A NEURAL NETWORK APPROACH FOR PREDICTING THE STRUCTURAL BEHAVIOR OF CONCRETE SLABS

SUSAN HENTSCHEL TULLY

A NEURAL NETWORK APPROACH FOR PREDICTING THE STRUCTURAL BEHAVIOR OF CONCRETE SLABS

BY

© SUSAN HENTSCHEL TULLY, B.SC.

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING

FACULTY OF ENGINEERING AND APPLIED SCIENCE
MEMORIAL UNIVERSITY OF NEWFOUNDLAND

JUNE, 1997

ST. JOHN'S NEWFOUNDLAND CANADA

0-612-25895-5

ABSTRACT

A NEURAL NETWORK APPROACH FOR PREDICTING THE STRUCTURAL BEHAVIOR OF CONCRETE SLABS

Reinforced concrete slabs exhibit complexities in their structural behavior due to the composite nature of the material and the multitude and variety of factors that affect such behavior. As such, current methods for the design and analysis of reinforced concrete slabs are limited in scope and are approximate at best as they must rely on the results of experimental tests, which are both costly and time-consuming to perform. The research embodied by this document investigates the use of a branch of artificial intelligence known as Neural Networks (NN) as a quick and reliable alternative to such experimental testing.

Four neural network models are developed to predict the following aspects of the overall behavior of a concrete slab: 1) load-deflection behavior; 2) crack pattern at failure; 3) concrete strain distribution; and 4) reinforcing steel strain distribution. Results from experimental tests on thirty-four full scale slabs are utilized to develop these four models, incorporating all of the parameters that govern their behavior. The rationale behind and the details involved are explained for the setup, computer implementation and selection of each optimum neural network model. Results show that the neural network technique can perform as a satisfactory alternative to experimental testing or detailed calculations to provide speedy predictions of all four aspects of the structural behavior of concrete slabs.

A comprehensive spreadsheet tool is next created to incorporate all four of the optimum neural networks. The spreadsheet uses readily available software and can be used by structural engineers for instantaneous access to the prediction of any or all of the four aspects of a concrete slab's behavior given minimal data to describe the slab and the loading conditions. This tool, combined with the results for the four neural network models, demonstrates the powerful capabilities and success of neural networks in the realm of civil and structural engineering in general and reinforced concrete design in particular. This approach could readily be expanded to include the same predictions for other structural concrete elements such as beams and shear walls.

ACKNOWLEDGMENTS

The author wishes to express her appreciation to Dr. H. Marzouk, Professor of civil engineering at Memorial University of Newfoundland and Dr. T. Hegazy, Assistant Professor of civil engineering at the University of Waterloo for their support and guidance throughout the term of this research effort. Thanks are also due to Amgad Hussein for his help in providing access to the necessary data for the work.

Finally, the author would like to express her deep gratitude to Tom, Tommy and Katie for their continuing patience, support and affection.

Contents

List of Figures

List of Tables

List of Symbols and Abbreviations

ART = Adaptive Resonance Technique neural network

b = b_o = perimeter of the slab critical section

BAM = Bi-directional Associative Memory neural network

BP = Backpropagation neural network

c = column width

d = average effective depth of the slab

d_a = diameter of aggregate

d_s = average effective slab depth

E_c = concrete modulus of elasticity

E_s = steel modulus of elasticity

f'_c = concrete compressive strength

f_{cc} = concrete compressive strength

f_{ct} = concrete tensile strength

f_y = yield strength of reinforcing steel

GRNN = General Regression Neural Network

K_a = constant = 0.13 for normal density concrete (SI units)

K_{sc} = $1.15 \times [4\pi \times \text{column area} / (\text{column perimeter})^2]^{1/2}$ (SI units)

K_s = size effect term = $(300/d)^{1/4}$ (SI units)

L = slab span

NN = Neural Network

P_{flex} = ultimate flexural load capacity

$P_{u.act}$ = actual ultimate punching load

$P_{u.NN}$ = neural network predicted ultimate punching load

$P_{u.p}$ = calculated ultimate punching load $\left(\frac{bd}{\sqrt{f_c'}}\right)$

$P_{u.f}$ = calculated ultimate flexural load

s_r = spacing of reinforcing steel

t_s = slab thickness

u_o = column perimeter

V_u = ultimate shear capacity

α_c = 4 for interior columns and 3 for edge columns

β_c = ratio of long side to short side of concentrated load or reaction area

Δ = deflection

ε = strain

ϕ_c = resistance factor for concrete

λ = factor to allow for low density concrete

v_u = nominal shear stress

ρ = reinforcing steel ratio

Σ_c = perimeter of column

Chapter 1

Introduction

1.1 Neural Networks and Reinforced Concrete Slabs

Reinforced concrete slabs are used today in a variety of applications including building floor systems, bridge decks, and offshore oil platforms. In these applications, concrete is selected over other building materials primarily due to the superior formability, durability, fire resistance and insulation capabilities of concrete. Reinforced concrete, however, is composite and non-homogeneous by nature, therefore exhibiting non-linear or inelastic behavior. Calculations to predict the structural behavior of concrete slabs are therefore simplified and approximate at best and most often are formulated from the results of experimental testing on full or reduced-scale mockups of the slabs. Such tests require expensive setups and lengthy periods of time are involved in the performance of these tests.

The structural behavior of reinforced concrete is affected by many factors such as 1) concrete properties; 2) aggregate properties; 3) reinforcement steel properties; and 4) geometric properties of the structural element. Mathematical models have been used to describe aspects of this behavior, but they fall short in considering a large number of variables simultaneously. This thesis investigates the use of Neural Networks (NN) as a preliminary alternative to mathematical modeling or experimental testing for quick prediction of the structural behavior of reinforced concrete slabs. Such predictions could be utilized by a structural engineer on a preliminary basis to determine the initial suitability of a particular slab design. Once this suitability was determined, the engineer could then proceed with further, more traditional methods of design. This will serve to illustrate 1) the simple manner by which neural networks model the impact of a set of parameters (inputs) on a set of simultaneous conclusions (outputs); and 2) the powerful learn-by-example and generalization mechanism that neural networks use to detect the hidden relationships linking the inputs to their outputs (Hegazy et al., 1996).

Neural networks are computational models that adopt a training mechanism to extract the relationships that link a set of causal input parameters to their resulting conclusions. Once neural networks are trained, they can predict the results for an unknown case (not used in training) if provided with the input parameters alone. Some characteristics of neural networks that make them potentially useful for many different types of applications are (Moselhi et al., 1992):

- Neural networks are organized within a parallel, decentralized structure rather than the serial architecture found in conventional computer algorithms. As a result, processing occurs in a rapid manner;

- They have distributed memories; neural network memories are represented by interconnection weights spread over all of the network's processing elements;
- They are fault tolerant, that is, they are still functional even after several processing elements are damaged and become defective;
- They have the ability to learn-by-example;
- They have the ability to simulate the behavior of systems with limited modeling effort; and
- They can provide speedy and reasonably accurate solutions in complex, uncertain, and subjective situations.

1.2 Research Scope and Objectives

The main purpose of this thesis is to develop a set of neural networks to predict the structural behavior of reinforced concrete slabs. The research is applicable for normal strength, high strength and lightweight concrete slabs subjected to concentrated and flexural loads.

The objectives of the proposed research can be stated as follows:

1. Identify the detailed factors that govern the structural behavior of concrete slabs.
2. Investigate the suitability of neural networks for application in the structural analysis

domain, particularly in simulating the behavior patterns of structural elements, including reinforced concrete slabs.

3. Develop, train and implement a set of neural networks to predict the structural behavior of concrete slabs.
4. Compare the results of these neural networks with the results obtained from experimental tests.
5. Develop a comprehensive spreadsheet tool for the structural analysis of reinforced concrete slabs.

1.3 Research Methodology

The research methodology is as follows:

1. Evaluate the problem by reviewing the theory and current practices in both neural networks and the prediction of reinforced concrete slab behavior. Examine the literature to identify past work, both experimental and theoretical.
2. Model the structural behavior of reinforced concrete slabs in four complementary aspects, each of which lends itself to a neural network: 1) load-deflection behavior prediction; 2) crack pattern prediction; 3) concrete strain distribution; and 4) reinforcing steel strain distribution.
3. Select an appropriate neural network software. Then, conduct a preliminary

investigation on the load-deflection neural network to determine the suitability of the neural network technique for the problem at hand. Experiment with different ways of modeling the problem to achieve the optimum results.

4. Once a suitable neural network model is selected, repeat the process for the remaining three neural networks.
5. Incorporate all four neural network models into a single spreadsheet tool to summarize the research completed.
6. Validate the concepts proposed in the research scope and objectives.

1.4 Thesis Content

Chapter 2 encompasses a literature review of the state-of-the-art efforts related to neural networks and their use in the design and analysis of reinforced concrete. Traditional models for the analysis of selected types of reinforced concrete members are first discussed. The history of neural networks and their development is then reviewed. Components of neural networks are defined, and the various neural network paradigms are briefly described. General uses of neural networks in civil and structural engineering are examined as well as their specific uses in the design of reinforced concrete.

Chapter 3 describes the development of four neural network models to describe the structural behavior of reinforced concrete slabs. The applicability of neural networks to the particular problem at hand is discussed as well as the rationale behind the selection of the backpropagation paradigm. Details regarding model design, including problem analysis and structuring for each neural network model are then discussed, along with the development of alternative models for optimal network selection.

Chapter 4 describes the computer implementation of the four neural network models. Specifics regarding data preparation and software selection are discussed. Details for training and testing each of the four neural network models are then described.

Chapter 5 discusses the results and observations for all neural network models as well as providing analysis of the results.

Chapter 6 presents a comprehensive spreadsheet tool that includes the four neural network modules for the design and analysis of reinforced concrete slabs. Development of the spreadsheet is described and two sample problems are provided to illustrate the usefulness of the tool.

Chapter 7 is the thesis conclusion and summary. Prospects for further research as an extension to the results obtained from this thesis are also discussed.

Chapter 2

Literature Review

2.1 Introduction

This chapter evaluates current research efforts in the area of neural network applications in structural engineering. Traditional methods for determining the structural behavior of reinforced concrete slabs are first reviewed for the purpose of establishing a baseline for comparison to neural network research in this area. An overall introduction to neural networks and their history is next presented, along with the various neural network types applicable to the structural engineering domain. General civil engineering applications of neural networks are then briefly surveyed. State of the art research describing the use of neural networks for the structural behavior of reinforced concrete is then reviewed to assist in the development of a specific neural network model to predict the structural behavior of concrete slabs.

2.2 Traditional Models for Predicting Concrete Structural Behavior

Traditional research efforts in concrete structural analysis that have evolved in the literature during the past few decades generally aimed at developing mathematical models to predict concrete behavior under different loading conditions. These mathematical models, however, focused generally upon determining the behavior of individual structural elements which could not be generalized to describe the behavior of other elements. Also, the models require the calculation of several equations to arrive at predictions for more than one parameter. Modeling with neural networks is much simpler because, although a neural network captures the mathematical relationships in its collection of interconnections between its nodes, no formal mathematical rules or formula are used or observable within the model (Garrett et al., 1992).

Examples of some mathematical models which are in existence in the literature for describing the structural behavior of concrete are described below. These examples have been chosen as neural networks have also been developed to model these same behaviors. These neural networks are described more fully in Section 2.3.

The shear behavior of deep beams subjected to point loads can be simulated by the strut-and-tie model (Schliach, 1980), which applies a series of equations to define the ultimate shear forces in the beam. When compared to experimental test results, however, this model is only accurate when the ratio of shear span to beam height is less than 1.04; at higher values, the model results decline rapidly because deep beam behavior no longer applies (Schliach, 1980). This same limitation applies to alternate models which exist in the literature for evaluating the shear strength of deep beams (dePaiva and Siess, 1965; Ramakrishnan and Ananthanarayanana, 1968; Smith and Vantsiotis, 1982; and Subedi, 1988). Other models must therefore be applied to predict the structural behavior of shallower beams.

The behavior of reinforced concrete framed low-rise shearwalls can be predicted with the truss model theory (Mo and Shiau, 1993). This theory again applies a series of equations (given concrete and steel material properties) to predict the concrete shear strength, the shear distortion, the steel strains and the concrete strains. Although this model and others (Galletly, 1952; Benjamin and Williams, 1957; Hsu and Mo, 1985) do a reasonably accurate job in predicting the previously described values, they, like their counterpart models for deep beams, are limited because they only apply to low-rise shear walls.

Some research is described in the literature for mathematical models which predict the punching shear behavior of reinforced concrete slabs. Several models have been developed that predict the effect of concrete strength on the punching shear capacity of concrete slabs subjected to concentrated loads for normal strength (Elstner and

Hognestad, 1956 and Moe, 1961) and high strength (Marzouk and Hussein, 1991) concrete. The Moe equation is as follows:

$$v_u = \frac{V_u}{bd} = \frac{15(1-0.075\frac{c}{d})\sqrt{f'_c}}{1+\frac{5.25bd\sqrt{f'_c}}{P_{flex}}}$$

where: v_u = nominal shear stress

V_u = ultimate shear capacity

b = perimeter of the slab critical section

d = average effective depth of the slab

c = column width

f'_c = concrete compressive strength

P_{flex} = ultimate flexural load capacity

Marzouk and Hussein (1991) propose that this equation be modified to include the cubic root of f'_c when high strength concrete is used. The Elstner and Hognestad equation is:

$$\frac{P_{u,p}}{0.875u_o df'_c} = \frac{2.298}{f'_c} + \frac{0.046}{\frac{P_{u,p}}{P_{u,f}}}$$

where: u_o = column perimeter

$P_{u,p}$ = calculated ultimate punching load $\left(\frac{bd}{\sqrt{f'_c}}\right)$

$P_{u,f}$ = calculated ultimate flexural load

All of these models exhibit reasonably accurate predictions, however, as shown, different equations apply depending upon the strength of concrete used.

Kinnunen and Nylander (1960) also conducted a theoretical analysis for axisymmetric punching shear, by solving a series of equilibrium and strain compatibility equations. This model requires computer programming to formulate a solution, and is time-consuming to solve. Regan(1980) improved upon this by proposing the following equation for the punching shear capacity:

$$V_u = K_a K_{sc} K_s (\rho f'_c)^{1/3} \bullet d(\Sigma_c + 7.85d)$$

where: K_a = constant = 0.13 for normal density concrete (SI units)

K_{sc} = 1.15 $\bullet[4\pi \bullet$column area / (column perimeter)$^2]^{1/2}$ (SI units)

K_s = size effect term = $(300/d)^{1/4}$ (SI units)

ρ = reinforcing steel ratio

Σ_c = perimeter of column

The Canadian code (CSA A23.3-94) requires that the smallest v_r resulting from the following three equations be used to determine the factored shear resistance of a concrete slab:

$$v_r = \left(1+\frac{2}{\beta_c}\right)0.2\lambda\phi_c\sqrt{f_c'}\text{; or } v_r = \left(\frac{\alpha_c d}{b_o}+0.2\right)\lambda\phi_c\sqrt{f_c'}\text{; or } v_r = 0.4\lambda\phi_c\sqrt{f_c'}$$

where: β_c = ratio of long side to short side of concentrated load or reaction area

λ = factor to allow for low density concrete

ϕ_c = resistance factor for concrete

α_c = 4 for interior columns and 3 for edge columns

b_o = perimeter of critical section for shear

It is clear that there is still a wide range of uncertainty for explaining the punching shear behavior of reinforced concrete slabs. Each experimental program undertaken has produced different models for this behavior, according to the characteristics of the particular slabs used in each experimental testing program. Neural networks could be used to detect the subtle differences between the different types of slabs, thus eliminating the initial need for lengthy calculations for each model.

In addition to models for predicting the behavior of reinforced concrete, mathematical models also exist which describe the structural behavior of plain concrete. For example, the behavior of plain concrete in biaxial compression can be described by a series of stress-strain relations (Darwin and Pecknold, 1974; Kupfer and Gerstle, 1973; and Liu et al., 1972). These equations are applied, in matrix form, to describe a constitutive relationship in terms of stresses and strains; this relationship is then used in finite element investigations of the concrete's structural behavior. All three models, when compared to experimental data, are extremely accurate in representing the stress-strain curve for the concrete. However, the equations are complex and are more easily computed with the aid of time consuming serial computer algorithms. Neural network models developed for the same application (Wu and Ghaboussi, 1992) are much simpler and easier to use.

For all of the above described mathematical models, several iterations of the following procedure were necessary (Garrett et al., 1992):

- A material was tested and its behavior observed;
- Some mathematical relationship was postulated to explain its observed behavior;
- This mathematical model was used to predict yet untested concrete design and was checked against results from experiments; and
- The mathematical model was then modified to account for behaviors observed but unexplained by the model.

Such a process can be both tedious and time-consuming until a successful model is developed. Neural networks circumvent this process entirely as the underlying rationale for explaining the behavior of the model is ignored. In addition, the ability of all of the above described mathematical models to accurately predict concrete structural behavior is limited for the following reasons:

- One parameter only is measured and relationships are accordingly interpolated;
- Modeling is complex; and
- Extensive testing on new cases is often not performed and some of the governing factors of the concrete behavior, particularly subjective criteria, might be omitted.

It is clear that mathematical models, while usually quite accurate for predicting concrete structural behavior, are limited to the extent of the specific application for which they are developed and can not always be generalized to apply to those untested conditions. In addition, mathematical models can be cumbersome and time consuming. Neural network models present the possibility for circumventing both of these problems.

2.3 Neural Networks as a Modeling Tool

2.3.1 History of Neural Networks

Neural networks were first introduced as a concept in the early 1950's after Donald Hebb, a psychologist who studied the effect of learning on the neurons in the brain, introduced a simplified training mechanism called Hebb's law (Hebb, 1949). This concept was then extended by Rosenblatt (1958) with the introduction of the perceptron training algorithm; this became the first mathematical model suitable for computer simulation. In accordance with Hebb's law, this procedure viewed biological learning as a dynamic sensory process which was readily adapted to computer modeling (Hajela and Berke, 1991). Then, in 1969, with the influential publication by Minsky and Pappert of the book, Perceptrons, all research in neural networks was essentially halted; the book showed that a single or double layer perceptron network was inadequate for real world problems (Caudill and Butler, 1990). It wasn't until the 1980's that new architectures, such as the backpropagation training algorithm (see Section 2.3.1 for a description), were introduced, and the problems raised in Minsky and Papert's findings were addressed. This gave engineers (among others) reason to explore neural networks as a fast, simple alternative to mathematical modeling or experimental test setups.

2.3.2 Neural Network Basics

Neural networks are types of information processing systems whose architectures are inspired by the structure of biological neural systems (Caudill and Butler, 1990). Unlike traditional computer programming, which accepts and processes information in a digital and serial manner, neural networks actually store data among the individual neurons of the network; this data is then processed in a parallel manner. Neural networks do not contain algorithmic instructions for processing data. Rather, these models are trained to extract the relationships that link a set of causal input parameters to their resulting conclusions.

