4.73
412	10	170	3	750	35	22.3	4.73
413	10	170	3	750	40	20.3	4.73
414	10	170	3	750	45	18.8	4.73
415	10	170	4	450	35	23.2	4.73
416	10	170	4	450	40	22.6	4.73
417	10	170	4	450	45	21.5	4.73
418	10	170	4	600	35	23.1	4.73
419	10	170	4	600	40	22.5	4.73
420	10	170	4	600	45	22.2	4.73
421	10	170	4	750	35	23.8	4.73
422	10	170	4	750	40	23.2	4.73
423	10	170	4	750	45	21.4	4.73

	INPUTS					OUTPUTS	
S.No	**SPAN (m)**	**DEPTH (mm)**	**LOAD (kN/m^2)**	**COLUMN (mm)**	**GRADE OF CONCRETE**	**DEFLECTION (mm)**	**STEEL (kg/m^2)**
424	10	170	5	450	35	23.8	4.73
425	10	170	5	450	40	23.3	4.73
426	10	170	5	450	45	22.3	4.73
427	10	170	5	600	35	23.9	4.73
428	10	170	5	600	40	23.3	4.73
429	10	170	5	600	45	22.9	4.73
430	10	170	5	750	35	24.6	4.73
431	10	170	5	750	40	23.9	4.73
432	10	170	5	750	45	23.4	4.73
433	10	190	3	450	35	15.9	4.42
434	10	190	3	450	40	14.7	4.42
435	10	190	3	450	45	13.9	4.42
436	10	190	3	600	35	15.6	4.42
437	10	190	3	600	40	14.5	4.42
438	10	190	3	600	45	13.7	4.42
439	10	190	3	750	35	15.3	4.42
440	10	190	3	750	40	14.3	4.42
441	10	190	3	750	45	13.5	4.42
442	10	190	4	450	35	18	4.42
443	10	190	4	450	40	16.5	4.42
444	10	190	4	450	45	15.3	4.42
445	10	190	4	600	35	17.5	4.42
446	10	190	4	600	40	16	4.42
447	10	190	4	600	45	15	4.42
448	10	190	4	750	35	16.9	4.42
449	10	190	4	750	40	15.7	4.42
450	10	190	4	750	45	14.8	4.42
451	10	190	5	450	35	19.8	4.42
452	10	190	5	450	40	18.5	4.42
453	10	190	5	450	45	17.1	4.42
454	10	190	5	600	35	19.7	4.42
455	10	190	5	600	40	18	4.42
456	10	190	5	600	45	16.6	4.42
457	10	190	5	750	35	19	4.42
458	10	190	5	750	40	17.4	4.42
459	10	190	5	750	45	16.1	4.42
460	10	210	3	450	35	11.7	4.22
461	10	210	3	450	40	11	4.22
462	10	210	3	450	45	10.3	4.22

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
463	10	210	3	600	35	11.5	4.22
464	10	210	3	600	40	10.8	4.22
465	10	210	3	600	45	10.2	4.22
466	10	210	3	750	35	11.3	4.22
467	10	210	3	750	40	10.6	4.22
468	10	210	3	750	45	10	4.22
469	10	210	4	450	35	12.8	4.22
470	10	210	4	450	40	12	4.22
471	10	210	4	450	45	11.3	4.22
472	10	210	4	600	35	12.6	4.22
473	10	210	4	600	40	11.8	4.22
474	10	210	4	600	45	11.2	4.22
475	10	210	4	750	35	12.4	4.22
476	10	210	4	750	40	11.6	4.22
477	10	210	4	750	45	11	4.22
478	10	210	5	450	35	13.9	4.22
479	10	210	5	450	40	13	4.22
480	10	210	5	450	45	12.3	4.22
481	10	210	5	600	35	13.7	4.22
482	10	210	5	600	40	12.9	4.22
483	10	210	5	600	45	12.1	4.22
484	10	210	5	750	35	13.5	4.22
485	10	210	5	750	40	12.6	4.22
486	10	210	5	750	45	11.9	4.22
487	10	230	3	450	35	8.2	4.04
488	10	230	3	450	40	7.7	4.04
489	10	230	3	450	45	7.3	4.04
490	10	230	3	600	35	8.1	4.04
491	10	230	3	600	40	7.6	4.04
492	10	230	3	600	45	7.2	4.04
493	10	230	3	750	35	8	4.04
494	10	230	3	750	40	7.5	4.04
495	10	230	3	750	45	7.1	4.04
496	10	230	4	450	35	9.1	4.04
497	10	230	4	450	40	8.5	4.04
498	10	230	4	450	45	8	4.04
499	10	230	4	600	35	9	4.04
500	10	230	4	600	40	8.4	4.04
501	10	230	4	600	45	7.9	4.04

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
502	10	230	4	750	35	8.8	4.04
503	10	230	4	750	40	8.3	4.04
504	10	230	4	750	45	7.8	4.04
505	10	230	5	450	35	9.9	4.04
506	10	230	5	450	40	9.3	4.04
507	10	230	5	450	45	8.7	4.04
508	10	230	5	600	35	9.8	4.04
509	10	230	5	600	40	9.2	4.04
510	10	230	5	600	45	8.7	4.04
511	10	230	5	750	35	9.7	4.04
512	10	230	5	750	40	9	4.04
513	10	230	5	750	45	8.5	4.04
514	10	250	3	450	35	6.2	3.89
515	10	250	3	450	40	5.8	3.89
516	10	250	3	450	45	5.4	3.89
517	10	250	3	600	35	6.1	3.89
518	10	250	3	600	40	5.7	3.89
519	10	250	3	600	45	5.4	3.89
520	10	250	3	750	35	6	3.89
521	10	250	3	750	40	5.6	3.89
522	10	250	3	750	45	5.3	3.89
523	10	250	4	450	35	6.1	3.89
524	10	250	4	450	40	6.4	3.89
525	10	250	4	450	45	6	3.89
526	10	250	4	600	35	6.8	3.89
527	10	250	4	600	40	6.3	3.89
528	10	250	4	600	45	6	3.89
529	10	250	4	750	35	6.7	3.89
530	10	250	4	750	40	6.2	3.89
531	10	250	4	750	45	5.9	3.89
532	10	250	5	450	35	7.5	3.89
533	10	250	5	450	40	7	3.89
534	10	250	5	450	45	6.6	3.89
535	10	250	5	600	35	7.4	3.89
536	10	250	5	600	40	6.9	3.89
537	10	250	5	600	45	6.5	3.89
538	10	250	5	750	35	7.3	3.89
539	10	250	5	750	40	6.8	3.89
540	10	250	5	750	45	6.4	3.89