Each network is composed of three basic components as illustrated in Figure 2.1: 1) input neurons or processing elements, which represent the input for the problem, 2) connecting "axons," which connect input and output neurons and represent the connection weights that associate the input to the output, and 3) output neurons or processing elements, which represent the output for the problem. Neural networks can be composed of a single layer or many layers, according to the complexity of the architecture of the network Multi-layer neural networks may contain one or more middle layers. These middle or "hidden" layers (see Figure 2.1) consist of neurons with no direct connection to either the input or the output of the network; rather, they are used to further refine training by adjusting the connection weights for the network. These connection weights are applied

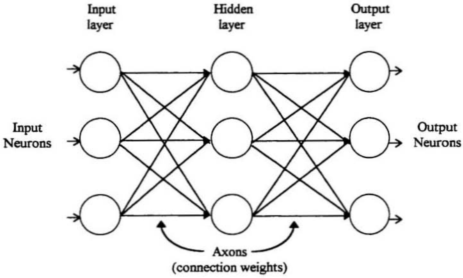


Figure 2.1. Simplified Neural Network Model

at the links connecting the inputs to the outputs (axons in Figure 2.1) and they associate the contribution or effect of each of these inputs on each output.

Training a neural network is accomplished by using a training algorithm that aims at optimally adjusting these network connection weights; training may be supervised or unsupervised. Supervised training, on the one hand, occurs when correct solutions are provided along with the problem description. In the case of unsupervised training, on the other hand, correct solutions are not provided. Neural networks trained in this manner are usually capable of self-organization and independent classification of the input data; that is, the network itself must decide how it will classify or partition the input data (Caudill and Butler, 1990).

One commonly used neural network architecture is the Backpropagation neural network (Rumelhart et al., 1968). Backpropagation networks are training algorithms in which patterns recognized by the network are associated through the layers, and thus the information flows in one direction at a time, either forward or backward. Backpropagation networks require at least three layers in order to work correctly, and training is conducted in a supervised manner. Training of a backpropagation neural network occurs in two stages (Caudill and Butler, 1990):

1) The input data pattern generates a forward flow of activation of the neurons from the input layer, through the hidden layers, and finally to the output layer, and

2) Errors in the output generate a flow of information from the output layer backward to the input layer. As the errors are propagated backward, the weights on the connecting "axons" are adjusted, therefore allowing the network to learn.

In addition to the backpropagation neural network, several other forms of neural network models or architectures have been experimented with, each of which has characteristics which make it appropriate for modeling different problems. These include the Perceptron network (Rosenblatt, 1961), the Counterpropagation network (Hecht-Nielson, 1987), the Boltzmann machine (Hinton and Sejnowski, 1986), the Hopfield network (Hopfield, 1982), the BAM (Bidirectional Associative Memory) architecture (Kosko, 1987), and the ART-2 (Adaptive Resonance Technique) (Carpenter and Grossberg, 1987). Table 2.1 (Moselhi et al., 1992) summarizes these architectures, along with their advantages and disadvantages.

Recently, researchers have examined the Counterpropagation neural network (Adeli and Park, 1995) for use in structural engineering. Counterpropagation networks were developed by Hecht-Nielson (1987); they contain a combination of several different neural network architectures and training algorithms as shown in Figure 2.2. In contrast to backpropagation networks, counterpropagation networks use both supervised and unsupervised training and therefore can map outputs in a self-organized manner. The counterpropagation network has been found to converge at a somewhat faster rate than

Table 2.1. Characteristics of Commonly Used Neural Network Paradigms

Network paradigm	Developer	Training time*	Execution time*	Information content*	Advantages	Disadvantages (limitations)	Utilization
Backpropagation	Rumelhart et al. 1986	Slow	Fast	High	• Powerful and accurate association • Suitable for static environment (input does not change with time)	• Could be trapped in local minima or paralyze • Not suitable for real-time applications • No incremental learning	High
Perceptron	Rosenblatt 1961	Medium	Fast	Low	• Very simple	• Limited representation capability	Low
Counterpropagation	Hecht-Nielson 1987	Medium	Fast	High	• Good for rapid prototyping • Suitable for static environment (input does not change with time) • Can generate a function andits inverse (Wassermann 1989) • Has powerful probabilistic capabilities	• Not as accurate as backpropagation • Could be trapped in local minima or paralyze • Not suitable for real-time applications • No incremental learning	High
Boltzmann machine	Hinton and Sejnowski 1986	Slow	Slow	High	• Alleviate local minima problem • Use probabilistic training method	• Very slow • Not suitable for real-time applications • No incremental learning	High
Hopfield network	Hopfield 1982	Fast	Medium	Low	• Provide dynamic (real-time) performance • Sufficiently responds to noisy, incomplete, or unseen inputs • Can be used for optimization	• Stability not guaranteed • Memory limitations (stored patterns) • Difficult to formalize training method • Could be trapped in local minima or paralyze • No incremental learning	High
BAM	Kosko 1987	Fast	Fast	Low	• Provide dynamic (real-time) performance • Sufficiently responds to noisy, incomplete, or unseen inputs • Can be used for optimization • Easier and more systematic training methodology than Hopfield • More simple and faster • No stability problem	• Could be trapped in local minima or paralyze • No incremental learning • Memory limitations (stored patterns)	Low
ART-2	Carpenter and Grossberg 1987	Fast	Medium	High	• Facilitate learning new examples without destroying previous experience (incremental learnins)		High

*Reference: Bailey and Thompson 1990

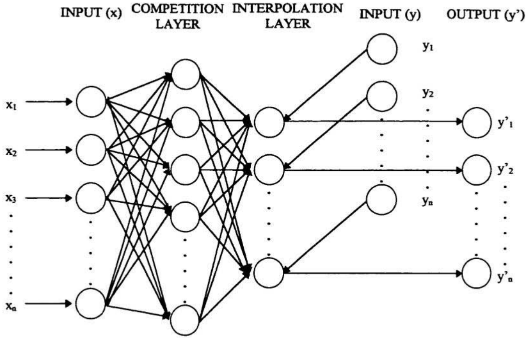


Figure 2.2. Counterpropagation Network (Adeli and Park, 1995)

the backpropagation network, therefore reducing the amount of time it takes to train the network. However, the errors produced while testing the network are comparable to those produced by the backpropagation network (Adeli and Park, 1995). Other applications of the counterpropagation neural network in structural engineering could not be found, thereby making this architecture an unexplored option for neural network users in structural engineering.

The fuzzy-ARTMAP technology has also been used to predict plain concrete material strength. This form of neural network architecture is a hybrid network that performs incremental, unsupervised learning of recognition categories and can perform a multidimensional mapping of space into a one dimensional space (Kasperkiewicz et al., 1995). This type of neural network, appears to perform successfully, however, like the counterpropagation network, research for this type of network in structural engineering is limited. In addition, because this type of network maps many dimensions into one dimension, it would work more successfully with problems that contained a great amount of input variables and only one or two output variables (Kasperkiewicz et al., 1995).

Despite the above-described recent research on alternate neural network architectures, backpropagation networks are the most widely used networks in civil and structural engineering. This is primarily because backpropagation neural networks are still the most simple form of neural network architecture. They also appear to be the most capable of learning the association between input and output patterns under a static environment given adequate training examples (Moselhi et al., 1992). Furthermore, most problems in civil and structural engineering involve the sort of predictions for which backpropagation networks are best suited.

During the past few years, the area of structural analysis has exhibited an increasing use of neural networks for a wide range of applications. Some of these include the modeling of initial design processes, the modeling of plain concrete material strength, and the modeling of reinforced concrete structural behavior. Researchers have the demonstrated the potential of using this technique in this domain.

2.3.3 General Applications in Civil Engineering

Most civil engineering systems are complex and are subject to a wide variety of internal and external forces (e.g., wave forces, weather conditions, seismic loads, and material mechanics). Analyzing such systems has been a difficult task and traditional tools that accurately predict and model the behavior of such systems are limited in scope. This is the main reason that Artificial Intelligence techniques have increasingly been experimented with in the civil engineering domain. Among these tools, Neural Networks (NNs) have been reported as efficient pattern recognition and classification tools that model the cause-effect relationships of a particular system or problem without exploring the underlying rationale used to model the behaviors (Hegazy et al., 1996). Correspondingly, the usefulness of neural networks as tools for design and decision support in civil engineering is well documented throughout the literature (e.g., Moselhi et al., 1992). Figure 2.3 summarizes examples found in the literature of applications of neural networks within civil engineering in the general realm, in the construction realm, and in the structural analysis realm.

Examples found in the literature of general applications of neural networks within civil engineering include a wide array of topics such as:

- Horizontal formwork selection (Hanna and Senouci, 1995)
- Control of structures under dynamic loading (Chen et al., 1995)
- Simple truss design (Kang & Yoon, 1994)
- Structural damage detection (Elkordy et al., 1994)
- Prediction of tower guy pretension (Issa et al., 1992)
- Dynamic analysis of bridges (Chen and Shah, 1992)
- Nondestructive examination of concrete (Pratt and Sansalone, 1992)

In addition, neural networks have been successfully applied to construction, specifically for equipment production estimation and construction trade productivity level estimation (Moselhi et al., 1992), as well as the assessment of construction risks in the bidding process (Hegazy,1993). Although none of these examples are directly related to the

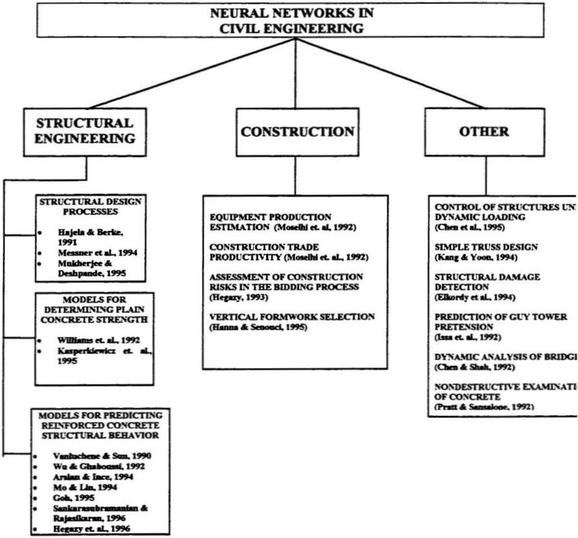


Figure 2.3. Examples of Neural Networks in Civil Engineering

current investigation for predicting the behavior of concrete slabs with neural networks, they serve to demonstrate the success of neural networks as an alternative to conventional algorithmic computation in obtaining solutions to general civil engineering problems.

2.3.4 General Applications in Structural Analysis

Neural networks are most suitable for applications that have the following features:

- A complex problem with a large number of governing parameters;
- A need for an alternative to a mathematical formulation of a solution to the problem; and
- Many examples of the problem are available for accurate training of a neural network.

Many problems, including those contained within structural analysis, meet the above criteria. As a result, structural engineers have, in recent years, found increasing interest in neural networks as an aid for both the design and analysis of structures. The first prototype application of neural networks as a tool for structural design was proposed by Vanluchene and Sun in 1990. The study demonstrated, through the use of three examples (a pattern recognition problem, a simple concrete beam design and an analysis of a rectangular steel plate), the wide range of possible uses for neural networks within the realm of structural design. Since then, neural networks have been applied to nearly every facet of structural engineering.

Examples in the literature of applications of neural networks in structural analysis as they apply to the current investigation can be grouped, as shown in Figure 2.3, into three categories: 1) models of the structural design process (e.g., Hajela and Berke, 1991; Messner et al., 1994; and Mukherjee and Deshpande, 1995), 2) models for determining plain concrete material strength (e.g., Williams et al., 1992, and Kasperkiewicz et al., 1995) and 3) models for predicting reinforced concrete material behavior (e.g., Wu and Ghaboussi, 1992; Mo and Lin, 1994; and Goh, 1995).

The first category includes an example of the implementation of a neural computing paradigm in automated structural design, where the structural analysis module is replaced by a neural network model to map load-displacement relationships (Hajela and Berke, 1991). Two architectures are proposed: a conventional, multilayered architecture and a functional-link net, which essentially involves non-linear transformation of the input data used in a flat, single-layered network. The report shows only limited success with the latter architecture, with errors reduced to the range of 3.9% to 5.2% and marginal increase in learning speed. The multilayered architecture performed better, with errors in the range of 1% to 2%. Because this problem is similar in nature to the problem at hand, specifically the load-deflection model, the information regarding applicabilities of neural network architectures is useful for selecting the appropriate architecture for the models of concrete material properties.

A second example within the first category describes the development of a computer application to select the most effective structural system for a building given attributes regarding the project size, budget, etc. (Messner et al., 1994). This paper explores the rationale for choosing a neural network model over a rule-based expert system model (another form of artificial intelligence). The ultimate reason for choosing a neural network is because of the many interrelations between the different project parameters and the lack of explicit causalities between these parameters (Messner et al., 1994). This can be directly related to the current research for determining the structural behavior of concrete slabs because the input data for these neural networks consists of many different properties with respect to the concrete slabs whose interrelationships have not exactly been determined.

Another example involves the modeling of initial design processes using neural networks. This example (Mukherhjee and Deshpande, 1995) uses traditionally selected design criteria as input and uses the neural network to determine the size (i.e., depth and width), reinforcing steel area, cost/m and moment capacity of a reinforced concrete beam. Unlike the current investigation, however, this model uses mathematically generated data to train the network, as initial design processes are readily modeled using more traditional mathematical computations. Nevertheless, the neural network model is found to perform as competently as mathematical models. Furthermore, the paper also explores the effect of damaging connection links on the desired output for the neural network; it is found that as many as two nodes could be damaged with little effect on the overall performance of the neural network.

2.3.5 Models for Determining Plain Concrete Material Strength

From a review of the literature it is found that a limited number of studies have been conducted on the use of neural networks for predicting concrete strength. The first example of such research (Williams et al., 1992) utilizes the same neural network development software (Neuroshell) as is proposed in the current investigation to determine the structural behavior of concrete slabs. The model utilizes data regarding one day, three day and seven day compression strengths as inputs for the model to determine the twenty-eight day compressive strength as output. The study observes reasonable performance of the neural network as compared to linear regression analysis. It also determines that, with the limited data used to train the model, the performance of the network appears to improve with the addition of input variables to the model; five different models are trained with an increasing number of input variables, and the accuracy of the network improves with the addition of each input variable.

The data from this report is extremely useful for the current investigation as it demonstrates the suitability of Neuroshell software for a model similar to the proposed model. However, the model described in the report only addresses plain concrete (homogeneous) material behavior as opposed to reinforced concrete (composite) material behavior.

A more recent study addresses the same problem, i.e., prediction of concrete strength, however a greater number of different variables are selected to model the input for the problem. Moreover, a different neural network architecture with a different learning paradigm (the fuzzy-ARTMAP neural network) is selected to model the problem (Kasperkiewicz et al., 1995). Once again, the network is found to perform satisfactorily, however, the study warns that satisfactory performance only occurs when the network is tested with problems containing data within the same domain as the data used to train the model (see discussion in Section 4.1).

2.3.6 Models for Predicting Reinforced Concrete Material Behavior

Several studies have been directed at the investigation of the use of neural networks to predict the behavior of a variety of reinforced concrete elements.

The first investigation regarding the feasibility of using neural networks to model reinforced concrete behavior studied a simple reinforced concrete beam subjected to bending moment (Vanluchene and Sun, 1990). This study utilized NNICE (Neural Networks in Civil Engineering), a neural network software package which employs the back-propagation training algorithm. This study used as input only a limited number of variables to describe the concrete behavior (bending moment applied, reinforcing steel strength, concrete compressive strength and reinforcing steel ratio) to arrive at an ideal depth for the beam (the only output for the network). Training and testing was conducted using randomly chosen patterns obtained from conventional mathematical formulas rather than data obtained from experimental results. While limited in its scope, this study was the initial impetus for the use of neural networks in concrete design.

More recently, a study was conducted to analyze framed shearwall behavior using neural networks (Mo and Lin, 1994). Again, only limited data was utilized as input to describe the concrete material behavior (concrete compressive strength, steel yield stress, longitudinal steel ratio and shear strain); the only output parameter was shear stress. Two study groups were used for training and testing the network; one study group included results from experimental tests while the other study group included results from calculations of the truss model theory (described in Section 2.2). Models for both study groups performed well. The paper suggested that the methods used could be applied to the behavior of other concrete structures. Also, it found that the effect of the transfer functions and learning rules on the network is significant while the effect of the number of processing elements in the hidden layers on network learning is insignificant (Mo and Lin, 1994).

The feasibility of using neural networks to evaluate the ultimate strength of deep reinforced concrete beams in shear has also been investigated (Goh, 1995). Again, both experimental data and data obtained from mathematical calculations were available for training and testing the network. The study showed that, when compared to conventional methods (the strut-and-tie model discussed in Section 2.2) for predicting the ultimate strength, the neural network approach was actually more reliable.

In recent years, researchers have studied the use of neural networks for material modeling. The major thrust of their research has been aimed toward the development of proper constitutive relationships for finite element modeling of the material (Ghaboussi et al., 1991; Wu and Ghaboussi, 1992; Sankarasubramanian and Rajasekaran, 1996). Because concrete is a difficult material to model from a finite element perspective, neural networks have been investigated as an alternative to lengthy mathematical derivations of constitutive equations. Like the proposed investigation, these neural networks were trained using results from actual experiments conducted on the concrete.

Ghaboussi, Garrett and Wu originally studied the use of neural networks to predict the stress and strain behavior of plain concrete (Ghaboussi et al., 1991) and later extended this study to include the stress and strain behavior of reinforced concrete (Wu and Ghaboussi, 1992). In the models for plain concrete, the input for the networks included stress and strain increments, and the output included either stress or strain increments, depending on whether the neural network was considered to be stress-controlled or strain-controlled. The models for reinforced concrete included all pertinent data to describe the behavior of the concrete. The data included the concrete compressive strength and strain and the cracking strength of the concrete; this information was implicitly included in the stress-strain material variables through normalization on the principal compressive and tensile stresses and principle compressive strains (Wu and Ghaboussi, 1992). Also included in the input data was information regarding the reinforcing steel such as yield stress and reinforcement ratio. Finally, the stress and strain states for two stress increments were included in the input data. The output data for the reinforced concrete model again included current stress or strain increment. All neural network models were found to perform satisfactorily, i.e., they were able to predict stress and strain states with reasonable accuracy.

Although the outcome of the Ghaboussi, Garrett and Wu models was then utilized in a finite element model for concrete, these rationale behind the formulation of the neural network models for these studies was similar to the current investigation. Therefore, the content and methods of these studies will be very useful for the current investigation. However, these studies do not directly conflict with the current investigation as they were completed for a different purpose. Also, the neural networks were trained using experimental data from tests conducted on reinforced concrete panels subjected to in-plane shear in contrast to the reinforced concrete slabs subjected to concentrated and flexural loads used in the present investigation. Therefore, the information sought in the current investigation is outside of the training domain for these studies.

A very recent study again addresses constitutive modeling of concrete using neural networks (Sankarasubramanian and Rajasekaran, 1996). However, neural networks are utilized only to predict one aspect of the stress-strain curve and do not consider concrete material properties for input. This study is useful as it again shows the success of neural networks in similar applications to the investigation for predicting concrete structural behavior.

Neural network research for concrete slabs has focused on the initial structural design of these slabs. The neural network developed by Arslan and Ince (1994), for example, takes the moment and slab support conditions as inputs to produce only the moment coefficients needed for slab design. None of such efforts, however, predict the slab's overall responses to loading conditions.

2.3.7 General Neural Network Models for Predicting Crack Patterns

Limited research has been conducted in using neural networks to model cracks in general. One application studied the detection and mapping of cracks in eggs (Patel et al., 1994). This investigation used computer vision to model the picture of a cracked egg using a grid of pixels. In contrast to the current investigation, the only output of this network predicted whether the egg was cracked or not; the study did not address the prediction of the actual pattern of the crack. Like one of the crack pattern neural networks in the current investigation (NN2a), however, the success or failure of this model was based on a percentage of correct predictions, not the actual numerical accuracy of the model.

2.4 Summary

In the present study, the use of neural networks in predicting the structural behavior of concrete in slabs is experimented with for several reasons, including:

- Neural network approximations are equally as accurate as other complex mathematical approximations (Carpenter and Barthelemy, 1993);
- Neural networks are able to generalize solutions to new, unseen cases, most accurately within the training domain (Flood and Kartam, 1994); and
- An adequate number of training cases will be used to train the network as experimental results on full-scale slabs were monitored since 1990 (Marzouk and Hussein, 1991).

This chapter has reviewed the previous work related to neural networks in civil engineering and structural analysis. While it is apparent that a large amount and variety of applications of neural networks exists in these fields, there is no single application that has been used to determine the structural behavior of concrete slabs in particular. However, all of the previous works described provide significant insight into the development and modeling of a neural network for the current investigation of the structural behavior of concrete slabs.

Chapter 3

Neural Network Model of a Reinforced Concrete Slab's Structural Behavior

3.1 Introduction

The structural behavior of reinforced concrete slabs can be quantitatively described in a number of different ways. In this study, four complementary aspects were chosen to represent this behavior as they can provide a structural engineer with valuable insight into the failure mechanism of a concrete slab. These aspects are: 1) load-deflection behavior; 2) final crack pattern formation; 3) reinforcing steel strain distribution at slab failure; and 4) concrete strain distribution at slab failure. Each aspect lends itself to a neural network, therefore, four separate neural network models have been developed to predict these aspects.

A structured methodology for neural network application development (Hegazy et al., 1994) was utilized as an overall framework for developing each neural network. The methodology incorporated three main phases as illustrated in Figure 3.1: 1) concept; 2) design; and 3) implementation. This chapter will focus on the completion of the first two phases of the model development, while chapter 4 will focus on the final phase, that is, the implementation phase of model development.

3.2 Neural Network Concept Development

The concept stage, as shown in Figure 3.1, includes two steps that involve selecting the application then the paradigm for the neural network model. The first step involves choosing an application which is amenable to neural network modeling. All four of the proposed neural network modules encompassed by the current study were easily converted to neural networks as the inputs and outputs for each module were readily defined, as described below (Section 3.3).

Since a concrete slab behaves differently according to the variety of combinations of factors that describe the slab, it could be said that the behavior of a concrete slab is patterned according to its makeup. Therefore, the four neural network applications that were selected for the current study are primarily pattern recognition problems. For the second step of the concept phase, the Backpropagation paradigm was selected as the neural network type suitable for modeling the applications. The principal reason this architecture was chosen is that, as described in Chapter 2, it is the predominant paradigm

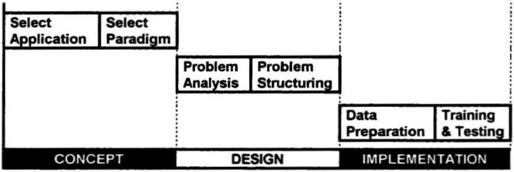


Figure 3.1. Neural Network Development Methodology

used in existing structural analysis applications due to its simplicity and its suitability for pattern-recognition problems.

3.3 Model Design

The next phase of neural network development is model design, which comprises two main tasks: 1) problem analysis; and 2) problem structuring. Problem analysis, on the one hand, is the identification and use of the independent (non-correlated) factors that fully describe the slab. Problem structuring, on the other hand, entails the representation of such descriptive factors along with their associated result in the form of inputs and outputs, as required by the modeling of each individual neural network. Identification of the input data was conducted simultaneously for all four neural networks (NNs) as the same data was used to describe all of the reinforced concrete slabs in the study. The factors which could describe the physical properties of a reinforced concrete slab were first grouped into four main categories: 1) slab geometrical dimensions; 2) aggregate properties; 3) concrete properties; and 4) reinforcement-steel properties. The inputs for all four NNs were then readily defined from these four categories; boundary and loading conditions for each slab were also added to the input descriptions. These resulted in a total of nineteen input factors as described in Table 3.1.

Problem analysis was required on an individual basis for each of the four NNs in order to determine their outputs. The outputs for all four models were obviously different, as each

Table 3.1. Description of Input Categories and Factors

	INPUT DEFINITION
Slab Geometric Properties	1. Slab thickness (mm) 2. Slab Depth (mm) 3. Ratio of Rebar Depth to Slab depth 4. Slab Span (mm)
Aggregate Properties	5. Aggregate Type (1=Sandstone; 2 = Granite) 6. Aggregate Size (mm)
Concrete Properties	7. Concrete Compressive Strength (MPa) 8. Concrete Tensile Strength (MPa) 9. Concrete Modulus of Elasticity (MPa)
Reinforcement Steel Properties	10. Reinforcing Steel Ratio 11. Rebar Size (1=M10; 2=M15) 12. Rebar Shape (0=Smooth; 1=Deformed) 13. Rebar Spacing (mm) 14. Number of Rebar Layers 15. Rebar Yield Strength (MPa) x 10,000 16. Rebar Modulus of Elasticity (MPa) x 10,000 17. Type of Shear Reinforcement (0=None; 1=Hat; 2=U-Shape; 3=W-Shape)
Loading & Boundary Conditions	18. Load Type (0=Axial; 1= Bending; 2=Axial+Bending; 3=Cyclic 19. Boundary Conditions (0=Simply Supported; 1=Fixed; 2=Partially Fixed)

model was designed to produce separate yet complementary results. In addition, different ways of problem structuring (the second half of model design) were experimented with for each NN in order to achieve the optimum network to solve the problem. The resulting structure for each NN is described in the following paragraphs, with the results and corresponding optimum structure described in Chapter 5.

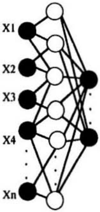
3.3.1 NN1: Load-Deflection Behavior

The load-deflection neural network model was the first model tested in the current study; the validity of the neural network technique was tested in this original neural network. For this reason, two separate neural network models were experimented with in detail in order to arrive at the optimal configuration of the outputs for this model. As shown in Figure 3.2, the number of inputs and outputs in each network are as follows:

Neural Network 1a: 19 inputs (slab descriptors) and 11 outputs (load-deflection descriptors). Load-deflection curve in this case is modeled as ten values corresponding to deflections at each 10% load increment and an eleventh value representing the ultimate load reached is provided.

Inputs

X1. Aggregate Type (1=Sandstone; 2=Granite)
X2. Aggregate Size (mm)
X3. Load Type (0=Axial; 1= Bending; 2=Axial+Bending; 3=Cyclic)
X4. Slab Thickness (mm)
X5. Slab Depth (mm)
X6. Ratio of Rebar Depth to Slab Depth
X7. Slab Span (mm)
X8. Boundary Conditions (0=Simply Supported; 1=Fixed; 2=Partially Fixed)
X9. Concrete Comp. Strength (MPa)
X10. Concrete Tensile Strength (MPa)
X11. Concrete Modulus of Elasticity (MPa)
X12. Reinforcing Steel Ratio
X13. Rebar Size (1=M10; 2=M15)
X14. Rebar Shape (0=Smooth; 1=Deformed)
X15. Rebar spacing (mm)
X16. Number of Rebar Layers
X17. Rebar Yield Strength (MPa) x 10,000
X18. Rebar Modulus of Elasticity (MPa) x 10,000
X19. Type of Shear Reinforcement (0=None; 1=Hat; 2=U-Shape; 3=W-Shape)



Outputs

Neural Network 1:
O1. Defl. at 10% Ult. Load (mm)
O2. Defl. at 20% Ult. Load (mm)
O3. Defl. at 30% Ult. Load (mm)
O4. Defl. at 40% Ult. Load (mm)
O5. Defl. at 50% Ult. Load (mm)
O6. Defl. at 60% Ult. Load (mm)
O7. Defl. at 70% Ult. Load (mm)
O8. Defl. at 80% Ult. Load (mm)
O9. Defl. at 90% Ult. Load (mm)
O10. Defl. at 100% Ult. Load (mm)
O11. Ultimate Load (KN)

Neural Network 2:
O1. Yield Load (KN)
O2. Deflection at Yield (mm)
O3. Ultimate Load (KN)
O4. Deflection at Ult. Load (mm)

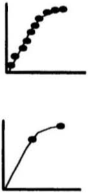


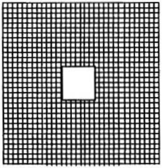
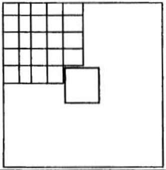
Figure 3.2. Description of L-D Neural Network Inputs and Outputs

Neural Network 1b: 19 inputs (slab descriptors) and 4 outputs (load-deflection descriptors). Load-deflection curve in this case is modeled as four values corresponding to a slab's yield load, deflection at yield, ultimate load, and deflection at ultimate load.

The final configurations for these two models are as shown in Figure 3.2.

3.3.2 NN2: Crack Pattern at Failure

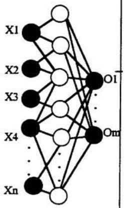
As previously described in Section 3.3, the inputs for this neural network model consisted of the nineteen factors describing the concrete slab. The selection of the outputs to describe the final crack pattern for a reinforced concrete slab, however, was a difficult task; while recognizing patterns of images is an intuitive and simple task for humans, it is a complex task for computers and requires specialized AI-based modeling. Despite the modeling difficulties, a concrete slab's crack pattern provides insight into the failure mechanism of the slab and its rate of deterioration under loading. The outputs for this neural network could be modeled in a variety of ways. These varied from exact detailing of the dimensional location of cracks to less detailed schematic representations. The detailed model, however, was expected to involve a large size neural network, thus requiring a larger number of training cases than were available from the experimental testing. Three schematic models were then proposed as shown in Figure 3.3, and the

MODEL	DESCRIPTION	NO. OF OUTPUTS	ADVANTAGES	DISADVANTAGES
1	Scanned image divided on a 32 x 32 grid 	1024 data points either 0 or 1	• Exact dimensional locations of cracks could be predicted • Crack width at any spot can be modeled using number greater than 1	• Network too large (too many outputs) • Network is NOT expected to train on the limited cases available
2	5x5 grid of one quadrant 	25,each with values corresponding to Horizontal (0), Vertical (1) or Inclined (2) crack	• Exact locations of cracks could be predicted • Symmetrical slabs would require less input	• Pattern needs to be symmetrical • Large amount of time required for data input • Difficult to model due to large grid size (each cell may contain more than one crack)
3	Identify the crack shape characteristics in terms of: • Type of Background • Crack Symmetry • Extent of Cracking • Location of Tangential Cracks • Density of Tangential Cracking • Radial Position of Tangential Cracks • Density of Radial Cracking Extent of Grid	8 characteristics	• Minimal number of outputs required • Incorrect outputs could still produce a satisfactory picture • Can be trained using the limited training cases available	• Exact location of cracks cannot be determined

advantages and disadvantages of each model were reviewed. After thorough analysis and initial experimentation with the three types of models, the third approach was selected due to its simple representation and its appropriate proportion of outputs to inputs. A fourth model, which was designed with a less subjective approach to produce quantitative results for the extent of radial and tangential cracking, was also selected for comparison with the results for the schematic model chosen. The final outputs for the two neural network models are as shown in Figure 3.4.

3.3.3 Concrete Strain Distribution

An effective indicator of the extent of cracking throughout a concrete slab is the distribution of the strains throughout the slab, that is, if the measured concrete strains are greater than the crushing strain of concrete (approximately greater than .00035), it can be assumed that a crack will have occurred at the measured location. Therefore, representative values for the concrete strain at the edge of the slab, at a midpoint of the slab and at the column face at failure of the slab would indicate the extent of the cracks throughout the slab. These values were easily converted to outputs for the first neural network model for predicting the distribution of the maximum concrete strains throughout the slab. Again, all nineteen inputs described in Section 3.3 were used in this model. Originally, prediction of both tangential and radial strain distribution was proposed, however, only tangential strains were measured along a radius for most of the tested slabs.



Outputs

Neural Network 1:

O1- Background (0=none; 1=radial; 2=grid; 3=radial + grid)

O2- Crack Symmetry (0=assymetrical; 1=sym about horiz axis; 2=sym about vert axis; 3= fully symmetrical)

O3- Extent of Cracking (0=none; 1=upper left quad; 2=UR; 3=LL; 4=LR; 5=all quads)

O4- Location of Tangential Cracks (0=none; 1=upper left quad; 2=UR; 3=LL; 4=LR; 5=all quads)

O5- Failure Mechanism (0=no tang. cracks; 1=pure punching shear; 2=ductile punching shear; 3=bending from unbalanced moment; 4-bending)

O6- Radial Position of Tangential Cracks (0=none; 1=@stub col; 2=inner 1/3; 3= outer 1/3)

O7- Density of Radial Cracking (0=none; 1=light; 2=heavy UL; 3=heavy UR; 4=heavy LL; 5=heavy LR; 6=all heavy)

O8- Extent of Grid (0=none; 1=throughout; 2=within tangential crack)

Neural Network 2:

O1 - Radius of Tangential Cracking (mm)

O2 - Extent of Radial Cracking (0=none; 1 = inner third; 2 = outer third)

Figure 3.4. Description of Crack Pattern Network Outputs

Thus, one neural network model was developed to predict only the tangential strain distribution.

A good indicator of the concentration of stresses at the face of the stub column would be the measured concrete strains at various loads at the column face. This would provide information regarding any failure that could occur at the slab-column connection, since this point is the most stressed point on the concrete slab. So, another neural network model was developed to predict concrete strains at various load increments at the column face. The final outputs for both concrete strain models are shown in Figure 3. 5.

3.3.4 Reinforcing Steel Strain Distribution

A fourth group of neural network models was developed to predict the distribution of reinforcing steel strains throughout the concrete slab. This neural network group was designed to provide information regarding the extent of yielding of the reinforcing steel. This information is useful because, when the reinforcing steel yields, the full tensile and compressive loads are carried by the concrete alone; failure of the slab would probably first occur at this location. One neural network model was constructed; this was designed to predict the radius of yield for the reinforcing steel only as well as the distribution of the maximum strains in the reinforcing steel along a radius through the slab. The resulting neural network models are shown in Figure 3.6.

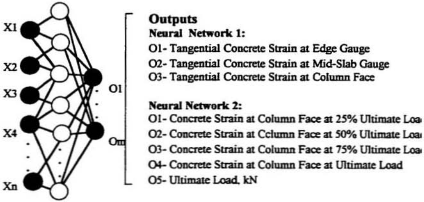


Figure 3.5. Description of Concrete Strain Network Outputs

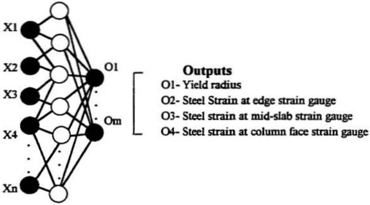


Figure 3.6. Description of Reinforcing Steel Strain Network Outputs

3.4 Summary

Development of a reliable neural network model is essential for the proper prediction of results for a problem. Therefore, the appropriate selection of inputs and outputs for each NN module proposed for the prediction of the structural behavior of reinforced concrete slabs must be conducted prior to implementation of the model. This chapter discusses the selection of these inputs and outputs, along with the reasons for their selection. Chapter 4 will then discuss the implementation of these models, along with the assignment of actual data to the inputs and outputs developed in this chapter.

Chapter 4

Computer Implementation of the Neural Network Models

4.1 Introduction

The implementation phase of neural network model development is comprised of two main tasks: 1) training data preparation; and 2) training and testing. Once the inputs and outputs for the four neural network models were defined in the concept and model design phases, the validity of the neural network concept was then tested by conducting training and testing on the data for the concrete slabs. This chapter addresses both phases of model implementation for the four neural networks that were designed in Chapter 3.

4.2 Data Preparation

The basis of neural network modeling is a training mechanism on a group of known examples of problems and their solutions. Therefore, to develop neural network predictive models of a concrete slab's structural behavior, existing data on some experimental slabs were used in this study. Over the past ten years, extensive research on the structural behavior of concrete slabs has been conducted at the Memorial University of Newfoundland including a number of experimental tests on full-scale reinforced concrete slabs. The research has been documented in several publications (Marzouk and Hussein, 1991; Emam et al., 1995, and Jiang, 1994). The experimental tests reported in Marzouk and Hussein (1991) studied the behavior of seventeen normal and high strength concrete slabs subjected to concentrated loads applied axially through a stub column. Following that, additional tests (Emam et al., 1995) were conducted on fourteen reinforced concrete slabs and column connections subjected to not only axial load but also bending moment. To further study the effects of shear reinforcement on the slabs' behavior, Jiang (1994) conducted supplemental tests on seven high strength concrete slabs. For each slab tested in these studies, detailed information regarding the factors that describe a concrete slab and accordingly affect its structural behavior were documented.

A data acquisition system was connected to the test setup of the concrete slabs and was used to automatically record several data elements during all of the tests. Deflection at the slab centers was measured at a series of loads using linear variable differential transformer (LVDT) gauges. Using electrical strain gauges, steel strains were measured at different points at the surface of the reinforcing steel while concrete strains were measured at the compression face of the concrete slab. Cracks were marked during loading and the final crack patterns were photographed.

The first step in preparation of data involved formulating the load-test results for the thirty-eight full scale reinforced concrete slabs in the appropriate input and output formats for each neural network model. Details on how the data was extrapolated for each neural network are included in Table 4.1. Final crack patterns were not available for all thirty-three slabs and some of the concrete and reinforcing steel strain gauges were damaged during testing, so not all of the slabs could be used for training and testing NNs 2 through 4. The following analysis of the strain gauge data (both for the concrete and reinforcing steel) was required. Strain gauge readings were reviewed for consistency; those that remained at the extremes (near 0 or 1) throughout the test or that fluctuated significantly during the course of the test were considered to be unreliable. If strain gauge readings did not appear reliable, the entire case was removed from the pool of data available for training and testing the neural networks. The resulting total number of slabs used for training cases and those that were reserved for later testing of all of the trained neural networks is shown in Table 4.1.

In order to validate the information content of the training cases used, a simple test was first conducted on training data for the seventeen slabs used in the Hussein study (1991); this was completed prior to training NN1 (load-deflection curve). The test examined the

Table 4.1. Data Extrapolation Methods

Neural Network #	Description	Extrapolation Method	# of Training Cases	Number of Test Cases
1	Load-Deflection	Data extrapolated from plot of load deflection curve (produced during testing)	27	7
2	Crack Pattern	Photos of crack patterns visually interpreted	28	7
3	Concrete Strain	Strain gauge readings	7	2
4	Steel Strain	Strain gauge readings	9	3

relationship between an input parameter (e.g., concrete compressive strength) and an output parameter (ultimate load reached), depicted in all of the training cases. These relationships (or general trends) were established through simple regression analysis and then compared with common knowledge in this domain. Following this analysis, the concrete compressive strength exhibited a logical direct relationship with the ultimate load reached by the slab, and as such, it was concluded that the data was sufficient for initial training of NN1.