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
541	11	170	3	450	35	29.6	5.19
542	11	170	3	450	40	28.7	5.19
543	11	170	3	450	45	28.1	5.19
544	11	170	3	600	35	29.7	5.19
545	11	170	3	600	40	28.8	5.19
546	11	170	3	600	45	28.2	5.19
547	11	170	3	750	35	30.5	5.19
548	11	170	3	750	40	29.5	5.19
549	11	170	3	750	45	28.8	5.19
550	11	170	4	450	35	30.8	5.19
551	11	170	4	450	40	29.8	5.19
552	11	170	4	450	45	28.5	5.19
553	11	170	4	600	35	30.8	5.19
554	11	170	4	600	40	29.9	5.19
555	11	170	4	600	45	29.1	5.19
556	11	170	4	750	35	31.6	5.19
557	11	170	4	750	40	30.5	5.19
558	11	170	4	750	45	29.7	5.19
559	11	170	5	450	35	31.9	5.19
560	11	170	5	450	40	30.9	5.19
561	11	170	5	450	45	29.5	5.19
562	11	170	5	600	35	32	5.19
563	11	170	5	600	40	30.9	5.19
564	11	170	5	600	45	30.2	5.19
565	11	170	5	750	35	32.7	5.19
566	11	170	5	750	40	31.6	5.19
567	11	170	5	750	45	30.7	5.19
568	11	190	3	450	35	24.6	4.84
569	11	190	3	450	40	23.8	4.84
570	11	190	3	450	45	23.2	4.84
571	11	190	3	600	35	25.3	4.84
572	11	190	3	600	40	24.4	4.84
573	11	190	3	600	45	23.8	4.84
574	11	190	3	750	35	25.9	4.84
575	11	190	3	750	40	25	4.84
576	11	190	3	750	45	23	4.84
577	11	190	4	450	35	25.6	4.84
578	11	190	4	450	40	24.7	4.84
579	11	190	4	450	45	24.1	4.84

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
580	11	190	4	600	35	25.7	4.84
581	11	190	4	600	40	24.8	4.84
582	11	190	4	600	45	24.7	4.84
583	11	190	4	750	35	26.3	4.84
584	11	190	4	750	40	25.3	4.84
585	11	190	4	750	45	24.7	4.84
586	11	190	5	450	35	26	4.84
587	11	190	5	450	40	25.1	4.84
588	11	190	5	450	45	24.5	4.84
589	11	190	5	600	35	26.7	4.84
590	11	190	5	600	40	25.8	4.84
591	11	190	5	600	45	25.1	4.84
592	11	190	5	750	35	27.3	4.84
593	11	190	5	750	40	26.3	4.84
594	11	190	5	750	45	25.6	4.84
595	11	210	3	450	35	19.8	4.49
596	11	210	3	450	40	18.1	4.49
597	11	210	3	450	45	16.7	4.49
598	11	210	3	600	35	19.2	4.49
599	11	210	3	600	40	17.6	4.49
600	11	210	3	600	45	16.5	4.49
601	11	210	3	750	35	18.7	4.49
602	11	210	3	750	40	17.2	4.49
603	11	210	3	750	45	16.3	4.49
604	11	210	4	450	35	21.9	4.49
605	11	210	4	450	40	20.3	4.49
606	11	210	4	450	45	18.8	4.49
607	11	210	4	600	35	21.6	4.49
608	11	210	4	600	40	19.7	4.49
609	11	210	4	600	45	18.2	4.49
610	11	210	4	750	35	21	4.49
611	11	210	4	750	40	19.2	4.49
612	11	210	4	750	45	17.7	4.49
613	11	210	5	450	35	22.3	4.49
614	11	210	5	450	40	21.5	4.49
615	11	210	5	450	45	20.9	4.49
616	11	210	5	600	35	22.4	4.49
617	11	210	5	600	40	22.1	4.49
618	11	210	5	600	45	20.3	4.49

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
619	11	210	5	750	35	22.9	4.49
620	11	210	5	750	40	21.4	4.49
621	11	210	5	750	45	19.7	4.49
622	11	230	3	450	35	14.5	4.24
623	11	230	3	450	40	13.6	4.24
624	11	230	3	450	45	12.8	4.24
625	11	230	3	600	35	14.4	4.24
626	11	230	3	600	40	13.5	4.24
627	11	230	3	600	45	12.7	4.24
628	11	230	3	750	35	14.2	4.24
629	11	230	3	750	40	13.3	4.24
630	11	230	3	750	45	12.5	4.24
631	11	230	4	450	35	16	4.24
632	11	230	4	450	40	14.8	4.24
633	11	230	4	450	45	13.9	4.24
634	11	230	4	600	35	15.6	4.24
635	11	230	4	600	40	14.6	4.24
636	11	230	4	600	45	13.8	4.24
637	11	230	4	750	35	15.4	4.24
638	11	230	4	750	40	14.4	4.24
639	11	230	4	750	45	13.6	4.24
640	11	230	5	450	35	17.8	4.24
641	11	230	5	450	40	16.2	4.24
642	11	230	5	450	45	15	4.24
643	11	230	5	600	35	17.3	4.24
644	11	230	5	600	40	15.8	4.24
645	11	230	5	600	45	14.9	4.24
646	11	230	5	750	35	16.8	4.24
647	11	230	5	750	40	15.6	4.24
648	11	230	5	750	45	14.7	4.24
649	11	250	3	450	35	11.5	4.24
650	11	250	3	450	40	10.7	4.24
651	11	250	3	450	45	10.1	4.24
652	11	250	3	600	35	11.3	4.24
653	11	250	3	600	40	10.6	4.24
654	11	250	3	600	45	10	4.24
655	11	250	3	750	35	11.2	4.24
656	11	250	3	750	40	10.4	4.24
657	11	250	3	750	45	9.8	4.24