4.3 NeuroShell 2 Software

NeuroShell (1990) is an existing neural network software package which contains all of the features that are necessary to train and test a neural network. This original version has been upgraded several times since the original issue. NeuroShell (1990) was used for the initial training and testing of NN1 (Load-deflection). The information was then transferred to NeuroShell 2, Release 3.0, and the upgraded software was then utilized for final modeling of the problem and training of all of the networks (NNs 1 through 4). The Windows-based neural network software was chosen for its ease-of-use, speed of training, and for its host of features that permit user optimization of network training. Some advantages of NeuroShell 2 include: 1) the ability to import and export data files; 2) the choice of several different neural network architectures, which allows the user to select the paradigm most suited to his/her particular application; and 3) visual training, which allows the user to evaluate when training is sufficient by viewing the training graphically or by viewing the network training statistics. Figure 4.1 demonstrates the user-friendly aspect of NeuroShell 2. For more details regarding the NeuroShell 2 software, the reader is referred to the NeuroShell 2 User's Manual (1995).

4.4 Training

After the data for the training cases (Appendix A) was input to the software, training was completed for all of the neural networks. Originally, for NN1 (load deflection), training was conducted on the twelve slabs contained in Marzouk and Hussein's study (1991). This was done to confirm the suitability of the neural network technique for the problem at hand. Once this was confirmed, fifteen cases from the remaining two studies were later added and retraining was conducted. The addition of the results from these tests widened the domain of concrete slabs included in training, thereby augmenting the ability of the neural network to generalize the model. The remaining three neural network models (NN2 through 4) were then trained with all of the training cases available for each model. The same iterative procedure was utilized for training each of the four neural networks.

Two separate forms of neural network architecture were utilized to train all four neural network models: the Backpropagation architecture (BP) and the General Regression

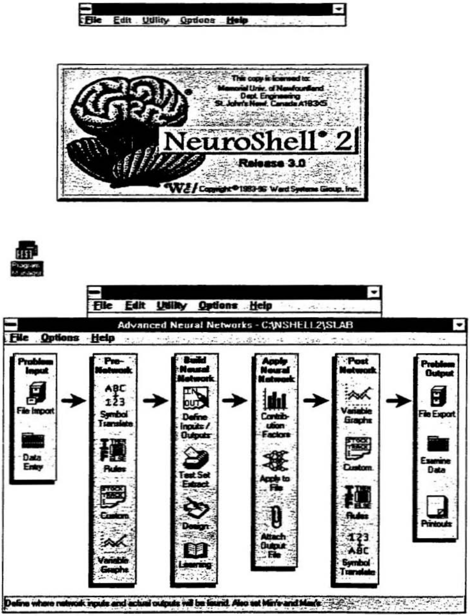


Figure 4.1. Neuroshell 2 Software

Neural Network (GRNN). These two architectures were selected to assure that the optimum neural network configuration was chosen to train the models.

4.4.1 Backpropagation using NetPerfect

Since training is essentially an iterative process, two simplified approaches were carried out to arrive at the optimal training level for each neural network using BP. In the first approach, the "NetPerfect" feature of the software was used where, at predetermined intervals during training, NeuroShell 2 would test the network on an independent test data set. If the error on the test set was lower than the previous optimal network, the new network would be saved; this process would continue until no improvement in the network occurred. An average error for all training cases would then be computed for comparison purposes. The TurboProp feature of NeuroShell2 was also used; this method adjusts network connection weights only after the network encounters an entire set or epoch of training patterns. By doing this, the network can converge at a faster rate than when weights are randomly updated without the network seeing an entire set of training patterns.

4.4.2 Backpropagation using Stepwise Training

Stepwise training was next conducted by successively increasing the number of training epochs (i.e., cycles through a complete training set) beyond which the error was minimized on the training set. This was done because the smaller number of cycles at the earlier stages exposed the network to less training time and thus the network did not focus on the training cases. This would result, theoretically, in good generalization capabilities of the neural network on any test cases for concrete slabs not previously applied to the neural network. On the one hand, if training time were not sufficient, this could mean unacceptable network performance. On the other hand, if the network were overtrained (i.e., higher number of cycles since average minimum error), this could occur at the expense of its generalization performance. Steps of 50, 100, 200 and 1000 cycles beyond the minimum error were progressively applied as the training time was increased for each neural network model. Once again, average errors for training cases were computed for comparison purposes.

4.4.3 General Regression Neural Network

An alternate neural network paradigm, the general regression neural network, was also applied to all of the NN models. This type of neural network has been shown to perform best on models for which there is only a minimal amount of data available to train and test the model. This architecture was experimented with in this study as there was a limited number of cases available to train and test some of the neural network models; this paradigm could theoretically provide lower errors than the more conventionally used Backpropagation neural network.

4.5 Testing

Once the neural networks were trained, the predictive capability of the neural networks was then checked on an independent test set. In this case, a weighted average of the errors for the sample and test cases was computed, using a 70% weight on the test cases, and a 30% weight on the training cases. These weights were randomly chosen to emphasize a greater weight on the test cases because, it could be assumed that the neural network would have a greater chance for learning the results for cases previously shown to it (training cases) and a lower error on these cases would be expected; however, a lower error for the test cases would indicate greater performance of the network. Once the results were reviewed, the optimum neural network models were chosen for each module (those with the lowest weighted error), along with the ideal training method to be utilized.

4.6 Summary

Data preparation and training/testing are processes which are used to implement the neural network model. Several iterations of this process are usually required in order to achieve optimum results for the model. General details regarding model implementation are discussed in this chapter, along with a review of the software utilized to accomplish the training and testing for all of the networks. Chapter 5 addresses the results obtained from training and testing each of the neural network models, as well as the selection of the ideal neural network models and training methods utilized for each problem.

Chapter 5

Results

5.1 Introduction

Results for all of the four neural network modules were produced by the software and were reviewed on an ongoing basis as each NN was trained and tested. From these results, the optimum network was selected for each problem in two stages. The first stage consisted of choosing the ideal training method and network architecture by minimizing the weighted error for each method and architecture. The ideal step for stepwise training was first selected and compared with the results produced by the use of the NetPerfect feature. Then, the results produced using backpropagation were compared with the GRNN model and the optimum architecture (with the lowest weighted error) was selected. As previously described, the weighted error is an average of the errors for the sample and test cases, with a 70% weight used on the test cases and a 30% weight on the training cases. The second stage involved choosing the ideal model for the problem, again by minimizing the weighted errors produced for each individual model and by also evaluating the ability of the network to produce results (for example, load-deflection curves) consistent with those produced by the experimental load tests. The results for the optimum model for each neural network are tabulated in Appendix B.

5.2 Load-Deflection Behavior

The results produced by stepwise training for NN1a, which predicted the deflection of the slab at ten different load increments and the ultimate load, and NN1b, which predicted the yield and ultimate loads and deflections, are plotted in Figure 5.1, showing the training stages versus the weighted network performance error for each network. As shown, NN1a performed ideally at the last step (training iterations beyond minimum average error = 1000), with a weighted error of 16.31%, while NN1b performed ideally at the third step (training iterations beyond minimum average error = 200), resulting in a weighted error of 15.09%. This illustrates the necessity for conducting stepwise training separately for each NN, as each NN could produce the minimum error at a different level.

Next, Table 5.1 compares the weighted errors resulting from stepwise training with those produced by use of the NetPerfect feature. It can be seen from the results for both networks that the NetPerfect feature trained the networks in the optimum manner. Although the average error on the training cases was lower at the optimum stepwise

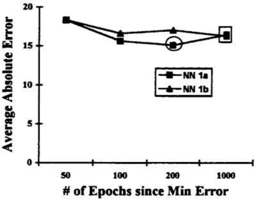


Figure 5.1. Stepwise Training of Neural Networks 1a and 1b

Table 5.1. **Results for Training Load-Deflection Curve Neural Network**

Model	Network Architecture	Training Mechanism	Average Error⁺ on Training Cases (%)	Average Error on Test Cases (%)	Weighted Error⁺⁺ on all Cases (%)
	Backpropagation	NetPerfect*	8.36	19.72	16.31
NN 1a:	Backpropagation	Stepwise**	1.91	22.49	16.31
	General Regression NN	N/A	1.72	13.85	10.22
	Backpropagation	NetPerfect	6.87	12.04	10.48
NN 1b:	Backpropagation	Stepwise	7.59	18.30	15.09
	General Regression NN	N/A	19.83	20.97	20.63

* - Auto-optimization feature of Neuroshell2

** - Training method by which the number of epochs since minimum error is sequentially increased until the optimum results are obtained.

\+ - Average error = Absolute value of $\frac{\text{(Network Output - Actual Output)}}{\text{Actual Output}}$

++ - Weighted Average Error = (0.3 x Training Error) + (0.7 x Test Error)

training level for each network, the average error for the independent test cases was much higher (>22% for NN1a and >18% for NN1b), leading one to conclude that the networks actually overtrained on the sample cases and may have lost their ability to generalize for any example presented to the network.

Table 5.1 also compares the weighted errors produced by the backpropagation NN with those produced by the GRNN model. While the GRNN version of NN1a showed a lower average weighted error than the optimum network trained using backpropagation (NN1b), the ultimate load and deflection predicted by the network fell far short of the actual ultimate load deflection reached in the experimental tests, as shown in the sample slabs in Figures 5.2 and 5.3. And, the error produced by the GRNN network for NN1b was in excess of 20%, the worst for all of the training conducted. Therefore, the usefulness of the backpropagation architecture for this particular neural network was confirmed.

The load-deflection curves generated by the optimum network for each NN architecture were then plotted against the actual curves produced by experimental load-tests. Sample load-deflection curves for each NN architecture and training method are shown in Figure 5.2 for NN1a and Figure 5.3 for NN1b. As is shown by these curves, the curves produced by NN1b (backpropagation/NetPerfect) more closely matched those produced from the experimental tests. With an overall minimum weighted error of 10.48%, it was determined that this neural network could be used as a reliable alternative to the costly test

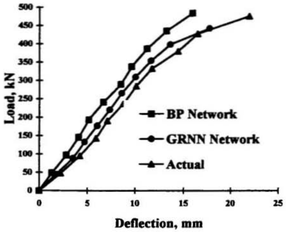


Figure 5.2. NN1a Load-Deflection Curve for Slab # M1

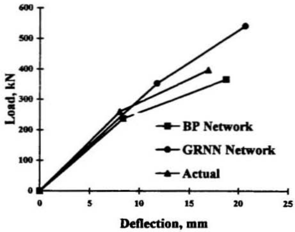


Figure 5.3. NN1b Load-Deflection Curve for Slab # M1

procedures for the prediction of the load-deflection curve values, both numerically and graphically.

5.3 Crack Pattern at Failure

The accuracy of the crack pattern model was more difficult to interpret as the ability of the network to produce a correct interpretation of final crack patterns was equally as important to the final outcome as the network performance error. Thus, the first crack pattern model, NN2a, which identified the crack shape characteristics in terms of eight output parameters, was first evaluated to determine whether it sufficiently represented the crack patterns. The errors for training and test sets of this model were computed in two separate ways. In the first method, a weight from 0 to 1 was chosen for each output according to the ability of that output to affect the overall picture of the crack pattern. These weights were then applied to the error for that output (error computed using the absolute value of $\frac{\text{Actual - Network}}{\text{Actual}}$). An average of all of the weighted errors was then computed for both training and test cases. As was done in the load-deflection curve model, a final weight was applied to this average, using a 70% weight on the test cases, with a 30% weight on the training cases. The final errors are summarized in Table 5.2 for both backpropagation and GRNN models. Using this method for comparing errors, the second stage of stepwise training (number of cycles since minimum average error = 100) provided the optimum results, with an average weighted error of 24.8%. As

Table 5.2. Results for Training Crack Pattern Network (NN2)

Model/ Error Method	Network Architecture	Training Mechanism	Average Error⁺on Training Cases (%)	Average Error on Test Cases (%)	Weighted Error⁺⁺ on all Cases (%)
NN 2a: Using Wtd Error	Backpropagation	NetPerfect*	19.49	55.82	30.39
	Backpropagation	Stepwise**	14.74	29.08	24.78
	General Regression NN	N/A	11.38	57.22	43.46
NN2a: % of Wrong Pictures	Backpropagation	NetPerfect	38.00	83.33	51.60
	Backpropagation	Stepwise	28.57	50.00	43.57
	General Regression NN	N/A	19.05	66.67	52.38
NN 2b:	Backpropagation	NetPerfect	16.33	23.36	21.25
	Backpropagation	Stepwise	2.00	28.07	20.25
	General Regression NN	N/A	2.87	20.03	14.88

* - Auto-optimization feature of Neuroshell2

** - Training method by which the number of epochs since minimum error is sequentially increased until the optimum results are obtained.

\+ - Average error = Absolute value of $\frac{\text{(Network Output - Actual Output)}}{\text{Actual Output}}$

++ - Weighted Average Error = (0.3 x Training Error) + (0.7 x Test Error)

can be seen from Table 5.2, this method of training produced a weighted error which was lower than that produced by training conducted with the use of the NetPerfect feature. Also, the choice of the backpropagation architecture over the GRNN architecture was again confirmed, as the GRNN model produced weighted errors in excess of 40%.

In the second method for computing errors, the total number of incorrect pictures of the crack pattern as a percentage of the overall number of crack patterns was calculated to determine the overall effectiveness of the model. These errors were then weighted for the training and test cases as was done in the first method. The second stage of stepwise training again produced the optimum results, however, with this method of analyzing errors, 44% of the predicted crack patterns would be incorrect or 56% of the predicted crack patterns produced the correct pictures. Table 5.2 again summarizes these errors for all levels of training.

Regardless of the method utilized to evaluate the overall error for the neural network, a weighted error of either 24.8% or 44% could not be considered accurate enough to reliably predict a crack pattern for a previously untested concrete slab. Because of these inaccuracies, a new model, NN2b, was proposed with fewer, more quantitative outputs in an effort to further minimize the errors. This model, as shown in Figure 3.4, predicted only the tangential cracking radius and the extent of radial cracking. Although less exact, this model still produced results conforming to an acceptable schematic representation of the final crack pattern formation. Neural networks generally perform better with fewer outputs to predict, as is illustrated by the two load-deflection models. As is indicated by Table 5.2, the weighted errors for the new NN model (NN2b) were substantially less than those reflected in the results for NN2a for all forms of training and architecture. With a weighted error of 14.88%, the general regression neural network produced the optimum results for this model.

Although a sufficient amount of training cases appeared to be available for this network, a high percentage of the training cases (more than 80%) predicted crack patterns indicative of failure due to pure or ductile punching shear. As a result, all of the networks appeared to focus on this type of failure pattern and had difficulty generalizing to other crack patterns. To illustrate this, Figure 5.4 contains network-produced sample crack patterns for punching shear failure and flexural failure as compared to the actual crack patterns encountered during experimental testing.

The GRNN model is designed to predict outputs around the average for the results in the training domain. As a result, this form of neural network was more successful than the more traditional backpropagation model in predicting outputs for this particular group of training and test cases. It is anticipated that, with the addition of further test cases from a wider training domain (ie., a wider variety of crack patterns), the backpropagation neural

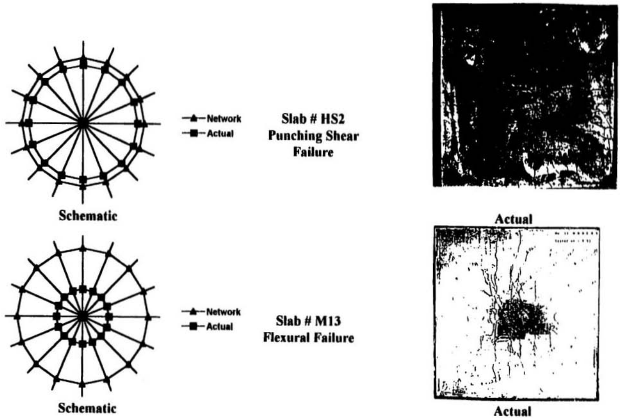


Figure 5.4. NN2b Crack Patterns (Schematic vs. Actual)

network would provide better results, thereby further improving the performance of this model.

5.4 Concrete Strain Distribution

Separate training and testing was conducted for each concrete strain distribution model (NN 3a, which predicted tangential concrete strain distribution at three points along a radius through the slab, and NN3b, which predicted the concrete strains at the column face at four load increments). The optimum NN architecture and training method was then chosen for each model as each was designed to provide mutually exclusive results. Unfortunately, a minimal set of results (nine in total) were available for training and testing both models as the majority of the strain gauges were damaged after cracks started forming in the concrete. Table 5.3 summarizes the errors produced during training and testing for both models. As shown, the errors were quite high for all forms of training and testing except that conducted using the NetPerfect and TurboProp features; the minimum weighted errors produced by these networks were 22.10% for NN3a and 17.26% for NN3b. When plotted against the results from actual experimental tests, these networks were both able to produce results that followed trends encountered by the actual tests as is shown by Figures 5.5 and 5.6. It is anticipated that the results for both networks could only improve with the addition of further experimental data for training and testing the networks.

Table 5.3. Results for Training Concrete Strain Distribution Neural Network (NN3)

Model/ Error Method	Network Architecture	Training Mechanism	Average Error+ on Training Cases (%)	Average Error on Test Cases (%)	Weighted Error++ on all Cases (%)
	Backpropagation	NetPerfect*	5.21	29.35	22.10
NN 3a:	Backpropagation	Stepwise**	7.81	48.02	44.09
	General Regression NN	N/A	40.72	35.28	36.92
	Backpropagation	NetPerfect	12.22	19.42	17.26
NN 3b:	Backpropagation	Stepwise	18.90	26.42	24.16
	General Regression NN	N/A	28.69	40.64	37.05

* - Auto-optimization feature of Neuroshell2

** - Training method by which the number of epochs since minimum error is sequentially increased until the optimum results are obtained.

\+ - Average error = Absolute value of $\frac{\text{(Network Output - Actual Output)}}{\text{Actual Output}}$

++ - Weighted Average Error = (0.3 x Training Error) + (0.7 x Test Error)

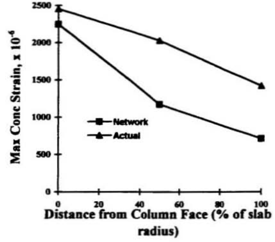


Figure 5.5. NN 3a Maximum Concrete Tangential Strain Distribution for Slab #M12

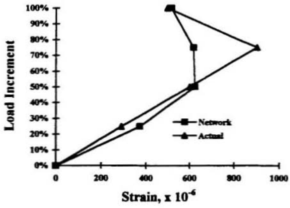


Figure 5.6. NN3b Concrete Strain Distribution at Column Face for Slab # M1

5.5 Steel Strain Distribution

As there was just one model for steel strain distribution, selection of the ideal model only involved choosing the training method and NN architecture which produced results with the minimum error for this model. Training and testing for this network again involved a smaller number of cases than those available for the load-deflection and crack pattern models (nine training cases and three test cases) as strain gauges were again damaged during experimental testing, reducing the data available. The neural network, however, performed well considering this limited amount of available data. Table 5.4 shows the weighted errors for the results for this model; as shown, the network trained with the NetPerfect feature provided the optimum results with an average weighted error of 14.52%. The network results as compared to those obtained during experimental testing for the steel strain distribution through a sample slab in the radial direction is also shown in Figure 5.7. The network predictions follow the actual results quite closely when plotted. While an error within this range can not be considered fully accurate, a reasonable distribution can still be shown and these results can be considered reasonable with respect to the complexity of the problem and the limited number of training cases available.

Table 5.4. Results for Training Reinforcing Steel Strain Distribution Neural Network (NN4)

Network Architecture	Training Mechanism	Average Error⁺on Training Cases (%)	Average Error on Test Cases (%)	Weighted Error⁺⁺ on all Cases (%)
Backpropagation	NetPerfect	13.63	14.91	14.52
Backpropagation	Stepwise	11.69	21.28	18.40
General Regression NN	N/A	22.11	22.58	22.44

* - Auto-optimization feature of Neuroshell2

** - Training method by which the number of epochs since minimum error is sequentially increased until the optimum results are obtained.

\+ - Average error = Absolute value of $\frac{\text{(Network Output - Actual Output)}}{\text{Actual Output}}$

++ - Weighted Average Error = (0.3 x Training Error) + (0.7 x Test Error)

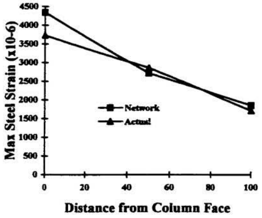


Figure 5.7. NN4 Maximum Steel Strain Distribution for Slab # M11

5.6 Summary

Results for the four neural network models are discussed in this chapter. The combined implementation of these models in a comprehensive spreadsheet tool is discussed in Chapter 6, and conclusions reached from these results are discussed in Chapter 7.

Chapter 6

Spreadsheet for the Prediction of the Structural Behavior of Reinforced Concrete Slabs

6.1 Introduction

This chapter focuses on the development of a spreadsheet which combines the four neural network models into a comprehensive tool that can be used for the structural analysis of reinforced concrete slabs. Given the factors that describe the slab, a spreadsheet can be utilized to predict, through four separate modules, the load-deflection curve, the failure crack pattern, the concrete strain distribution and the reinforcing steel strain distribution for the slab. A user-friendly "interface" sheet guides the operator of the spreadsheet through the four modules for simple and quick predictions which can then be printed for further use.

6.2 Development of the Spreadsheet Model

The spreadsheet was developed as a Microsoft Excel 5.0 Workbook which interfaces with the NeuroShell2 software for neural network predictions. The workbook is divided into seven separate worksheets: an "interface" or main menu sheet, an instructions sheet, an input data sheet and four output sheets which display: the predicted load-deflection curve, the predicted crack pattern at failure, the predicted concrete tangential strain distribution and the predicted strain development at the column face, and the distribution of the maximum reinforcing steel strain distribution in a radial direction. The user moves through the workbook by clicking directly on buttons on the main interface sheet, first by inputting the nineteen factors which describe the slab on the input data sheet, then by moving to each output module to view the NN predictions.

The buttons on the spreadsheet were all customized using the Visual Basic recorder feature of Microsoft Excel 5.0. This feature records the mouse movements of the programmer to a macro which then simulates these movements whenever the button is activated. The neural network predictions for each module were accessed through a Dynamic Link Library (DLL), which executes the trained networks within NeuroShell2. The "CALL" function of Microsoft Excel 5.0 was utilized within cells on the output sheets to call the procedure in the DLL. A separate cell for each output item would then call the

"Predict" function of Excel to open and execute the trained neural network. The following syntax was used in each output cell:

=CALL("NSHELL2.DLL","Predict","pppp","def_path",input_array, output)

where def_path is the file path for the trained NN, input_array is the array of cells which contain the input data, and output is the output node number. The "Chart Wizard" feature of Excel was then utilized to create graphic representations for the NN predictions for each module.

An example problem will serve to further illustrate this spreadsheet tool.

6.3 Example Problem using the Spreadsheet

A reinforced concrete slab from the Emam et al. study (1995) is chosen (Slab # M1), because the results for this slab are known and can be referred to by the reader for comparison purposes. These results were used to train or test all of the four NNs in the current investigation.

Figure 6.1 shows the interface sheet that the user sees when the file is executed. The "Instructions" button can be clicked upon at any time for help in using the spreadsheet, if

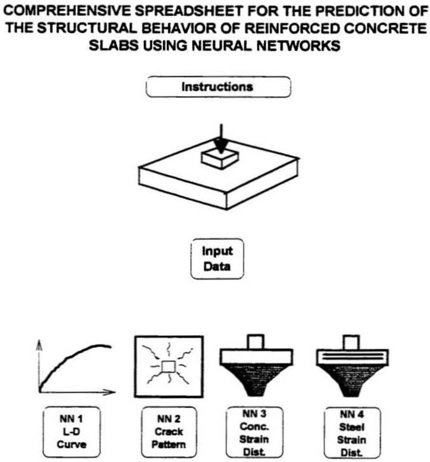


Figure 6.1. Interface Sheet for Example Problem

necessary. The "Input Data" button is first clicked upon for input of the data; this input data screen is shown in Figure 6.2. After inputting and/or editing the data, the user is returned to the interface sheet. The user can then choose any one of the four icons representing each of the NN modules for prediction of the behavior of this example slab. Figures 6.3 through 6.6 show each of the screens that are displayed when each of the buttons is clicked.

The total time spent inputting the data and receiving the results was approximately three minutes. As can be seen, this spreadsheet provides a very quick method by which one could estimate several aspects of the structural behavior of reinforced concrete slabs.

6.4 Comparison of Spreadsheet Predictions with Actual Test Results

The spreadsheet model was next validated by comparing predictions for the ultimate punching load with actual results for selected tests conducted by Elstner and Hognestad (1956), Kinunen and Nylander (1960), Regan et al. (1993) and Hallgren (1996). Representative slabs for each series of tests were chosen and were compared by using the ratio of the spreadsheet predicted punching load divided by actual punching load. The results of this comparison are compiled in Tables 6.1 and 6.2; results for the tests used to train the neural network within the spreadsheet (Marzouk and Hussein, 1991; Emam et al., 1995, and Jiang, 1994) are included in Table 6.3 for comparison.

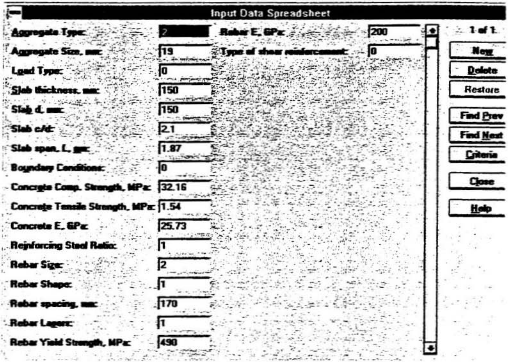
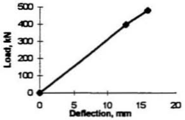


Figure 6.2. Input Data Sheet for Example Problem

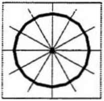
LOAD-DEFLECTION CURVE



Defl., mm	*Load, kN*	
12.8	398	*Yield*
16.0	480	*Ultimate*

Figure 6.3. Predicted Load-Deflection Curve Sheet for Example Problem

CRACK PATTERN PREDICTION



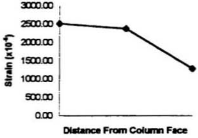
Punching Shear Radius 460
*Extent of Radial Cracking** 4

* 0 - none
1 - 25% through slab
2 - 50% of slab
3 - 75% of slab
4 - throughout slab

Figure 6.4. Predicted Crack Pattern Sheet for Example Problem

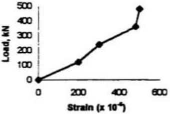
CONCRETE STRAIN DISTRIBUTION

Tangential Strain Distribution



Strain @ Column Face	2516.39
Strain @ Mid Slab	2377.92
Strain @ Slab Edge	1282.51

Strain Distribution at Column Face

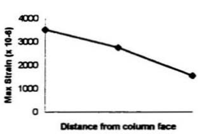


Strain @ 25% Ult. Load	200
Strain @ 50% Ult. Load	300
Strain @ 75% Ult. Load	480
Strain @ Ult. Load	500
*Ultimate Load, kN**	480

* Calculated in NN1 - Load-Deflection Neural Network

Figure 6.5 Predicted Concrete Strain Distribution Sheet for Example Problem

STEEL STRAIN DISTRIBUTION



Yield radius, mm	382.5121
Max. steel strain @ col face	3511.863
Max. steel strain @ mid slab	2749.809
Max. steel strain @ slab edge	1552.465

Figure 6.6 Predicted Steel Strain Distribution Sheet for Example Problem

Table 6.1. Comparison of Spreadsheet Predictions with Actual Test Results

Slab No	Agg Type	d_a, mm	Load Type	t_s, mm	d_s, mm	Slab c/d	L, mm	BC	f_{co}, MPa	f_{ct}, MPa	E_c, GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y, MPa	E_s, GPa	Shear rebar	$P_{u\,act}$, kN	$P_{u\,NN}$, kN	P_{uNN}/P_{uact}
Elstner and Hognestad, 1956																						
A-1(a)	2	15	0	120	116	2.19	1.78	0	14.1	1.07	19.37	1.15	1	1	114	1	332	200	0	302	325	1.08
A-1(b)	2	15	0	120	116	2.19	1.78	0	20.3	1.54	21.86	1.15	1	1	114	1	332	200	0	356	325	0.91
A-1(c)	2	15	0	120	116	2.19	1.78	0	25.2	1.92	23.57	1.15	1	1	114	1	332	200	0	365	325	0.89
A-1(d)	2	15	0	120	116	2.19	1.78	0	36.8	2.80	27.04	1.15	1	1	114	1	332	200	0	351	415	1.18
A-4	2	15	0	120	116	2.19	1.78	0	26.1	1.98	23.86	1.18	1	1	114	1	332	200	0	400	334	0.84
B-9	2	15	0	120	114	2.23	1.78	0	43.9	3.34	28.90	2.00	1	1	114	1	341	200	0	505	558	1.10
B-11	2	15	0	120	114	2.23	1.78	0	13.5	1.03	19.10	3.00	1	1	114	1	410	200	0	329	434	1.32
B-14	2	15	0	120	114	2.23	1.78	0	50.5	2.42	30.49	3.00	1	1	114	1	325	200	0	578	429	0.74
A-11	2	15	0	120	114	3.12	1.78	0	25.9	1.97	23.80	2.47	1	1	114	1	332	200	0	529	434	0.82
Kinunen and Nylander, 1960:																						
5215	2	15	0	149	117	1.28	1.66	0	33.5	2.55	26.12	0.78	1	1	114	1	450	200	0	229	280	1.22
5233	2	15	0	150	117	1.28	1.66	0	32.5	2.47	25.83	0.75	1	1	114	1	455	200	0	252	273	1.08
5269	2	15	0	153	121	1.24	1.66	0	40.1	3.05	27.92	0.77	1	1	114	1	445	200	0	346	317	0.92
3651	2	15	0	153	127	1.18	1.66	0	32.4	2.46	25.80	0.75	1	1	120	1	457	200	0	192	276	1.44
3467	2	15	0	153	128	1.17	1.66	0	33.4	2.54	26.09	1.47	1	1	60	1	449	200	0	229	372	1.62
3415	2	15	0	150	125	1.2	1.66	0	33.9	2.58	26.23	0.64	1	1	121	1	452	200	0	207	275	1.33
3465	2	15	0	150	124	1.21	1.66	0	33.8	2.57	26.20	1.32	1	1	121	1	450	200	0	272	334	1.23
5107	2	15	0	158	128	2.34	1.66	0	33	2.51	25.97	1.01	1	1	93	1	465	200	0	259	406	1.57
5251	2	15	0	151	119	2.52	1.66	0	32.4	2.46	25.80	1.01	1	1	93	1	445	200	0	412	402	0.98
5281	2	15	0	151	120	2.5	1.66	0	37.6	2.86	27.26	1.08	1	1	93	1	443	200	0	500	446	0.89
5089	2	15	0	155	123	2.44	1.66	0	32.9	2.50	25.94	0.49	1	1	171	1	457	200	0	263	320	1.22
5125	2	15	0	152	127	2.34	1.66	0	34.3	2.61	26.34	0.5	1	1	171	1	470	200	0	338	325	0.96
3390	2	15	0	152	127	2.34	1.66	0	33.7	2.56	26.17	1.2	1	1	74	1	438	200	0	286	440	1.54
3448	2	15	0	153	128	2.34	1.66	0	35.8	2.72	26.76	2.39	1	1	37	1	443	200	0	362	535	1.48
3408	2	15	0	150	125	2.4	1.66	0	33	2.51	25.97	0.94	1	1	90	1	442	200	0	305	405	1.33
3436	2	15	0	153	130	2.31	1.66	0	34.3	2.61	26.34	2.07	1	1	35	1	461	200	0	387	521	1.35

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{co} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity, $P_{u\,act}$ = actual ultimate punching load, $P_{u\,NN}$ = network predicted ultimate punching load

Table 6.2. Comparison of Spreadsheet Predictions with Actual Test Results

Slab No	Agg Type	d_a, mm	Load Type	t_s, mm	d_s, mm	Slab c/d	L, mm	BC	f_{cc}, MPa	f_{ct}, MPa	E_c, GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y, MPa	E_s, GPa	Shear rebar	$P_{u\,act}$, kN	$P_{u\,NN}$, kN	P_{uNN}/P_{uact}
Regan et al, 1993:																						
1	2	10	0	120	98	1.38	1.53	0	88	5.3	38.04	0.58	1	1	100	1	550	200	0	224	201	0.90
2	2	10	0	120	98	1.38	1.53	0	56	4.3	31.74	0.58	1	1	100	1	550	200	0	212	211	1.00
3	2	10	0	120	98	1.38	1.53	0	27	2.60	24.15	0.58	1	1	100	1	550	200	0	169	178	1.05
4	2	10	0	120	98	1.38	1.53	0	58	4.6	32.18	0.58	1	1	100	1	550	200	0	233	211	0.91
5	2	12	0	120	98	1.38	1.53	0	54	3.3	31.30	0.58	1	1	100	1	550	200	0	190	208	1.09
6	2	10	0	120	98	1.38	1.53	0	102	5.4	40.43	0.58	1	1	100	1	550	200	0	233	201	0.86
21	2	20	0	120	98	1.38	1.53	0	42	2.016	28.42	1.28	1	1	100	1	550	200	0	286	177	0.62
22	2	20	0	120	98	1.38	1.53	0	84	6.8	37.33	1.28	1	1	100	1	550	200	0	405	276	0.68
23	2	20	0	120	100	1.35	1.53	0	56	4.5	31.74	0.87	1	1	100	1	550	200	0	341	266	0.78
24	2	6	0	120	98	1.38	1.53	0	45	3.3	29.17	1.28	1	1	100	1	550	200	0	270	246	0.91
25	2	10	0	120	100	1.35	1.53	0	33	2.508	25.97	1.27	1	1	100	1	550	200	0	244	197	0.81
26	2	20	0	120	100	1.35	1.53	0	38	2.888	27.37	1.27	1	1	100	1	550	200	0	294	223	0.76
27	2	20	0	120	102	1.32	1.53	0	34	2.584	26.26	1.03	1	1	100	1	550	200	0	227	209	0.92
Hallgren, 1996																						
HSC0	2	19	0	225	200	2.1	2.1	0	90.3	4.334	38.45	0.8	1	1	100	1	490	200	0	965	477	0.49
HSC1	2	19	0	225	200	2.1	2.1	0	91.3	4.382	38.62	0.8	1	1	100	1	490	200	0	1021	405	0.40
HSC2	2	19	0	225	194	2.1	2.1	0	85.7	4.114	37.63	0.82	1	1	100	1	490	200	0	889	477	0.54
HSC4	2	19	0	225	200	2.1	2.1	0	91.6	4.397	38.68	1.19	1	1	100	1	490	200	0	1041	504	0.48
HSC6	2	19	0	225	201	2.1	2.1	0	109	5.222	41.53	0.6	1	1	100	1	490	200	0	960	450	0.47
HSC9	2	19	0	225	198	2.1	2.1	0	84.1	4.037	37.35	0.33	1	1	100	1	490	200	0	565	451	0.80

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{cc} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity, $P_{u\,act}$ = actual ultimate punching load, $P_{u\,NN}$ = network predicted ultimate punching load

Table 6.3. Comparison of Spreadsheet Predictions with Actual Test Results

Slab No	Agg Type	d_a, mm	Load Type	t_s, mm	d_s, mm	Slab c/d	L, mm	BC	f_{co}, MPa	f_{ct}, MP	E_c, GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y, MPa	E_s, GPa	Shear rebar	$P_{u\,act}$, kN	$P_{u\,NN}$, kN	P_{uNN}/P_{uact}
Marzouk & Hussein, 1991																						
HS2	1	20	0	120	95	1.58	1.7	0	70	3.4	34.68	0.84	1	1	125	1	490	200	0	249	265	1.06
HS3	1	20	0	120	95	1.58	1.7	0	69	3.3	34.48	1.47	1	1	71.4	1	490	200	0	356	357	1.00
HS4	1	20	0	120	90	1.67	1.7	0	66	3.2	33.87	2.37	2	1	93.7	1	490	200	0	418	417	1.00
HS5	1	20	0	150	95	1.58	1.7	0	68	3.3	34.28	0.64	1	1	125	1	490	200	0	365	398	1.09
HS6	1	20	0	150	120	1.25	1.7	0	70	3.4	34.68	0.94	1	1	125	1	490	200	0	489	489	1.00
HS7	1	20	0	120	95	1.58	1.7	0	74	3.6	35.46	1.19	1	1	88.2	1	490	200	0	356	358	1.01
HS8	1	20	0	150	120	1.25	1.7	0	69	3.3	34.48	1.11	2	1	150	1	490	200	0	436	430	0.99
HS9	1	20	0	150	120	1.25	1.7	0	74	3.6	35.46	1.61	2	1	100	1	490	200	0	543	556	1.02
HS10	1	20	0	150	120	1.25	1.7	0	80	3.8	36.59	2.33	2	1	71.4	1	490	200	0	645	645	1.00
HS11	1	20	0	90	70	2.14	1.7	0	70	3.4	34.68	0.95	1	1	150	1	490	200	0	196	164	0.83
HS12	1	20	0	90	70	2.14	1.7	0	75	3.6	35.65	1.52	1	1	93.8	1	490	200	0	258	246	0.95
HS13	1	20	0	90	70	2.14	1.7	0	68	3.3	34.28	2	1	1	71.4	1	490	200	0	267	287	1.07
HS14	1	20	0	120	95	2.32	1.7	0	72	3.5	35.07	1.47	1	1	71.4	1	490	200	0	498	474	0.95
HS15	1	20	0	120	95	3.16	1.7	0	71	3.4	34.87	1.47	1	1	71.4	1	490	200	0	560	549	0.98
HS16	1	20	0	120	95	1.58	1.7	0	42	3.2	28.42	1.47	1	1	71.4	1	490	200	0	320	315	0.98
HS17	1	20	0	150	119	2.1	1.95	0	67	3.2	34.08	1.09	2	1	166	1	490	200	0	511	517	1.01
Jiang, 1994																						
HS17	1	20	0	150	120	1.25	1.7	0	30	2.3	25.08	0.94	1	1	125	1	490	200	0	396	336	0.85
HS18	2	19	0	150	119	2.1	1.95	0	68	3.3	34.28	1.09	2	1	166	1	490	200	1	512	469	0.92
HS20	2	19	0	150	119	2.1	1.95	0	74	3.6	35.46	1.09	2	1	166	1	490	200	2	482	544	1.13
HS21	2	19	0	150	119	2.1	1.95	0	72	3.5	35.07	1.09	2	1	166	1	490	200	3	610	584	0.96
HS22	2	19	0	150	119	2.1	1.95	0	60	2.9	32.62	1.09	2	1	166	1	490	200	4	605	611	1.01
HS23	2	19	0	150	111	2.25	1.95	0	60	2.9	32.62	1.09	2	1	166	1	490	200	5	590	629	1.07
Emam et al., 1995																						
M1	2	19	0	150	150	2.1	1.87	0	32.16	2.4	25.73	1	2	1	170	1	490	200	0	476	431	0.91
M2	2	19	0	150	119	2.1	1.87	0	37.21	2.8	27.15	1	2	1	170	1	490	200	0	485	520	1.07
M3	2	19	2	150	115	2.17	1.87	0	43.2	3.3	28.72	0.5	1	1	170	1	490	200	0	266	278	1.05
M4	2	19	2	150	119	2.1	1.87	0	42.68	3.2	28.59	1	2	1	170	1	490	200	0	408	376	0.92
M5	2	19	2	150	119	2.1	1.87	0	36.2	2.8	26.88	1	2	1	170	1	490	200	0	164	244	1.49
M6	2	19	2	150	125	2	1.87	0	34.02	2.6	26.26	0.5	1	1	170	1	490	200	0	164	164	1.00
M7	2	19	2	150	119	2.1	1.87	0	35.26	2.7	26.61	1	2	1	170	1	490	200	0	245	228	0.93
M8	2	19	0	150	119	2.1	1.87	0	67.16	3.2	34.11	1	2	1	170	1	490	200	0	512	480	0.94
M9	2	19	2	150	125	2	1.87	0	73.98	3.6	35.46	0.5	1	1	170	1	490	200	0	200	217	1.08
M10	2	19	2	150	119	2.1	1.87	0	73.82	3.5	35.42	1	2	1	170	1	490	200	0	262	299	1.14