	INPUTS					OUTPUTS	
S.No	**SPAN (m)**	**DEPTH (mm)**	**LOAD (kN/m^2)**	**COLUMN (mm)**	**GRADE OF CONCRETE**	**DEFLECTION (mm)**	**STEEL (kg/m^2)**
658	11	250	4	450	35	12.4	4.24
659	11	250	4	450	40	11.6	4.24
660	11	250	4	450	45	11	4.24
661	11	250	4	600	35	12.3	4.24
662	11	250	4	600	40	11.5	4.24
663	11	250	4	600	45	10.8	4.24
664	11	250	4	750	35	12.1	4.24
665	11	250	4	750	40	11.3	4.24
666	11	250	4	750	45	10.7	4.24
667	11	250	5	450	35	13.4	4.24
668	11	250	5	450	40	12.6	4.24
669	11	250	5	450	45	11.8	4.24
670	11	250	5	600	35	13.2	4.24
671	11	250	5	600	40	12.4	4.24
672	11	250	5	600	45	11.7	4.24
673	11	250	5	750	35	13.1	4.24
674	11	250	5	750	40	12.2	4.24
675	11	250	5	750	45	11.5	4.24
676	12	170	3	450	35	38.7	5.86
677	12	170	3	450	40	37.2	5.86
678	12	170	3	450	45	36.2	5.86
679	12	170	3	600	35	39.5	5.86
680	12	170	3	600	40	38	5.86
681	12	170	3	600	45	36.9	5.86
682	12	170	3	750	35	40.3	5.86
683	12	170	3	750	40	38.7	5.86
684	12	170	3	750	45	37.4	5.86
685	12	170	4	450	45	36.9	5.86
686	12	170	4	600	40	38.7	5.86
687	12	170	4	600	45	37.5	5.86
688	12	170	4	750	35	41	5.86
689	12	170	4	750	40	39.3	5.86
690	12	170	4	750	45	38.1	5.86
691	12	170	5	750	45	40.6	5.86
692	12	190	3	450	35	30.9	5.49
693	12	190	3	450	40	30.8	5.49
694	12	190	3	450	45	29.8	5.49
695	12	190	3	600	35	32.3	5.49
696	12	190	3	600	40	31	5.49

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
697	12	190	3	600	45	30	5.49
698	12	190	3	750	35	33	5.49
699	12	190	3	750	40	31.6	5.49
700	12	190	3	750	45	30.6	5.49
701	12	190	4	450	35	31.9	5.49
702	12	190	4	450	40	31.2	5.49
703	12	190	4	450	45	30.2	5.19
704	12	190	4	600	35	33.9	5.19
705	12	190	4	600	40	32.5	5.19
706	12	190	4	600	45	31.4	5.19
707	12	190	4	750	35	34.5	5.19
708	12	190	4	750	40	33	5.19
709	12	190	4	750	45	31.9	5.19
710	12	190	5	450	35	33.5	5.19
711	12	190	5	450	40	32.1	5.19
712	12	190	5	450	45	31.1	5.19
713	12	190	5	600	35	33.7	5.19
714	12	190	5	600	40	34	5.19
715	12	190	5	600	45	32.8	5.19
716	12	190	5	750	35	34.3	5.19
717	12	190	5	750	40	34.5	5.19
718	12	190	5	750	45	33.3	5.19
719	12	210	3	450	35	27.1	4.96
720	12	210	3	450	40	26	4.96
721	12	210	3	450	45	25.2	4.96
722	12	210	3	600	35	27.7	4.96
723	12	210	3	600	40	26.6	4.96
724	12	210	3	600	45	25.7	4.96
725	12	210	3	750	35	27.4	4.96
726	12	210	3	750	40	27	4.96
727	12	210	3	750	45	26.1	4.96
728	12	210	4	450	35	28	4.96
729	12	210	4	450	40	26.8	4.96
730	12	210	4	450	45	25.9	4.66
731	12	210	4	600	35	28.6	4.66
732	12	210	4	600	40	27.3	4.66
733	12	210	4	600	45	26.4	4.66
734	12	210	4	750	35	29.1	4.66
735	12	210	4	750	40	27.8	4.66

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
736	12	210	4	750	45	26.8	4.66
737	12	210	5	450	35	28.9	4.66
738	12	210	5	450	40	27.6	4.66
739	12	210	5	450	45	26.7	4.66
740	12	210	5	600	35	29.4	4.66
741	12	210	5	600	40	28.1	4.66
742	12	210	5	600	45	27.1	4.66
743	12	210	5	750	35	29.9	4.66
744	12	210	5	750	40	28.5	4.66
745	12	210	5	750	45	27.5	4.66
746	12	230	3	450	35	22.9	4.56
747	12	230	3	450	40	22.3	4.56
748	12	230	3	450	45	21.3	4.56
749	12	230	3	600	35	23.4	4.56
750	12	230	3	600	40	22.4	4.56
751	12	230	3	600	45	20.7	4.56
752	12	230	3	750	35	23.9	4.56
753	12	230	3	750	40	21.8	4.56
754	12	230	3	750	45	20.1	4.56
755	12	230	4	450	35	23.7	4.56
756	12	230	4	450	40	23.4	4.56
757	12	230	4	450	45	22.6	4.51
758	12	230	4	600	35	24.2	4.51
759	12	230	4	600	40	23.8	4.51
760	12	230	4	600	45	23	4.51
761	12	230	4	750	35	24.7	4.51
762	12	230	4	750	40	24.2	4.51
763	12	230	4	750	45	22.3	4.51
764	12	230	5	450	35	24.9	4.51
765	12	230	5	450	40	23.8	4.51
766	12	230	5	450	45	23	4.51
767	12	230	5	600	35	25.4	4.51
768	12	230	5	600	40	24.2	4.51
769	12	230	5	600	45	23.3	4.51
770	12	230	5	750	35	25.8	4.51
771	12	230	5	750	40	24.6	4.51
772	12	230	5	750	45	23.6	4.51
773	12	250	3	450	35	18.3	4.51
774	12	250	3	450	40	16.7	4.51

	INPUTS					OUTPUTS	
S.No	SPAN (m)	DEPTH (mm)	LOAD (kN/m^2)	COLUMN (mm)	GRADE OF CONCRETE	DEFLECTION (mm)	STEEL (kg/m^2)
775	12	250	3	450	45	15.6	4.51
776	12	250	3	600	35	17.8	4.51
777	12	250	3	600	40	16.4	4.51
778	12	250	3	600	45	15.4	4.51
779	12	250	3	750	35	17.4	4.51
780	12	250	3	750	40	16.2	4.51
781	12	250	3	750	45	15.3	4.51
782	12	250	4	450	35	20.3	4.51
783	12	250	4	450	40	18.5	4.51
784	12	250	4	450	45	17.1	4.41
785	12	250	4	600	35	19.8	4.41
786	12	250	4	600	40	18.1	4.41
787	12	250	4	600	45	16.7	4.41
788	12	250	4	750	35	19.3	4.41
789	12	250	4	750	40	17.6	4.41
790	12	250	4	750	45	16.5	4.41
791	12	250	5	450	35	21.2	4.41
792	12	250	5	450	40	20.4	4.41
793	12	250	5	450	45	18.9	4.41
794	12	250	5	600	35	21.6	4.41
795	12	250	5	600	40	19.9	4.41
796	12	250	5	600	45	18.4	4.41
797	12	250	5	750	35	21.2	4.41
798	12	250	5	750	40	19.4	4.41
799	12	250	5	750	45	17.9	4.41

Annexure II:

Out of the 810 configurations of PT slabs, 11 slab configurations failed in design. These slab configurations are listed below:

SLAB CONFIGURATIONS FAILING IN DESIGN					
S.No	**SPAN (m)**	**DEPTH (mm)**	**LOAD (kN/m^2)**	**COLUMN (mm)**	**CONCRETE**
1	12	170	4	450	35
2	12	170	4	450	40
3	12	170	4	600	35
4	12	170	5	450	35
5	12	170	5	450	40
6	12	170	5	450	45
7	12	170	5	600	35
8	12	170	5	600	40
9	12	170	5	600	45
10	12	170	5	750	35
11	12	170	5	750	40

Annexure III:

Comparative results of data as obtained from ADAPT PT software and as calculated manually:

Data generated using ADAPT								Manual Design Data	
S.No	**SPAN (m)**	**DEPTH (mm)**	**LOAD (kN/m^2)**	**COLUMN (mm)**	**CONCRETE**	**DEFLECTION (mm)**	**STEEL (kg/m^2)**	**DEFLECTION (mm)**	**STEEL (kg/m^2)**
8	7	170	3	750	40	2.3	3.250	3.78	2.657
90	7	230	3	750	45	-0.2	2.703	1.73	2.214
110	7	250	3	450	40	-0.5	2.555	1.58	2.214
140	8	170	3	600	40	5.9	3.895	6.66	3.100
150	8	170	4	600	45	6.3	3.895	7.17	3.100
260	8	250	4	750	40	1	2.686	2.90	2.713
280	9	170	4	450	35	13.8	4.347	13.359	3.789
310	9	190	4	600	35	9.6	4.044	9.967	3.444
350	9	210	5	750	40	6.9	3.734	7.963	3.444
460	10	210	3	450	35	11.7	4.220	10.803	4.030
480	10	210	5	450	45	12.3	4.220	11.897	4.030
500	10	230	4	600	40	8.4	4.041	8.970	4.030
560	11	170	5	450	40	30.9	5.188	31.626	5.636
650	11	250	3	450	40	10.7	4.240	9.857	4.509
670	11	250	5	600	35	13.2	4.240	12.702	4.509
690	12	170	4	750	45	38.1	5.858	36.766	6.717
740	12	210	5	600	35	29.4	4.660	27.832	5.683

Annexure IV: Design of Post Tensioned Slab:

A three way continuous PT slab, with drop panels has been considered in this design example. The deflection and requirement of PT-steel in slab having span 9 m and thickness 190 mm have been worked out in the following design example of post tensioned slab. Live load of 3 kPa is considered in excess of its self-weight. The slab is considered to be supported on columns of 3 m height having square cross section of 450 x 450 mm^2. Grade of concrete is taken as M45.

Input parameters:

1. Span (S)	:	9 m
2. Slab depth (S_d)	:	190 mm
3. Live load (L_l)	:	3 kN/m^2
4. Column size (C_s)	:	450 mm x 450 mm
5. Concrete grade	:	M45

Constants:

6. Column height (C_h)	:	3 m
7. Elastic modulus of prestressing steel (E_{ps})	:	195000 N/mm^2
8. Characteristic strength of PT steel (f_{yps})	:	1840 N/mm^2
9. Unit weight of concrete (C_w)	:	24000 N/m^3
10. Diameter of strands (IS:6006-1983)	:	12.7 mm
11. Area of strands (A_s) (IS:6006-1983)	:	100 mm^2
12. Cover provided	:	25 mm

Design calculation:

Drop Size, (D_l) = S/6 = 9/6 = 1.5 m

Drop thickness (D_t) = S_d/4 = 190/4 = 0.0475 m

Self-weight of slab (W_S) = S_d x C_w = $\frac{190 \times 24}{1000}$ = 4.56 kN/m^2

Total load $(W_t) = W_S + L_l = 4.56 + 3 = 7.56\ kN/m^2$

(*Note: Weight of drop panels has not been considered in load balancing)

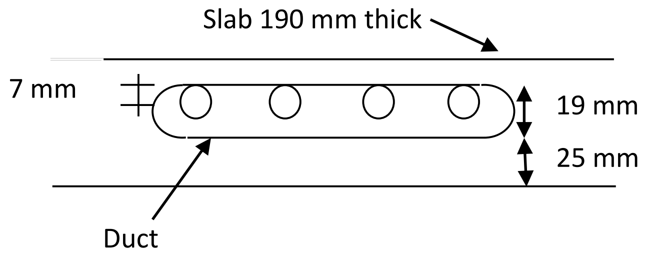


Fig: Duct parameters

Maximum depth upto the centre of gravity of strands:

$d_{cg} = 190 – 25 – (19 – 7) = 153$ mm

Maximum cable drape in exterior span:

$h_e = \{190/2 – 25 – (19 – 7)\}/2 + \{190/2 – 25 – (19 – 7)\} = 87$ mm

Maximum cable drape in interior span:

$h_i = 2 \times \{190/2 – 25 – (19 – 7)\} = 116$ mm

Jacking force assumed in a strand:

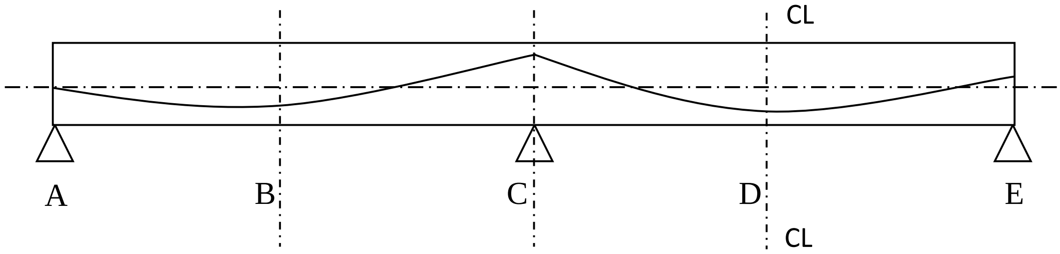
$P_j = 0.85\ f_{yps}.A_s = (0.85 \times 1840 \times 100)/1000 = 156.4$ kN

Slope of prestressing line in exterior span:

$\alpha/2 = 8\ P_j \times e \times L^2 = 8 \times 156.4 \times \{190/2 – 25 – (19 – 7)\}/9^2 = 0.896$