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{co} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity, $P_{u\,act}$ = actual ultimate punching load, $P_{u\,NN}$ = network predicted ultimate punching load

As shown, the spreadsheet provided excellent predictions for the ultimate punching loads for all of the Elstner and Hognestad (1956) slabs except for slabs B-9 and B-14. Both of these slabs, however contained a reinforcing steel ratio of 3.0, which was higher than the reinforcing steel ratio for any of the slabs (Marzouk and Hussein, 1991; Jiang, 1994; and Eman et al., 1995) used to train the neural network within the spreadsheet (NN1b). The spreadsheet also provided satisfactory predictions for many of the Kinunen and Nylander (1960) slabs as well as the Regan et al. (1993) slabs. Again, in the cases for which predictions exceeded 25% of the actual loads reached (ratio higher than 1.25 or lower than .75), at least one input parameter was outside or at the boundaries of the range or domain of the parameters for the slabs used to train the neural network within the spreadsheet. For example, the ratio for slab # 3390 (Kinunen and Nylander, 1960) was 1.54; for this particular slab, the spacing of the reinforcing steel was 74 mm, which was just above the minimum spacing (71.4 mm for slab # HS3, Marzouk and Hussein, 1991) of the slabs used to train NN1b. Finally, rather poor predictions were provided for the Hallgren (1996) slabs; however, the concrete compressive strength for most of these slabs was well in excess (at least 20 MPa) of the concrete compressive strength for the slabs used to train the neural network contained within the spreadsheet. Excellent predictions were provided for all of the slabs within the Marzouk and Hussein (1991), Jiang (1994) and Emam et al. (1995) slabs because these were the slabs which were used to train and test the neural network embedded within the spreadsheet. The results from these comparisons therefore suggest that the spreadsheet model performs well when presented with slabs within the domain of the slabs used to train the neural networks within the spreadsheet.

6.5 Summary

This chapter has described the development and implementation of a spreadsheet model which can be used by structural engineers for preliminary prediction of the structural behavior of a reinforced concrete slab. The spreadsheet combines the four previously developed neural network models into one simple-to-use tool which can provide predictions in minutes. The capabilities of the spreadsheet are demonstrated through an example problem, and the accuracy of its predictions with respect to the ultimate punching load of a slab are established by comparison with the results of four separate series of previously performed experimental tests on reinforced concrete slabs.

Chapter 7

Summary and Conclusions

7.1 Summary

The investigation under consideration was conducted to evaluate the feasibility of using a branch of artificial intelligence known as neural networks to predict several aspects of the structural behavior of reinforced concrete slabs. This technique was examined because, in previous studies, neural networks have been found to be a quick and reliable alternative to lengthy experimental testing or detailed calculations. Four separate neural networks (NN1b: load-deflection; NN2b: crack pattern; NN3a & NN3b: concrete strain distribution; and NN4: reinforcing steel strain distribution) were developed using a variety of models and training techniques for each network in an attempt to seek the optimum neural network that could be constructed for the problem under consideration. One neural network software program, NeuroShell2, was utilized for modeling, training and testing of all of the neural networks in order to achieve consistency of results for comparison purposes. All four neural networks were trained and tested using the results from three series of experimental tests conducted at Memorial University of Newfoundland which evaluated the behavior of normal and high strength concrete slabs subjected to concentrated, flexural and cyclic loading conditions. In addition, all four neural networks considered the same input data, which consisted of a number of variables (grouped under slab geometric dimensions, aggregate properties, concrete properties, reinforcing steel properties, and loading and boundary conditions) which could affect the behavior of the slabs. NN1 predicted the load-deflection behavior of reinforced concrete slabs in two ways; one model predicted the deflection at ten load increments while the other model predicted the yield and ultimate loads and deflections. Next, NN2 predicted the crack pattern at failure of the concrete slabs using two approaches. The first approach predicted a schematic representation of the crack pattern whereas the second approach predicted just two numerical aspects of the final crack pattern. The third neural network, NN3, predicted the distribution of concrete strains throughout the slab through two versions. The first version predicted the maximum concrete strain distribution at three points along a radius of the slab while the second version predicted the concrete strains at the column face at various load increments. The final neural network, NN4, predicted the distribution of reinforcing steel strains throughout the concrete slab.

Results from the four neural networks, either individually or combined, could provide useful information to a structural engineer regarding the prediction of the behavior of the concrete slab. To facilitate access to this information, a comprehensive spreadsheet tool was developed which included all four of the neural networks in one easy-to-understand format. Results could then be obtained for any or all of the four neural network models, providing valuable information for subsequent design or analysis of reinforced concrete slabs.

7.2 Conclusions

Based on the neural network modeling, training and testing conducted, the conclusions reached regarding the use of neural networks to predict the structural behavior of reinforced concrete slabs are summarized below.

7.2.1 NN1: Load-Deflection Behavior

The load-deflection neural network model was considered to be the trial model which would determine the applicability of the neural network technique for the problem at hand. To this end, the following was determined:

1. The backpropagation technique was the most accurate training algorithm for this neural network model, confirming previous findings that backpropagation appears to be the architecture most suited to problems within the civil engineering realm, due to the simplicity of the architecture.

2. The errors produced were higher for the neural network (NN1a) with the greater number of outputs to predict, leading to the conclusion that neural networks are more accurate when fewer outputs are predicted. The reason for this may be that, with fewer ouputs, the connections within the neural network layers would be less complicated, therefore making it easier for the neural network to determine the correct connection weights, thus providing a lower error for the problem.

3. The optimum neural network model was able to produce results with an overall error of 10.48%, which can be considered a reliable approximation to those produced either by experimental testing or mathematical calculations.

4. The load-deflection curves produced by the neural network models closely matched those produced during experimental testing, again confirming the suitability of the neural network technique as a reliable alternative to such testing.

7.2.2 NN2: Crack Pattern at Failure

From the results produced by this neural network, the following conclusions can be drawn:

1. Mapping crack patterns using binary numbers to indicate the exact locations of cracks is a complex task which is not suitable for neural network modeling due to the high number of outputs which are required.

2. The network that predicted the quantitative (analog) outputs, NN2b, produced an overall error that was substantially less than that produced by the network that predicted the classified (binary) outputs. This suggests that, for this particular problem, the neural network models were more accurate for predicting those outputs which could be quantified rather than classified.

3. The General Regression Neural Network (GRNN) architecture produced the optimum results for this model due to the "clustering" of data in the training domain around the same values. This confirms the suitability of this neural network type for predicting results for problems where the results are similar for each training case presented to the network. It is anticipated that, because the backpropagation neural network was more able to provide better generalization abilities in the other neural network models for this problem (NN1, NN2 and NN4), this form of architecture could provide improved results with the addition of training cases with a wider variety of crack patterns.

7.2.3 NN3: Concrete Strain Distribution and NN4: Steel Strain Distribution

Despite the minimum number of cases available for training and testing of both NN3 and NN4, the neural networks were still able to predict results that differed by 17.26% and 14.52%, respectively from those produced during experimental testing. Similarly, NNs 3 and 4 were able to predict strain distribution curves that almost matched those produced during experimental testing. The following conclusions can also be drawn from these results:

1. Neural networks can, given sparse training data, predict results for cases previously unknown to the network that generally concur with known results.

2. The backpropagation technique again provided results with errors that were in the most acceptable range.

3. As a result of the improved network performance encountered when further training data was added to NN1, the addition of training data will most likely improve the results of these networks in a similar manner.

7.2.4 General Conclusions

In summary, the following conclusions can be drawn from the combined results for all four neural networks:

1. The backpropagation technique is the most reliable form of neural network architecture for the problem at hand, except when the results for training cases "cluster" around an average; in this case, the GRNN architecture is most suitable.

2. Neural networks perform best when a minimal number of outputs are predicted by the model.

3. The neural network models predicted results with the minimum errors when they were presented with test cases within the domain of the training cases, especially when a minimal number of cases was used to train the neural network model.

4. Neural networks can be used as a reliable alternative to costly experimental testing as well as lengthy empirical calculations for predicting the structural behavior of reinforced concrete slabs.

5. Simple spreadsheets are powerful tools that can be used to illustrate and summarize vast amounts of data.

7.3 Opportunities for Further Research

While the effectiveness of the neural network technique has been conclusively demonstrated by the work contained within this thesis, further research could serve to enhance this effectiveness. For example, the addition of results from a greater number of reinforced concrete slabs with a wider variety of properties and loading conditions will most likely serve to further improve the accuracy of the neural network models already developed. These neural networks could be expanded to predict results for a wider variety of reinforced concrete structural elements such as beams, columns and shear walls. From this, a general comprehensive tool for the prediction of the structural behavior of reinforced concrete could then be developed for general use by structural engineers as a quick and reliable alternative to existing methods of prediction.

References

Adeli, H., and Park, H. S., (1995). "Counterpropagation Neural Networks in Structural Engineering." *Journal of Structural Engineering*, ASCE, 121(8), 1205-1212.

Arslan, A., and Ince, R., (1994). "The Neural Network Based Design of Edge Supported Reinforced Concrete Slabs." *Artificial Intelligence and Object Oriented Approaches for Structural Engineering*, CIVIL-COMP PRESS, 91-97.

Bailey, D., and Thompson, D., (1990a). "How to Develop Neural-Network Applications." *AI Expert*, June 38-47.

Bailey, D., and Thompson, D., (1990b). "Developing Neural-Network Applications." *AI Expert*, September, 34-41.

Benjamin, J. R. and Williams, H. A., (1957). "The Behaviour of One-Story Reinforced concrete shear walls." *Proc. ASCE.*, 83, 1254-1-1254-49.

Canadian Standards Association, (1994). "Design of Concrete Structures (CSA A23.3-94)." *Canadian Standards Association*, Rexdale, Ont.

Carpenter, G., and Grossberg, S. (1987). "ART-2: Self-organization of stable category recognition codes for analog input patterns." *Applied Optics*, 26(23).

Carpenter, W. C., and Barthelemy, J-F., (1993). "Common Misconceptions about Neural Networks as Approximators." *Proc., 5th Intl. Conf. on Civ. and Struct. Engrg. Computing & 3rd Intl. Conf. in the Appl. Artif. Intell. Civ. and Struct. Engrg.*, Civil-Comp Limited, Edinburgh, Scotl., 11-17.

Caudill, M., and Butler, C., (1990). *Naturally Intelligent Systems*, MIT Press, Cambridge, Ma.

Chauvin, Y., and Rumelhart, D., ed. (1995). *BACKPROPAGATION: Theory, Architectures, and Applications.* Lawrence Erlbaum Assoc., Inc., Publ, Hillsdale, N.J.

Chen, S. S., and Shah, K., "Neural networks in dynamic analysis of bridges." *Proceedings of the Eighth Conference on Computing in Civil Engineering*, Dallas, Texas, 1992, 1058-65.

Chen, H. M., Tsai, K. H., Qi, G. Z., Yang, J. C. S., and Amini, F., (1995). "Neural Network for Structure Control." *Journal of Computing in Civ. Engrg.*, ASCE, 9(2), 168-176.

Darwin, D., and Pecknold, D. A., (1974). "Inelastic Model for Cyclic Biaxial Loading of Reinforced Concrete." *Civil Engineering Studies, Structural Research Series,* No. 409, University of Illinois, Urbana, Illinois, July, 1974.

dePaiva, H. A. R., and Siess, C. P., (1965). "Strength and Behavior of Deep Beams in Shear." *Journal of the Structural Division, ASCE,* 91(ST5), 19-41.

Elkordy, M. F., Chang, K. C., and Lee, G. C., (1994). "A Structural Damage Neural Network Monitoring System." *Microcomp. in Civ. Engrg.*, 9, 83-96.

Elstner, R.C., and Hognestad, E., (1956). "Shearing Strength of Reinforced Concrete Slabs." *ACI Journal*, 28(1), 29-57.

Emam, M., Marzouk, H. M. and Hilal, M. (1995). "Effect of Concrete Strength on the Behaviour of Slab-Column Connections Subjected to Monotonic and Cyclic Loading." *Engineering and Applied Science Research Report No. 95-002,* Memorial University of Newfoundland, St. John's, Newfoundland, Canada.

Flood, I., and Kartam, N., (1994). "Neural Networks in Civil Engineering." *Journal of Computing in Civ. Engrg.*, ASCE, 8(2), 131-147.

Galletly, G. D., (1952). "Behaviour of Reinforced Concrete Shear Walls Under Static Load." Department of Civil and Sanitary Engineering, MIT.

Garrett, J. H., (1992). "Neural Networks and their Applicability within Civil Engineering." *Proc., 8th Conf. of Comp. in Civ. Engrg.*, ASCE, New York, N.Y., 1155-1162.

Garrett, J. H., Ghaboussi, J., and Wu X., (1992). "Neural Networks." *Expert Systems for Civil Engineers: Knowledge Representation,* ASCE, New York, N.Y., 104-143.

Ghaboussi, J., Garrett, J. H., and Wu, X. (1991). "Knowledge-Based Modeling of Material Behavior with Neural Networks." *Journal of Engrg. Mech.*, ASCE, 117(1), 132-153.

Goh, A. T. C., (1995). "Prediction of Ultimate Shear Strength of Deep Beams Using Neural Networks." *ACI Structural Journal*, 92(1), 28-32.

Hajela, P., and Berke, L., (1991). "Neurobiological Computational Models in Structural Analysis and Design." *Computers & Structures*, 41(4), 657-667.

Hanna, A. S., and Senouci, A. B., (1995). "NEUROSLAB - Neural Network System for Horizontal Formwork Selection." *Can. Journal of Civ. Engrg.*, 22, 785-792.

Hebb, D. O. , (1949). *The Organization of Behavior.* Wiley, New York, N. Y.

Hecht-Nielsen, R. (1987). "Counterpropagation Networks." *Applied Optics*, 26, 979-984.

Hegazy, T. (1993). "Integrated Bid Preparation with Emphases on Risk Assessment Using Neural Networks." *Ph.D. Thesis*, Centre for Building Studies, Concordia University, Montreal, Quebec.

Hegazy, T., Fazio, P., and Moselhi, O., (1994). "Developing Practical Neural Network Applications Using Back-Propagation." *Microcomp. in Civ. Engrg.*, 9, 145-159.

Hegazy, T., Marzouk, H., and Tully, S., (1996). "Predicting Load-Deflection Behavior of Concrete Slabs Using Neural Networks." *Proc., 3rd Canadian Conference on Computing in Civil and Building Engineering.*

Hinton, G., and Sejnowski, T., (1986). "Learning and Relearning in Boltzmann Machines." *Parallel Distributed Processing, Vol. 1.*, MIT Press, Cambridge, Mass., 282-317.

Hopfield, J., (1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proc., Natl. Academy of Sciences*, 79, 2554-2558.

Hsu, T. T. C. and Mo, Y. L., (1985). "Softening of Concrete in Low-Rise Shearwalls." *ACI J. Proc.*, 82(6), 883-889.

Issa, R. R. A., Fletcher, D., and Cade, R. A., (1992). "Predicting Tower Guy Pretension Using a Neural Network." *Proc., 8th Conf. of Comp. in Civ. Engrg.*, ASCE, New York, N.Y., 1074-1081.

Jiang, D., (1994). "Evaluation of Shear Enhancement for High-Strength Concrete Plates." *M.Eng. Thesis*, Memorial University of Newfoundland, St. John's, Newfoundland, Canada.

Kang, H-T., and Yoon, C. J., (1994). "Neural Network Approaches to Aid Simple Truss Design Problems." *Microcomp. in Civ. Engrg.*, 9, 211-218.

Kasperkiewicz, J., Racz, J., and Dubrawski, A., (1995). "HPC Strength Prediction Using Artificial Neural Network." *Journal of Computing in Civ. Engrg.*, ASCE, 9(4), 279-284.

Kinnunen, S., and Nylander, H., (1960). "Punching of Concrete Slabs Without Shear Reinforcement." *Transactions No. 158*, Royal Institute of Technology, Stockholm, Sweden.

Kosko, B., (1987). "Bi-directional Associative Memories." *IEEE Transactions on Systems, Man and Cybernetics*, 18(1), 49-60.

Kupfer, H. B., and Gerstle, K. H. (1973). "Behavior of Concrete Under Biaxial Stresses." *J. of Engineering Mechanics*, ASCE, 99(EM4), 852-866.

Liu, T. C. Y., Nilson, A. H., and Slate, F. O. (1972). "Biaxial Stress-Strain Relations for Concrete." *Journal Struct. Div.*, ASCE, 98(ST5), 1025-1034.

Marzouk, H. and Hussein, A., (1991). "Experimental Investigation on the Behavior of High-Strength Concrete Slabs." *ACI Structural Journal*, 88(6), 701-713.

Messner, J. I., Sanvido, V. E., and Kumara, S. R. T., (1994). "StructNet: A Neural Network for Structural System Selection." *Microcomp. in Civ. Engrg.*, 9, 109-118.