Length of beam affected by draw in:

L_{di} = sqrt(2 x Eps x Ap x $\delta/\ \alpha$) = sqrt(195000 x 100 x 6/0.896) = 11.428 m



Slope at A = 1.25 (4e/L)(1-2x/L)

Slope at C = 1.75 (4e/L)(1-2x/L)

α_A = 0

α_B = Slope at B - Slope at A

α_C = Slope at C - Slope at A

	Slope	**Alpha**	**L_{pa}**	**P_a/P_j**	**P_i/P_j**
A	-0.04056	0.00000	0	1.000	0.869
B	0.01811	0.03867	4.5	0.978	0.899
C	0.05678	0.07733	9	0.946	0.919
	-0.05156	0.18567	9		
D	0.00000	0.23722	13.5	0.913	0.913
E	0.05156	0.28878	18	0.882	0.882
	-0.05156	0.39189	18		

In the above given table,

L_{pa} = distance along a tendon from the jack

P_a = prestressing force

P_i = prestressing force immediately after transfer

P_j = prestressing force at the jack before transfer

Loss of force at the jack due to slip at A:

$\delta P = \alpha L_{di}$ = 2 x 0.896 x11.428 = 20.48 kN

= 0.131 P_j

Similarly, draw in loss at B = 12.41 kN

= 0.079 P_j

and draw in loss at C = 4.35 kN

= 0.028 P_j

Effective prestress per meter in exterior span:

P_e (ext) = $W_sS^2/(8 \times h_e)$ = 4.56 x 9 x 9 /(8 x 0.087) = 530.69 kN/m

Effective prestress per meter in interior span:

P_e (int) = $W_sS^2/(8 \times h_i)$ = 4.56 x 9 x 9 /(8 x 0.116) = 398.02 kN/m

Force required at jack in exterior span (assuming 15% average time dependent losses):

P_j (ext) = P_e (ext)/ (0.85 x 0.899) = 530 / (0.85 x 0.899) = 693.78 kN/m

Force required at jack in interior span (assuming 15% average time dependent losses):

P_j (int) = P_e (int)/ (0.85 x 0.913) = 530 / (0.85 x 0.913) = 512.66 kN/m

Total jacking force:

F_j = 693.78 x 9 = 6244.02 kN

Total area of prestressing steel required:

A_{pt} = $F_j/(0.85f_{yps})$ = 6244.02/(0.85 x 1840) = 3992.34 mm^2

Total number of ducts required (one duct consists of 4 strands):

N_d = A_{pt}/Area of Duct = 3992.34/400 = 10 ducts

or 40 tendons

Weight of each tendon per meter (As per IS: 6006-1983) = 0.775 kgs

Weight of all the tendons = (40 x 0.775 x 9)/(9 x 9) = **3.44 kgs/m²**

<u>Calculation for deflection:</u>

1. Moment of inertia at the column:

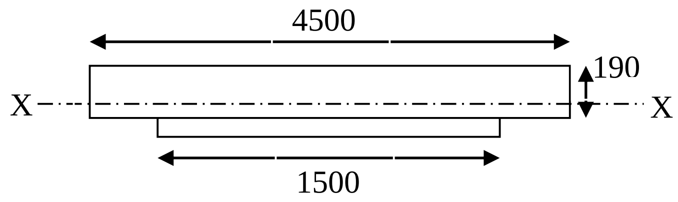


Fig : Column strip for negative moment

Area of this strip = (4.500 x 0.190) + (1.500 x 0.0475) = 0.926 m^2

Centre of gravity from bottom for this system is calculated as follows:

Z_1 = $(A_1X_1 + A_2X_2)/(A_1 + A_2)$

Where,

A_1 is the area of slab section as shown in the figure above

A_2 is the area of drop panel section

X_1 is the distance of CG of slab section from bottom of drop panel

X_2 is the distance of CG of drop panel from the bottom

Z_1 = {(4.500 x 0.190)(0.0475+0.190/2) + (1.500 x 0.0475)(0.0475/2)}/0.926

= 0.1334 m^3

Moment of inertia of this system is given by (using parallel axis theorem):

I_{xx1} = $I_{cg} + Ah^2$

Where,

I_{xx1} is the moment of inertia at an axis at the CG of the system

I_{cg} is the moment of inertia at centre of gravity of the object

A is the area of the object

h is the distance of CG of the object from the xx axis

I_{xx1} = $(4.5 \times 0.19^3/12) + \{(4.5 \times 0.19) \times (0.0475+0.190/2-0.1334)^2\}$

+

$(1.5 \times 0.0475^3/12) + \{(1.5 \times 0.0475) \times (0.1334-0.0475/2)^2\}$

= $0.0035\ m^4$

2. Moment of inertia of slab between adjacent column:

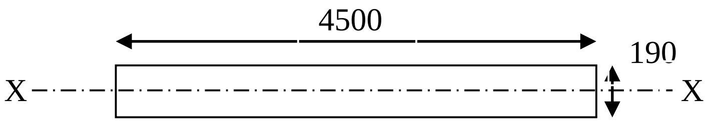


Fig : Column strip for positive moment

Centre of gravity for this system is calculated as follows:

Z_2 = (Slab thickness/2)

= 190/2 = 95 mm

Moment of inertia of this system is given by:

I_{xx2} = $bd^3/12$

Where,

I_{xx2} is the moment of inertia at an axis at the CG of the system

b is width considered of slab section

d is the depth considered of slab section

I_{xx2} = $4.5 \times 0.19^3/12$ = $0.00257\ m^4$

Total moment contribution from the column:

I_{col} = $(0.3\ I_{xx1} + 0.7\ I_{xx2})$

= $\{(0.7 \times 2.57 \times 10^9) + (0.3 \times 3.5 \times 10^9)\}$ = $2.85 \times 10^9\ mm^4$

The net moment of inertia at the mid span will be given by:

I_{ms} = $I_{col} + I_{xx2}$

= $\{(2.85 \times 10^9) + (2.57 \times 10^9)\}$ = $5.42 \times 10^9\ mm^4$

The maximum average deflection is given by:

$$\delta_{avg} = \beta \times \frac{w_u L_e^4}{E_c\, \mathrm{I}_{ms}}$$

Where,

δ_{avg} is the maximum average deflection

β is the slab deflection coefficient

w_u is the ultimate load

L_e is the effective length

δ_{avg} = $\{(2.6/384) \times (9 \times 4.56/2) \times (9 - 0.25 \times 0.45)^4\} / \{2 \times 33541 \times 5.426 \times 10^9\}$

= 7.90 mm

This design of post tensioned slab has been done on the guidelines and examples given in a book, “Design of Prestressed Concrete” by R.I Gilbert and N.C. Mickleborough. Value of β is taken as (2.6/384) from the same book for post tensioned slabs.