Minsky, M.L. & Pappert, S., (1969). *Perceptrons.* MIT Press, Cambridge, Mass.

Mo, Y. L., and Lin, S. S., (1994). "Investigation of Framed Shearwall Behaviour with Neural Networks." *Magazine of Concrete Research*, 46(169), 289-299.

Mo, Y. L. and Shiau, W. C., (1993). "Ductility of Low-Rise Structural Walls." *Magazine of Concrete Research*, 45(163), 131-138.

Moe, J. (1961). "Shearing Strength of Reinforced Concrete Slabs and Footings under Concentrated Loads." *Development Department Bulletin* No. D47, Portland Cement Association, Skokie, Ill., April, 1961.

Moselhi, O., Hegazy, T., and Fazio, P., (1992). "Potential Applications of Neural Networks in Construction." *Can. Journal of Civ. Engrg.*, 19, 521-529.

Mukherjee, A., and Deshpande, J. M., (1995). "Modeling Initial Design Process Using Artificial Neural Networks." *Journal of Computing in Civ. Engrg.*, ASCE, 9(3), 194-200.

Patel, V. C., McClendon, R. W., and Goodrum, J. W., (1994). "Crack Detection in Eggs Using Computer Vision and Neural Networks." *AI Applications*, 8(2), 21-31.

Pratt, D., and Sansalone, M., (1992). "Impact-Echo Signal Interpretation Using Artificial Intelligence." *ACI Materials Journal*, 89(2), 178-187.

Regan, P.E., "Behaviour of Reinforced Concrete Slabs." *CIRIA Report 89*, 1980.

Regan, P. E., Al-Hussaini A., Ramdane, K. E., Xue, H. Y., (1993). "Behaviour of High Strength Concrete Slabs." *Concrete 2000. Proc. of Int. Conf.*, E&FN Spon., Cambridge, Mass., 761-773.

Ramakrishnan, V. and Ananthanarayanana, V., (1968). "Ultimate Strength of Deep Beams in Shear." *ACI J. Proc.*, 65(2), 87-98.

Rosenblatt, F., (1961). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Spartan Books, Washington, D.C.

Rumelhart, D., Hinton, G., and Williams, R., (1986). "Learning Internal Representations by Error Propagation." *Parallel Distributed Processing, Vol. 1*, MIT Press, Cambridge, Mass., 318-362.

Sankarasubramanian G., and Rajasekaran S., (1996). "Constitutive Modeling of Concrete Using a New Failure Criterion." *Computers & Structures*, 58(5), 1003-1014.

Schliach, J.. Schafer, K. and Jennewein, M., (1980). "Towards a Consistent Design of Reinforced Concrete Structures. " *Journal of the Prestressed Concrete Institute*, 32(3), 74-150.

Siao, W. B., (1993). "Strut-and-Tie Model for Shear Behavior in Deep Beams and Pile Caps Failing in Diagonal Splitting." *ACI Structural Journal*, 90(4), 356-363.

Smith, K. N. and Vantsiotis, A. S., (1982). "Shear Strength of Deep Beams." *ACI Journal, Proc.*, 79(3), 201-213.

Subedi, N. K., (1988). "Reinforced Concrete Deep Beams: A Method of Analysis." *Proc., ICE Journal*, 85, 1-30.

Vanluchene, R. D., and Sun, R., (1990). "Neural Networks in Structural Engineering." *Microcomp. in Civ. Engrg.*, 5, 207-215.

Ward Systems Group, (1990). *NeuroShell User's Manual*, Ward Systems Group, Frederick, MD.

Ward Systems Group, (1994). *NeuroShell 2 User's Manual*, Ward Systems Group, Frederick, MD.

Williams, T. P., Khajuria, A., and Balaguru, P., (1992). "Neural Network for Predicting Concrete Strength." *Proc., 8th Conf. of Comp. in Civ. Engrg.*, ASCE, New York, N.Y., 1082-1088.

Wu, X., and Ghaboussi, J., (1992). "Neural Network-Based Modeling of Composite Material with Emphasis on Reinforced Concrete." *Proc., 8th Conf. of Comp. in Civ. Engrg.*, ASCE, New York, N.Y., 1179-1186.

APPENDIX A

NEURAL NETWORK INPUT DATA

Table A.1.a. Input Data for Load-Deflection Neural Network (NN1) - Training Cases

NN1a Case #	NN 1b Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	L m	BC	f_{co} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
Marzouk and Hussein, 1991:																					
1	1	16	1	20	0	120	95	1.6	1.7	0	42	3.19	28.42	1.5	1	1	71	1	490	200	0
2		1	1	20	0	120	95	1.6	1.7	0	67	3.22	34.08	0.5	1	1	214	1	490	200	0
3	2	2	1	20	0	120	95	1.6	1.7	0	70	3.36	34.68	0.8	1	1	125	1	490	200	0
4	3	7	1	20	0	120	95	1.6	1.7	0	74	3.55	35.46	1.2	1	1	88	1	490	200	0
5	4	3	1	20	0	120	95	1.6	1.7	0	69	3.31	34.48	1.5	1	1	71	1	490	200	0
	5	4	1	20	0	120	90	1.7	1.7	0	66	3.17	33.87	2.4	2	1	94	1	490	200	0
6	6	6	1	20	0	150	12	13	1.7	0	70	3.36	34.68	0.9	1	1	125	1	490	200	0
7	7	8	1	20	0	150	12	13	1.7	0	69	3.31	34.48	1.1	2	1	150	1	490	200	0
8	8	9	1	20	0	150	12	13	1.7	0	74	3.55	35.46	1.6	2	1	100	1	490	200	0
9	9	10	1	20	0	150	12	13	1.7	0	80	3.84	36.59	2.3	2	1	71	1	490	200	0
10	10	12	1	20	0	90	70	2.1	1.7	0	75	3.60	35.65	1.5	1	1	94	1	490	200	0
11	11	13	1	20	0	90	70	2.1	1.7	0	68	3.26	34.28	2	1	1	71	1	490	200	0
12	12	15	1	20	0	120	95	3.2	1.7	0	71	3.41	34.87	1.5	1	1	71	1	490	200	0
Jiang, 1994:																					
13	13	HS17	1	20	0	150	119	2.1	2	0	67	3.22	34.08	1.1	2	1	166	1	490	200	0
14	14	HS18	2	19	2	150	119	2.1	2	0	68	3.26	34.28	1.1	2	1	166	1	490	200	1
15	15	HS20	2	19	2	150	119	2.1	2	0	74	3.55	35.46	1.1	2	1	166	1	490	200	2
16	16	HS21	2	19	2	150	119	2.1	2	0	72	3.46	35.07	1.1	2	1	166	1	490	200	3
17	17	HS22	2	19	2	150	119	2.1	2	0	60	2.88	32.62	1.1	2	1	166	1	490	200	4
Emam et al., 1995:																					
18	18	M2	2	19	0	150	119	2.1	1.9	0	37	1.79	27.15	1	2	1	170	1	490	200	0
19	19	M3	2	19	2	150	115	2.2	1.9	0	43	2.07	28.72	0.5	1	1	170	1	490	200	0
20	20	M4	2	19	2	150	119	2.1	1.9	0	43	2.05	28.59	1	2	1	170	1	490	200	0
21	21	M5	2	19	2	150	119	2.1	1.9	0	36	1.74	26.88	1	2	1	170	1	490	200	0
22	22	M6	2	19	2	150	125	2	1.9	0	34	1.63	26.26	0.5	1	1	170	1	490	200	0
23	23	M7	2	19	2	150	119	2.1	1.9	0	35	1.69	26.61	1	2	1	170	1	490	200	0
24	24	M8	2	19	0	150	119	2.1	1.9	0	67	3.22	34.11	1	2	1	170	1	490	200	0
25	25	M9	2	19	2	150	125	2	1.9	0	74	3.55	35.46	0.5	1	1	170	1	490	200	0
26	26	M10	2	19	2	150	119	2.1	1.9	0	74	3.54	35.42	1	2	1	170	1	490	200	0

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{co} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity

Table A.1.b. Input Data for Load-Deflection Neural Network (NN1) - Test Cases

NN1a Case #	NN 1b Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	L m m	BC	f_{cc} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
Marzouk and Hussein, 1991:																					
1		4	1	20	0	120	90	1.67	2	0	66	3.17	####	2.4	2	1	94	1	490	200	0
2	1	17	1	20	0	150	12	12.5	2	0	30	2.28	####	0.9	1	1	125	1	490	200	0
3	2	5	1	20	0	150	95	1.58	2	0	68	3.26	####	0.6	1	1	125	1	490	200	0
4	3	11	1	20	0	90	70	2.14	2	0	70	3.36	####	1	1	1	150	1	490	200	0
5	4	14	1	20	0	120	95	2.32	2	0	72	3.46	####	1.5	1	1	71	1	490	200	0
Jiang, 1994																					
6	5	HS23	2	19	3	150	111	2.25	2	0	60	2.88	####	1.1	2	1	166	1	490	200	5
Emam et al., 1995																					
7	6	M1	2	19	0	150	150	2.25	2	0	32.2	1.54	####	1	2	1	170	1	490	200	0

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{cc} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity

Table A.2.a. Output Data for Load-Deflection Neural Network (NN1) - Training Cases

NN1a Case #	NN 1b Case #	P_y kN	Δ_y mm	P_u kN	Δ_u mm	$\Delta_{10\%}$ mm	$\Delta_{20\%}$ mm	$\Delta_{30\%}$ mm	$\Delta_{40\%}$ mm	$\Delta_{50\%}$ mm	$\Delta_{60\%}$ mm	$\Delta_{70\%}$ mm	$\Delta_{80\%}$ mm	$\Delta_{90\%}$ mm
Marzouk and Hussein, 1991:														
1	1	265.1	10.9	320	14.6	1.2	2	2.5	4	6.5	7.2	8	10.5	12.5
2		132.5		178	25.4	1	4	5	7.5	8	11	12	16	19
3	2	185.6	10.5	249	17.5	1	2.5	2.5	4	6	8	11	12	14
4	3	216.9	8.09	356	17	1	2.5	4	6	7.5	8	10	11.5	15
5	4	272.7	8.84	356	13.1	1	2	2.6	5	6	7	9	10	11.5
	5	286.6	8.99	418	14.7	1	2.5	4	5	7	8.5	10	11	13
6	6	286.6	8.99	489	14.9	1.1	202	3.5	5	6	6.5	8	10	13
7	7	279.3	6.69	436	13.1	1	2.1	3.4	5	6	7	7.5	9	11
8	8	402.7	7.88	543	10.8	1.1	2.3	3.5	5	6	7	7.7	9	10
9	9	606.4	9.4	645	9.86	1.2	2.4	3.5	4	5	6	7	8	9
10	10	144.2	9.65	258	26.3	1	2.5	5	6	9	11	14	15	17.5
11	11	251.4	14.8	267	16.2	1	2.5	5	6	7	9	10	11	13
12	12	359.6	8.69	560	20.8	2	2.5	4	5	6.5	8	10	13	16
Jiang, 1994:														
13	13	364.4	11	511.1	23.4	1.1	2.4	3.7	5.4	6.5	8.7	10.5	12.9	16.8
14	14	333.8	12.7	511.7	20.2	2.5	5.1	7.6	9.4	10.9	12.1	13.4	15	17.2
15	15	283.4	10.2	481.8	18.2	2	4.3	6.2	7.5	9.1	11.1	12.1	14.3	16.2
16	16	431.4	14.8	609.7	22.9	2.1	5.2	6.8	8.4	10.1	11.9	14.5	17.1	18.1
17	17	440.3	9.1	605.1	18.3	1.1	2.1	3.4	4.3	6.6	7.3	8.9	10.1	13.8
Emam et al., 1995:														
18	18	444	15	484.8	19.3	1.9	3.9	5.3	6.8	8.2	9.5	11	12.7	14.9
19	19	219.8	10.6	266.2	22.5	1	2	3.2	4.3	5.4	7.6	9.4	10.5	14
20	20	279.4	7.76	408.2	15.4	1	1.6	3.3	4.8	5.9	7.2	9.1	10.1	12.3
21	21	130	6.6	163.6	8.8	0.75	1.3	1.9	2.6	3.4	4.4	5.9	6.7	7.3
22	22	62.07	3.7	164.3	11.7	0.9	1.5	2.1	3.8	4.1	4.8	6.2	7.2	9.9
23	23	135	5.3	245.3	10.6	0.8	1.4	1.8	2.2	3.1	4.2	5.1	6.1	7.2
24	24	300.3	11.8	511.5	23.7	1.2	2.5	4.4	5.2	6.9	9.9	11.5	13.9	16.1
25	25	70.12	5.56	200.2	22.3	1.1	1.9	2.9	4.3	4.9	5.2	7.5	8.1	10.1
26	26	137.8	6.8	262.4	18	1.5	2.1	3.3	4.2	5.4	7	8.5	10	12.5

Notations: P_y = Load at first yield of reinforcing steel, P_u = Ultimate slab load, Δ_u = Deflection at ultimate slab load, $\Delta_{n\%}$ = Deflection at n % of ultimate slab load

Table A.2.b. Output Data for Load-Deflection Neural Network (nn1) - Test Cases

NN1a Case #	NN 1b Case #	P_y kN	Δ_y mm	P_u kN	Δ_u mm	$\Delta_{10\%}$ mm	$\Delta_{20\%}$ mm	$\Delta_{30\%}$ mm	$\Delta_{40\%}$ mm	$\Delta_{50\%}$ mm	$\Delta_{60\%}$ mm	$\Delta_{70\%}$ mm	$\Delta_{80\%}$ mm	$\Delta_{90\%}$ mm
Marzouk and Hussein, 1991:														
1		287	9	418	14.7	1	2.5	4	5	7	8.5	10	11	13
2	1	267	7.2	396	13.1	1	2	4	5	6	7	8	9	11
3	2	261	8	365	16.9	1	2.5	2.5	3.5	4	7	8	10	15
4	3	111	8.6	196	27	1	2	4	5	7.5	10	14	16	22.5
5	4	292	8.3	498	17.8	2	2.5	5	6	7	8	11	12.5	16
Jiang, 1994														
6	5	403	8.9	590.1	19.6	1.4	2.3	3.8	4.9	6.7	7.5	9.2	11.2	12
Emam et al., 1995														
7	6	350	14	475.5	22	2.2	4.2	5.9	7.1	8.8	10.1	11.7	14.4	16.5

Notations: P_y = Load at first yield of reinforcing steel, P_u = Ultimate slab load, Δ_u = Deflection at ultimate slab load, $\Delta_{n\%}$ = Deflection at n % of ultimate slab load

Table A.3.a. Input Data for Crack Pattern Neural Network (NN2) - Training Cases

NN2 Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	L mm	BC	f_{cc} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
Marzouk and Hussein, 1991:																				
1	16	1	20	0	120	95	1.6	1.7	0	42	3.19	28.42	1.5	1	1	71	1	490	200	0
2	1	1	20	0	120	95	1.6	1.7	0	67	3.22	34.08	0.5	1	1	214	1	490	200	0
3	7	1	20	0	120	95	1.6	1.7	0	74	3.55	35.46	1.2	1	1	88	1	490	200	0
4	4	1	20	0	120	90	1.7	1.7	0	66	3.17	33.87	2.4	2	1	94	1	490	200	0
5	5	1	20	0	150	95	1.6	1.7	0	68	3.26	34.28	0.6	1	1	125	1	490	200	0
6	6	1	20	0	150	12	13	1.7	0	70	3.36	34.68	0.9	1	1	125	1	490	200	0
7	8	1	20	0	150	12	13	1.7	0	69	3.31	34.48	1.1	2	1	150	1	490	200	0
8	9	1	20	0	150	12	13	1.7	0	74	3.55	35.46	1.6	2	1	100	1	490	200	0
9	10	1	20	0	150	12	13	1.7	0	80	3.84	36.59	2.3	2	1	71	1	490	200	0
10	11	1	20	0	90	70	2.1	1.7	0	70	3.36	34.68	1	1	1	150	1	490	200	0
11	12	1	20	0	90	70	2.1	1.7	0	75	3.60	35.65	1.5	1	1	94	1	490	200	0
12	13	1	20	0	90	70	2.1	1.7	0	68	3.26	34.28	2	1	1	71	1	490	200	0
13	14	1	20	0	120	95	2.3	1.7	0	72	3.46	35.07	1.5	1	1	71	1	490	200	0
14	15	1	20	0	120	95	3.2	1.7	0	71	3.41	34.87	1.5	1	1	71	1	490	200	0
Jiang, 1994:																				
15	HS18	2	19	2	150	119	2.1	2	0	68	3.26	34.28	1.1	2	1	166	1	490	200	1
16	HS19	2	19	0	150	109	2.3	2	0	61	2.93	32.80	1.1	3	1	266	1	490	200	0
17	HS20	2	19	2	150	119	2.1	2	0	74	3.55	35.46	1.1	2	1	166	1	490	200	2
18	HS21	2	19	2	150	119	2.1	2	0	72	3.46	35.07	1.1	2	1	166	1	490	200	3
19	HS22	2	19	2	150	119	2.1	2	0	60	2.88	32.62	1.1	2	1	166	1	490	200	4
20	HS23	2	19	3	150	111	2.3	2	0	60	2.88	32.62	1.1	2	1	166	1	490	200	5
Emam et al., 1995:																				
21	M3	2	19	2	150	115	2.2	1.9	0	43	2.07	28.72	0.5	1	1	170	1	490	200	0
22	M4	2	19	2	150	119	2.1	1.9	0	43	2.05	28.59	1	2	1	170	1	490	200	0
23	M5	2	19	2	150	119	2.1	1.9	0	36	1.74	26.88	1	2	1	170	1	490	200	0
24	M7	2	19	2	150	119	2.1	1.9	0	35	1.69	26.61	1	2	1	170	1	490	200	0
25	M9	2	19	2	150	125	2	1.9	0	74	3.55	35.46	0.5	1	1	170	1	490	200	0
26	M14	2	19	3	150	119	2.1	1.9	0	35	1.70	26.64	1	2	1	170	1	490	200	0
27	M1	2	19	0	150	150	2.3	1.9	0	32	1.54	25.73	1	2	1	170	1	490	200	0
28	M6	2	19	2	150	125	2	1.9	0	34	1.63	26.26	0.5	1	1	170	1	490	200	0

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{cc} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity

Table A.3.b. Input Data for Crack Pattern Neural Network (NN2) - Test Cases

NN1a Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	m m	BC	f_{co} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
Marzouk and Hussein, 1991:																				
1	2	1	20	0	120	95	1.58	2	0	70	3.36	34.68	0.84	1	1	125	1	490	200	0
2	3	1	20	0	120	95	1.58	2	0	69	3.31	34.08	1.47	1	1	71	1	490	200	0
Jiang, 1994																				
3	HS17	1	20	0	150	119	2.1	2	0	67	3.22	34.08	1.09	2	1	166	1	490	200	0
Emam et al., 1995																				
4	M2	2	19	0	150	119	2.1	2	0	37.2	1.79	27.15	1	2	1	170	1	490	200	0
5	M11	2	19	3	150	119	2.1	2	0	72.3	3.47	35.13	1	2	1	170	1	490	200	0
6	M12	2	19	3	150	119	2.1	2	0	75.8	3.64	35.80	0.5	1	1	170	1	490	200	0
7	M13	2	19	3	150	119	2.1	2	0	36.8	1.76	27.03	0.5	1	1	170	1	490	200	0