<u>Annexure V(A)</u>: Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using Resubstitution validation technique.

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
1	NET 1	MODEL 1_1L_RP_Tan	5	-	0.237103	0.005648	0.22185	0.00555
2		MODEL 2_1L_RP_Tan	10	-	0.120606	0.003622	0.12026	0.00360
3		MODEL 3_1L_RP_Tan	15	-	0.094	0.003634	0.09623	0.00357
4		MODEL 4_1L_RP_Tan	20	-	0.12697	0.003756	0.12487	0.00365
5	NET 2	MODEL 1_1L_RP_Log	5	-	0.227249	0.004517	0.23017	0.00432
6		MODEL 2_1L_RP_Log	10	-	0.079978	0.002408	0.08104	0.00248
7		MODEL 3_1L_RP_Log	15	-	0.105336	0.003312	0.12585	0.00320
8		MODEL 4_1L_RP_Log	20	-	0.09456	0.002805	0.09499	0.00286
9	NET 3	MODEL 1_1L_LM_Tan	5	-	0.092123	0.003107	0.09209	0.00310
10		MODEL 2_1L_LM_Tan	10	-	0.038121	0.001176	0.03810	0.00117
11		MODEL 3_1L_LM_Tan	15	-	0.034847	0.000711	0.03458	0.00071
12		MODEL 4_1L_LM_Tan	20	-	0.026107	0.000326	0.02620	**0.00033**
13	NET 4	MODEL 1_1L_LM_Log	5	-	0.102758	0.003137	0.10265	0.00310
14		MODEL 2_1L_LM_Log	10	-	0.042335	0.001171	0.04265	0.00117
15		MODEL 3_1L_LM_Log	15	-	0.030038	0.000608	0.03047	0.00059
16		MODEL 4_1L_LM_Log	20	-	0.024548	0.000415	**0.02484**	0.00041

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
17	NET 5	MODEL 1_2L_LM_Tan_Tan	5	5	0.05364	0.001216	0.05286	0.00120
18		MODEL 2_2L_LM_Tan_Tan	7	7	0.035403	0.000747	0.03496	0.00074
19		MODEL 3_2L_LM_Tan_Tan	9	9	0.021235	0.000268	0.02100	**0.00025**
20	NET 6	MODEL 1_2L_LM_Log_Log	5	5	0.05622	0.001352	0.05587	0.00135
21		MODEL 2_2L_LM_Log_Log	7	7	0.03777	0.000612	0.03822	0.00061
22		MODEL 3_2L_LM_Log_Log	9	9	0.024344	0.000285	0.02421	0.00029
23	NET 7	MODEL 1_2L_LM_Tan_Log	5	5	0.052713	0.001336	0.02368	0.00133
24		MODEL 2_2L_LM_Tan_Log	7	7	0.042632	0.000567	0.04224	0.00056
25		MODEL 3_2L_LM_Tan_Log	9	9	0.020061	0.000357	**0.01998**	0.00035
26	NET 8	MODEL 1_2L_LM_Log_Tan	5	5	0.056362	0.001106	0.05504	0.00111
27		MODEL 2_2L_LM_Log_Tan	7	7	0.027164	0.000562	0.02835	0.00053
28		MODEL 3_2L_LM_Log_Tan	9	9	0.019243	0.000355	0.02008	0.00034
29	NET 9	MODEL 1_2L_RP_Tan_Tan	5	5	0.173466	0.004337	0.16837	0.00429
30		MODEL 2_2L_RP_Tan_Tan	7	7	0.109049	0.002754	0.09977	0.00275
31		MODEL 3_2L_RP_Tan_Tan	9	9	0.099832	0.002531	0.09755	0.00253
32	NET 10	MODEL 1_2L_RP_Log_Log	5	5	0.165351	0.004038	0.16638	0.00394
33		MODEL 2_2L_RP_Log_Log	7	7	0.100711	0.002998	0.09979	0.00305
34		MODEL 3_2L_RP_Log_Log	9	9	0.097603	0.001849	0.09522	0.00185

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
35	NET 11	MODEL 1_2L_RP_Tan_Log	5	5	0.144699	0.003813	0.14208	0.00370
36		MODEL 2_2L_RP_Tan_Log	7	7	0.103581	0.002632	0.10057	0.00231
37		MODEL 3_2L_RP_Tan_Log	9	9	0.092976	0.002019	0.09115	0.00198
38	NET 12	MODEL 1_2L_RP_Log_Tan	5	5	0.206116	0.004056	0.18037	0.00401
39		MODEL 2_2L_RP_Log_Tan	7	7	0.150719	0.001611	0.15177	0.00162
40		MODEL 3_2L_RP_Log_Tan	9	9	0.054006	0.002012	0.05355	0.00200
Min MSE Deflection			**0.01998**	**NET 7 : MODEL 3_2L_LM_Tan_Log**				
Min MSE Weight			**0.00025**	**NET 5 : MODEL 3_2L_LM_Tan_Tan**				

<u>Annexure V(B)</u>: Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using Holdout validation technique.

S.No.	**Networks**	**Models**	**No. of Neurons**		**Training MSE**		**Testing MSE**	
					Deflection	**Weight**	**Deflection**	**Weight**
1	NET 1	MODEL 1_1L_RP_Tan	5	-	0.175144	0.00475	0.16491	0.00436
2		MODEL 2_1L_RP_Tan	10	-	0.114983	0.003259	0.10547	0.00306
3		MODEL 3_1L_RP_Tan	15	-	0.118429	0.003306	0.08032	0.00295
4		MODEL 4_1L_RP_Tan	20	-	0.078127	0.003014	0.07503	0.00278
5	NET 2	MODEL 1_1L_RP_Log	5	-	0.20237	0.004429	0.18547	0.00434
6		MODEL 2_1L_RP_Log	10	-	0.084478	0.003077	0.07844	0.00285
7		MODEL 3_1L_RP_Log	15	-	0.09054	0.002586	0.07003	0.00228
8		MODEL 4_1L_RP_Log	20	-	0.079378	0.003129	0.07065	0.00296
9	NET 3	MODEL 1_1L_LM_Tan	5	-	0.088241	0.003138	0.08953	0.00309
10		MODEL 2_1L_LM_Tan	10	-	0.036993	0.001078	0.03479	0.00101
11		MODEL 3_1L_LM_Tan	15	-	0.035925	0.000621	0.03275	0.00061
12		MODEL 4_1L_LM_Tan	20	-	0.023858	0.000446	**0.02048**	0.00042
13	NET 4	MODEL 1_1L_LM_Log	5	-	0.099714	0.002555	0.09164	0.00238
14		MODEL 2_1L_LM_Log	10	-	0.042144	0.001136	0.04255	0.00113
15		MODEL 3_1L_LM_Log	15	-	0.040863	0.000769	0.03854	0.00075
16		MODEL 4_1L_LM_Log	20	-	0.023836	0.000421	0.02146	**0.00042**

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
17	NET 5	MODEL 1_2L_LM_Tan_Tan	5	5	0.050761	0.001225	0.04835	0.00122
18		MODEL 2_2L_LM_Tan_Tan	7	7	0.028054	0.00056	0.02523	0.00054
19		MODEL 3_2L_LM_Tan_Tan	9	9	0.020089	0.000239	0.01845	0.00022
20	NET 6	MODEL 1_2L_LM_Log_Log	5	5	0.050325	0.001413	0.04884	0.00129
21		MODEL 2_2L_LM_Log_Log	7	7	0.022847	0.000626	0.02199	0.00060
22		MODEL 3_2L_LM_Log_Log	9	9	0.022398	0.000239	0.02002	**0.00021**
23	NET 7	MODEL 1_2L_LM_Tan_Log	5	5	0.054876	0.001231	0.05131	0.00104
24		MODEL 2_2L_LM_Tan_Log	7	7	0.027648	0.000616	0.02350	0.00060
25		MODEL 3_2L_LM_Tan_Log	9	9	0.019104	0.000318	**0.01599**	0.00030
26	NET 8	MODEL 1_2L_LM_Log_Tan	5	5	0.059468	0.001297	0.05584	0.00108
27		MODEL 2_2L_LM_Log_Tan	7	7	0.027355	0.00065	0.02514	0.00062
28		MODEL 3_2L_LM_Log_Tan	9	9	0.022275	0.000375	0.02208	0.00036
29	NET 9	MODEL 1_2L_RP_Tan_Tan	5	5	0.121944	0.004381	0.10652	0.00428
30		MODEL 2_2L_RP_Tan_Tan	7	7	0.08643	0.003203	0.07453	0.00315
31		MODEL 3_2L_RP_Tan_Tan	9	9	0.061684	0.002348	0.05987	0.00205
32	NET 10	MODEL 1_2L_RP_Log_Log	5	5	0.165973	0.003528	0.14542	0.00349
33		MODEL 2_2L_RP_Log_Log	7	7	0.104874	0.002618	0.10541	0.00242
34		MODEL 3_2L_RP_Log_Log	9	9	0.068175	0.002297	0.06529	0.00218

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
35	NET 11	MODEL 1_2L_RP_Tan_Log	5	5	0.127459	0.003159	0.11537	0.00339
36		MODEL 2_2L_RP_Tan_Log	7	7	0.101942	0.002421	0.08747	0.00225
37		MODEL 3_2L_RP_Tan_Log	9	9	0.063474	0.00209	0.06208	0.00201
38	NET 12	MODEL 1_2L_RP_Log_Tan	5	5	0.185582	0.003524	0.20475	0.00388
39		MODEL 2_2L_RP_Log_Tan	7	7	0.078308	0.002883	0.08325	0.00306
40		MODEL 3_2L_RP_Log_Tan	9	9	0.065077	0.002227	0.08008	0.00239
Min MSE Deflection			**0.01599**		**NET 7 : MODEL 3_2L_LM_Tan_Log**			
Min MSE Weight			**0.00021**		**NET 6 : MODEL 3_2L_LM_Log_Log**			

<u>Annexure V(C)</u>: Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using Three way data split validation technique.

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
1	NET 1	MODEL 1_1L_RP_Tan	5	-	0.301662	0.005996	0.23423	0.00586
2		MODEL 2_1L_RP_Tan	10	-	0.299303	0.004522	0.33651	0.00557
3		MODEL 3_1L_RP_Tan	15	-	0.270045	0.005168	0.28537	0.00598
4		MODEL 4_1L_RP_Tan	20	-	0.256034	0.00586	0.25987	0.00487
5	NET 2	MODEL 1_1L_RP_Log	5	-	0.281608	0.005405	0.26108	0.00446
6		MODEL 2_1L_RP_Log	10	-	0.224304	0.004548	0.19825	0.00418
7		MODEL 3_1L_RP_Log	15	-	0.200884	0.005722	0.20054	0.00506
8		MODEL 4_1L_RP_Log	20	-	0.212521	0.005536	0.19683	0.00471
9	NET 3	MODEL 1_1L_LM_Tan	5	-	0.190089	0.003557	0.15359	0.00275
10		MODEL 2_1L_LM_Tan	10	-	0.085004	0.00272	0.08367	0.00255
11		MODEL 3_1L_LM_Tan	15	-	0.063537	0.003188	**0.06197**	0.00221
12	NET 4	MODEL 1_1L_LM_Log	5	-	0.088706	0.003015	0.15350	0.00276
13		MODEL 2_1L_LM_Log	10	-	0.099494	0.003751	0.09727	**0.00201**
14		MODEL 3_1L_LM_Log	15	-	0.06729	0.0033	0.06478	0.00333

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
15	NET 5	MODEL 1_2L_LM_Tan_Tan	5	5	0.078649	0.003298	0.08456	0.00350
16		MODEL 2_2L_LM_Tan_Tan	7	7	0.056794	0.002897	**0.05384**	**0.00178**
17	NET 6	MODEL 1_2L_LM_Log_Log	5	5	0.084078	0.003646	0.07790	0.00288
18		MODEL 2_2L_LM_Log_Log	7	7	0.062613	0.003321	0.06010	0.00212
19	NET 7	MODEL 1_2L_LM_Tan_Log	5	5	0.083066	0.003199	0.10754	0.00484
20		MODEL 2_2L_LM_Tan_Log	7	7	0.065091	0.003486	0.05458	0.00343
21	NET 8	MODEL 1_2L_LM_Log_Tan	5	5	0.102631	0.003158	0.10291	0.00255
22		MODEL 2_2L_LM_Log_Tan	7	7	0.063083	0.003171	0.05542	0.00239
23	NET 9	MODEL 1_2L_RP_Tan_Tan	5	5	0.32596	0.00492	0.24660	0.00467
24		MODEL 2_2L_RP_Tan_Tan	7	7	0.166411	0.005044	0.13966	0.00504
25		MODEL 3_2L_RP_Tan_Tan	9	9	0.121405	0.004752	0.19758	0.00334
26	NET 10	MODEL 1_2L_RP_Log_Log	5	5	0.213813	0.004838	0.20778	0.00411
27		MODEL 2_2L_RP_Log_Log	7	7	0.157904	0.003887	0.14798	0.00356
28		MODEL 3_2L_RP_Log_Log	9	9	0.157474	0.00436	0.14759	0.00465
29	NET 11	MODEL 1_2L_RP_Tan_Log	5	5	0.190651	0.004595	0.15749	0.00372
30		MODEL 2_2L_RP_Tan_Log	7	7	0.196942	0.004453	0.17147	0.00382
31		MODEL 3_2L_RP_Tan_Log	9	9	0.186032	0.004541	0.19230	0.00288

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
32	NET 12	MODEL 1_2L_RP_Log_Tan	5	5	0.199396	0.004653	0.19246	0.00418
33		MODEL 2_2L_RP_Log_Tan	7	7	0.202778	0.004288	0.22366	0.00334
34		MODEL 3_2L_RP_Log_Tan	9	9	0.155562	0.004253	0.18203	0.00336
Min MSE Deflection			**0.05384**		**NET 5 : MODEL 2_2L_LM_Tan_Tan**			
Min MSE Weight			**0.00178**		**NET 5 : MODEL 2_2L_LM_Tan_Tan**			

<u>Annexure V(D)</u>: Testing MSE for both single and double layered neural models for deflection and weight of post tensioning steel by using k-fold cross validation technique.

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
1	NET 1	MODEL 1_1L_RP_Tan	5	-	0.163163	0.004725	0.25380	0.00490
2		MODEL 2_1L_RP_Tan	10	-	0.097527	0.003203	0.24396	0.00476
3		MODEL 3_1L_RP_Tan	15	-	0.116359	0.003757	0.21634	0.00367
4		MODEL 4_1L_RP_Tan	20	-	0.11891	0.003797	0.11986	0.00364
5	NET 2	MODEL 1_1L_RP_Log	5	-	0.226425	0.004201	0.23078	0.00408
6		MODEL 2_1L_RP_Log	10	-	0.119149	0.003228	0.20048	0.00350
7		MODEL 3_1L_RP_Log	15	-	0.10906	0.002924	0.14838	0.00319
8		MODEL 4_1L_RP_Log	20	-	0.099208	0.002977	0.14163	0.00305
9	NET 3	MODEL 1_1L_LM_Tan	5	-	0.094484	0.003082	0.12946	0.00384
10		MODEL 2_1L_LM_Tan	10	-	0.043336	0.0011	0.04546	0.00126
11		MODEL 3_1L_LM_Tan	15	-	0.038214	0.000667	0.03605	0.00062
12		MODEL 4_1L_LM_Tan	20	-	0.026158	0.000376	**0.03010**	**0.00044**
13	NET 4	MODEL 1_1L_LM_Log	5	-	0.094908	0.003087	0.10350	0.00276
14		MODEL 2_1L_LM_Log	10	-	0.055367	0.001033	0.05862	0.00113
15		MODEL 3_1L_LM_Log	15	-	0.030829	0.000744	0.03470	0.00081
16		MODEL 4_1L_LM_Log	20	-	0.027974	0.000405	0.03022	0.00051

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
17	NET 5	MODEL 1_2L_LM_Tan_Tan	5	5	0.044927	0.001273	0.05102	0.00103
18		MODEL 2_2L_LM_Tan_Tan	7	7	0.033434	0.000539	0.03639	0.00061
19		MODEL 3_2L_LM_Tan_Tan	9	9	0.023957	0.000424	0.02738	0.00046
20	NET 6	MODEL 1_2L_LM_Log_Log	5	5	0.06441	0.001468	0.06824	0.00158
21		MODEL 2_2L_LM_Log_Log	7	7	0.025069	0.000692	0.04034	0.00084
22		MODEL 3_2L_LM_Log_Log	9	9	0.021409	0.000318	0.02951	0.00065
23	NET 7	MODEL 1_2L_LM_Tan_Log	5	5	0.063981	0.001134	0.06541	0.00149
24		MODEL 2_2L_LM_Tan_Log	7	7	0.032415	0.000721	0.03016	0.00082
25		MODEL 3_2L_LM_Tan_Log	9	9	0.022404	0.000310	**0.02708**	**0.00040**
26	NET 8	MODEL 1_2L_LM_Log_Tan	5	5	0.054094	0.001243	0.04506	0.00161
27		MODEL 2_2L_LM_Log_Tan	7	7	0.030069	0.000473	0.04416	0.00156
28		MODEL 3_2L_LM_Log_Tan	9	9	0.025893	0.000298	0.04393	0.00141
29	NET 9	MODEL 1_2L_RP_Tan_Tan	5	5	0.139238	0.004091	0.16438	0.00482
30		MODEL 2_2L_RP_Tan_Tan	7	7	0.102367	0.003503	0.10429	0.00417
31		MODEL 3_2L_RP_Tan_Tan	9	9	0.064669	0.002748	0.09260	0.00269
32	NET 10	MODEL 1_2L_RP_Log_Log	5	5	0.203649	0.003192	0.27537	0.00158
33		MODEL 2_2L_RP_Log_Log	7	7	0.092027	0.002547	0.10162	0.00228
34		MODEL 3_2L_RP_Log_Log	9	9	0.079182	0.001931	0.08449	0.00198

S.No.	Networks	Models	No. of Neurons		Training MSE		Testing MSE	
					Deflection	Weight	Deflection	Weight
35	NET 11	MODEL 1_2L_RP_Tan_Log	5	5	0.118247	0.004443	0.16240	0.04833
36		MODEL 2_2L_RP_Tan_Log	7	7	0.090516	0.003152	0.11643	0.003125
37		MODEL 3_2L_RP_Tan_Log	9	9	0.072763	0.002487	0.09295	0.00273
38	NET 12	MODEL 1_2L_RP_Log_Tan	5	5	0.13127	0.004329	0.14925	0.00484
39		MODEL 2_2L_RP_Log_Tan	7	7	0.072747	0.003069	0.11643	0.00312
40		MODEL 3_2L_RP_Log_Tan	9	9	0.076076	0.002562	0.09295	0.00273
Min MSE Deflection		**0.02708**	**NET 7 : MODEL 3_2L_LM_Tan_Log**					
Min MSE Weight		**0.00040**	**NET 7 : MODEL 3_2L_LM_Tan_Log**					