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{co} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity

Table A.4. Output Data for Crack Pattern Neural Network (NN2) - Training and Test Cases

NN2 Training Case #	Slab #	Punching Radius, mm	Extent of radial cracking
Marzouk and Hussein, 1991:			
1	16	212	4
2	1	425	4
3	7	212	4
4	4	250	4
5	5	425	4
6	6	425	4
7	8	284	4
8	9	250	4
9	10	212	4
10	11	212	4
11	12	212	4
12	13	200	4
13	14	284	4
14	15	200	4
Jiang, 1994:			
15	HS18	488	2
16	HS19	488	2
17	HS20	488	2
18	HS21	800	3
19	HS22	800	3
20	HS23	488	1
Emam et al., 1995:			
21	M3	510	4
22	M4	460	4
23	M5	350	3
24	M7	575	3
25	M9	475	4
26	M14	238	2
27	M1	550	4
28	M6	675	3

NN2 Test Case #	Slab #	Punching Radius, mm	Extent of radial cracking
Marzouk & Hussein, 1991			
1	2	212	4
2	3	425	4
Jiang, 1994:			
3	HS17	488	2
Emam et al., 1995:			
4	M2	550	3
5	M11	475	2
6	M12	575	4
7	M13	238	4

Table A.5. Input Data for Concrete Strain Neural Network (NN3a and NN3b) - Training and Test Cases

TRAINING CASES

NN3a Case #	NN3b Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	L mm	BC	f_{co} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
Marzouk and Hussein, 1991:																					
1		5	1	20	0	150	95	1.6	1.7	0	68	3.26	34.28	0.6	2	1	125	1	490	200	0
	1	8	1	20	0	150	120	1.3	1.7	0	69	3.31	34.48	1.1	2	1	150	1	490	200	0
Emam et al., 1995:																					
2		M1	2	19	0	150	150	1.7	1.9	0	32	1.54	25.73	1	2	1	170	1	490	200	0
	2	M2	2	19	0	150	119	2.1	1.9	0	37	1.79	27.15	1	2	1	170	1	490	200	0
3	3	M4	2	19	2	150	119	2.1	1.9	0	43	2.05	28.59	1	2	1	170	1	490	200	0
4	4	M5	2	19	2	150	119	2.1	1.9	0	36	1.74	26.88	1	2	1	170	1	490	200	0
5	5	M6	2	19	2	150	125	2	1.9	0	34	1.63	26.26	0.5	1	1	170	1	490	200	0
6	6	M7	2	19	2	150	119	2.1	1.9	0	35	1.69	26.61	1	2	1	170	1	490	200	0
	7	M9	2	19	2	150	125	2	1.9	0	74	3.55	35.46	0.5	1	1	170	1	490	200	0
	8	M10	2	19	2	150	119	2.1	1.9	0	74	3.54	35.42	1	2	1	170	1	490	200	0
7		M11	2	19	3	150	125	2	1.9	0	72	3.47	35.13	1	2	1	170	1	490	200	0

TEST CASES

NN3a Case #	NN3a Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	L mm	BC	f_{co} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
	1	14	1	20	0	120	95	2.3	1.7	0	72	3.46	35.07	1.5	1	1	71	1	490	200	0
Emam et al., 1995:																					
	2	M1	2	19	0	150	150	1.7	1.9	0	32	1.54	25.73	1	2	1	170	1	490	200	0
1		M2	2	19	0	150	119	2.1	1.9	0	37	1.79	27.15	1	2	1	170	1	490	200	0
2		M12	2	19	3	150	125	2	1.9	0	76	3.64	35.80	0.5	1	1	170	1	490	200	0

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{cc} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity

Table A.6. Output Data for Concrete Strain Neural Network (NN3) - Training and Test Cases

TRAINING CASES

NN 3a Case #	Slab #	ε_{maxcf} $x10^{-3}$	ε_{maxms} $x10^{-3}$	ε_{maxe} $x10^{-3}$
Marzouk and Hussein, 1991				
1	5	1750	650	350
Emam et al., 1995				
2	M1	2500	2400	2100
4	M4	2970	2760	1710
5	M5	1371	747	260
6	M6	1400	750	250
7	M7	1277	979	360
8	M11	1880	940	520

NN3b Case #	Slab #	$\varepsilon_{25\%}$ $x10^{-6}$	$\varepsilon_{50\%}$ $x10^{-6}$	$\varepsilon_{75\%}$ $x10^{-6}$	$\varepsilon_{100\%}$ $x10^{-6}$
Marzouk and Hussein, 1991					
1	8	500	750	1250	2400
Emam et al., 1995					
2	M2	300	380	480	500
3	M4	300	500	1100	1500
4	M5	250	500	750	1000
5	M6	200	300	850	1000
6	M7	800	1200	1350	1000
7	M9	200	400	900	1700
8	M10	500	1000	1500	1800

TEST CASES

NN 3a Case #	Slab #	ε_{maxcf} $x10^{-3}$	ε_{maxms} $x10^{-3}$	ε_{maxe} $x10^{-3}$
Emam et al., 1995				
3	M2	1580	1520	1500
9	M12	2455	2030	1420

NN3b Case #	Slab #	$\varepsilon_{25\%}$ $x10^{-6}$	$\varepsilon_{50\%}$ $x10^{-6}$	$\varepsilon_{75\%}$ $x10^{-6}$	$\varepsilon_{100\%}$ $x10^{-6}$
Marzouk and Hussein, 1991					
1	14	300	600	800	1600
Emam et al., 1995					
2	M1	290	600	900	500

Notes: ε_{maxcf} = max strain @ column face; ε_{maxms} = max strain @ mid-slab; ε_{maxe} = max strain @ slab edge; $\varepsilon_{n\%}$ = strain @ n% ultimate load

Table A.7. Input Data for Steel Strain Neural Network (NN4) - Training and Test Cases

TRAINING CASES

NN4 Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	L mm	BC	f_{co} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
Marzouk and Hussein, 1991:																				
1	3	1	20	0	120	95	1.6	1.7	0	69	3.31	34.48	1.5	1	1	71	1	490	200	0
2	9	1	20	0	150	120	1.3	1.7	0	74	3.55	35.46	1.6	2	1	100	1	490	200	0
Emam et al., 1995:																				
3	M1	2	19	0	150	150	1.7	1.9	0	32	1.54	25.73	1	2	1	170	1	490	200	0
4	M5	2	19	2	150	119	2.1	1.9	0	36	1.74	26.88	1	2	1	170	1	490	200	0
5	M6	2	19	2	150	125	2	1.9	0	34	1.63	26.26	0.5	1	1	170	1	490	200	0
6	M7	2	19	2	150	119	2.1	1.9	0	35	1.69	26.61	1	2	1	170	1	490	200	0
7	M8	2	19	0	150	119	2.1	1.9	0	67	3.22	34.11	1	2	1	170	1	490	200	0
8	M13	2	19	3	150	125	2	1.9	0	37	1.76	27.03	0.5	1	1	170	1	490	200	0
9	M14	2	19	3	150	125	2	1.9	0	35	1.70	26.64	1	2	1	170	1	490	200	0

TEST CASES

NN4 Case #	Slab #	Agg Type	d_a mm	Load Type	t_s mm	d_s mm	Slab c/d	L mm	BC	f_{co} MPa	f_{ct} MPa	E_c GPa	ρ	Rebar Size	Rebar Shape	s_r, mm	Rebar Layers	f_y MPa	E_s GPa	Shear rebar
Emam et al., 1995:																				
1	M2	2	19	0	150	119	2.1	1.9	0	37	1.79	27.15	1	2	1	170	1	490	200	0
2	M11	2	19	3	150	125	2	1.9	0	72	3.47	35.13	1	2	1	170	1	490	200	0
3	M12	2	19	3	150	125	2	1.9	0	76	3.64	35.80	0.5	1	1	170	1	490	200	0

Notations: d_a = aggregate diameter, t_s = slab thickness, d_s = slab effective depth, L = slab span, f_{cc} = concrete compressive strength, f_{ct} = concrete tensile strength, E_c = concrete modulus of elasticity, ρ = reinforcing steel ratio, s_r = steel spacing, f_y = steel yield strength, E_s = steel modulus of elasticity

Table A.8. Output Data for Steel Strain Neural Network (NN4) - Training and Test Cases

TRAINING CASES

NN4 Case #	Slab #	Yield radius, mm	ε_{maxcf} $x10^{-3}$	ε_{maxms} $x10^{-3}$	ε_{maxe} $x10^{-3}$
Marzouk and Hussein, 1991					
1	3	252	2920	720	330
2	9	344	2990	1850	450
Emam et al., 1995					
3	M1	350	2850	2750	1250
7	M5	200	2710	1708	1145
8	M6	425	3650	2730	2090
9	M7	480	5200	2850	1430
10	M8	580	5000	3250	2000
14	M13	600	3970	2250	1960
15	M14	200	2530	890	1130

TEST CASES

NN4 Case #	Slab #	Yield radius, mm	ε_{maxcf} $x10^{-3}$	ε_{maxms} $x10^{-3}$	ε_{maxe} $x10^{-3}$
Emam et al., 1995					
1	M2	385	3230	2030	1700
2	M11	550	3730	2860	1710
3	M12	550	5590	3520	1240

Notes: ε_{maxcf} = max strain @ column face; ε_{maxms} = max strain @ mid-slab; ε_{maxe} = max strain @ slab edge

APPENDIX B

RESULTS FOR OPTIMALLY TRAINED NEURAL NETWORKS

Table B.1. RESULTS FOR OPTIMUM LOAD-DEFLECTION NEURAL NETWORK (NN1b)

	Yield		Ultimate		Yield		Ultimate		Yield		Ultimate		Yield		Ultimate		
Slab #	Act. Load	Act. Def.	Act. Load	Act. Def.	NN Load	NN Def.	NN Load	NN Def.	Δ Load	Δ Def	Δ Load	Δ Def	Abs Error Load	Abs Error Def.	Abs Error Load	Abs Error Def.	Avg. Abs. Error
TRAINING CASES																	
HS16	265.10	10.93	320.00	14.60	270.397	11.005	314.901	13.436	-5.30	-0.16	5.10	1.16	2.00	1.51	1.59	7.97	3.27
HS2	185.60	10.47	249.00	17.50	167.792	8.7039	264.871	18.574	17.81	1.77	-15.87	-1.07	9.59	16.87	6.37	6.14	9.74
HS7	216.90	8.09	356.00	17.00	248.992	8.9881	358.255	16.692	-32.09	-0.90	-2.26	0.31	14.80	11.10	0.63	1.81	7.09
HS3	272.70	8.84	356.00	13.10	258.578	9.0705	356.858	13.687	14.12	-0.24	-0.86	-0.59	5.18	2.68	0.24	4.48	3.14
HS4	286.60	8.99	418.00	14.74	298.146	9.2784	416.766	12.873	-11.55	-0.29	1.23	1.87	4.03	3.21	0.30	12.66	5.05
HS6	355.80	8.40	489.00	14.90	355.164	8.2723	488.53	14.484	0.64	0.13	0.47	0.42	0.18	1.52	0.10	2.79	1.15
HS8	279.30	6.69	436.00	13.10	274.748	6.8828	430.028	13.184	4.55	-0.19	5.97	-0.08	1.63	2.88	1.37	0.64	1.63
HS9	402.70	7.88	543.00	10.80	430.442	7.7125	556.429	11.155	-27.74	0.17	-13.43	-0.35	6.89	2.13	2.47	3.28	3.69
HS10	606.40	9.40	645.00	9.86	570.857	9.2706	645	10.147	35.54	0.13	0.00	-0.29	5.86	1.38	0.00	2.91	2.54
HS12	144.20	9.65	258.00	26.30	153.967	10.831	245.747	23.49	-9.77	-1.18	12.25	2.81	6.77	12.24	4.75	10.69	8.61
HS13	251.40	14.82	267.00	16.15	225.103	13.513	286.701	18.972	26.30	1.31	-19.70	-2.82	10.46	8.82	7.38	17.46	11.03
HS15	359.60	8.69	560.00	20.80	370.788	8.3717	549.415	20.412	-11.19	0.32	10.58	0.39	3.11	3.66	1.89	1.86	2.63
HS17	364.40	11.00	511.10	23.40	364.414	10.884	517.132	23.555	-0.01	0.12	-6.03	-0.15	0.00	1.05	1.18	0.66	0.72
HS18	333.60	12.70	511.70	20.20	291.075	10.837	469.334	18.6	42.72	1.86	42.37	1.60	12.80	14.67	8.28	7.92	10.92
HS20	283.40	10.20	481.80	18.20	349.313	11.939	544.068	20.281	-65.91	-1.74	-62.27	-2.08	23.26	17.05	12.92	11.43	16.17
HS21	431.40	14.80	609.70	22.90	390.327	11.991	584.305	20.388	41.07	2.81	25.40	2.51	9.52	18.98	4.17	10.97	10.91
HS22	440.30	9.10	605.10	18.30	427.901	9.8017	610.575	17.712	12.40	-0.70	-5.48	0.59	2.82	7.71	0.90	3.21	3.66
M2	444.00	15.03	484.80	19.26	453.642	14.102	520.461	19.668	-9.64	0.93	-35.66	-0.41	2.17	6.17	7.36	2.12	4.46
M3	219.80	10.61	268.20	22.50	210.462	10.385	278.196	22.592	9.34	0.22	-12.00	-0.09	4.25	2.12	4.51	0.41	2.82
M4	279.40	7.76	408.20	15.43	270.758	8.275	375.969	14.81	8.64	-0.52	32.23	0.62	3.09	6.64	7.90	4.02	5.41
M5	130.00	6.60	163.60	8.80	161.122	5.6573	243.981	10.278	-31.12	0.94	-80.38	-1.48	23.94	14.28	49.13	16.79	26.04
M6	62.07	3.70	164.30	11.74	78.0923	4.2696	163.6	11.213	-16.02	-0.57	0.70	0.53	25.81	15.39	0.43	4.49	11.53
M7	135.00	5.30	245.30	10.64	146.566	5.3892	227.564	9.6422	-11.57	-0.09	17.74	0.80	8.57	1.68	7.23	7.50	6.24
M8	300.30	11.80	511.52	23.65	310.589	12.118	480.225	22.875	-10.29	-0.32	31.29	0.77	3.43	2.69	6.12	3.28	3.88
M9	70.12	5.56	200.21	22.32	83.743	5.2941	216.831	21.834	-13.62	0.27	-16.62	0.49	19.43	4.78	8.30	2.18	8.67
M10	137.78	6.80	262.42	18.00	129.599	6.3148	299.323	18.557	8.18	0.49	-36.90	-0.56	5.94	7.13	14.06	3.09	7.56
															Overall Average Error (Training)		6.87
TEST CASES																	
HS17	266.9	7.2	396	13.07	266.032	5.0074	336.391	8.8	0.87	2.19	59.61	4.27	0.33	30.45	15.05	32.67	19.63
HS5	260.5	8.04	365	16.9	275.989	8.7837	398.252	17.283	-15.49	-0.74	-33.25	-0.38	5.95	9.25	9.11	2.27	6.64
HS11	110.9	8.57	196	27	62.07	8.1064	163.6	25.598	48.83	0.46	32.40	1.40	44.03	5.41	16.53	5.19	17.79
HS14	292.4	8.27	498	17.8	330.367	8.7092	474.195	17.067	-37.97	-0.44	23.81	0.73	12.98	5.31	4.78	4.12	6.80
HS23	402.5	8.9	590.1	19.6	445.552	10.856	629.079	20.335	-43.05	-1.96	-38.96	-0.74	10.70	21.96	6.61	3.75	10.76
M1	350	14.3	475.5	21.95	379.662	13.791	431.487	17.332	-29.66	0.51	44.01	4.62	8.47	3.56	9.28	21.04	10.58
															Overall Average Error (Testing)		12.03
															Wtd Abs. Error		10.48

Table B.2. RESULTS FOR OPTIMUM CRACK PATTERN NEURAL NETWORK (NN2b)

TRAINING CASES

Slab #	ActPunchingRed	Act.ExtentRed	NNPunchingRed	NN ExtentRed	Δ PunchingRed	Δ ExtentRed	AbsErrorPunchingRed	Abs Error ExtentRed	Avg Abs Error
HS16	212	4	212.00	4.00	0.00	0.00	0.00	0.00	0.00
HS7	212	4	212.55	4.00	-0.55	0.00	0.26	0.00	0.13
HS4	250	4	243.24	4.00	6.76	0.00	2.70	0.00	1.35
HS6	425	4	425.00	4.00	0.00	0.00	0.00	0.00	0.00
HS8	284	4	283.67	4.00	0.33	0.00	0.11	0.00	0.06
HS9	250	4	250.30	4.00	-0.30	0.00	0.12	0.00	0.06
HS10	212	4	212.02	4.00	-0.02	0.00	0.01	0.00	0.01
HS12	212	4	211.93	4.00	0.07	0.00	0.03	0.00	0.02
HS13	200	4	200.07	4.00	-0.07	0.00	0.04	0.00	0.02
HS15	200	4	200.00	4.00	0.00	0.00	0.00	0.00	0.00
HS18	488	2	488.00	2.00	0.00	0.00	0.00	0.00	0.00
HS19	488	2	488.00	2.00	0.00	0.00	0.00	0.00	0.00
HS20	488	2	488.45	2.00	-0.45	0.00	0.09	0.07	0.08
HS21	800	3	799.55	3.00	0.45	0.00	0.06	0.05	0.05
HS22	800	3	799.85	3.00	0.15	0.00	0.02	0.03	0.03
M3	510	4	510.29	4.00	-0.29	0.00	0.06	0.04	0.05
M4	460	4	448.74	3.81	11.26	0.19	2.45	4.69	3.57
M5	350	3	411.27	2.83	-61.27	0.17	17.50	5.61	11.51
M7	575	3	414.43	2.82	160.57	0.18	27.93	5.98	16.95
M9	475	4	475.51	4.00	-0.51	0.00	0.11	0.06	0.08
M14	238	2	368.71	2.60	-130.71	-0.60	54.92	29.81	42.37
NS1	425	4	422.83	4.00	2.17	0.00	0.51	0.00	0.26
HS5	425	4	425.00	4.00	0.00	0.00	0.00	0.00	0.00
HS11	212	4	212.00	4.00	0.00	0.00	0.00	0.00	0.00
HS14	284	4	284.00	4.00	0.00	0.00	0.00	0.00	0.00
HS23	488	1	488.15	1.00	-0.15	0.00	0.03	0.10	0.06
M1	550	4	535.50	3.82	14.50	0.18	2.64	4.61	3.62
M6	675	3	674.71	3.00	0.29	0.00	0.04	0.06	0.05
							Overall Avg Abs Error (Training)		2.87

TEST CASES

Slab #	ActPunchingRed	Act.ExtentRed	NNPunchingRed	NN ExtentRed	Δ PunchingRed	Δ ExtentRed	AbsErrorPunchingRed	Abs Error ExtentRed	Avg Abs Error
HS2	212	4	238.53	4.00	-26.53	0.00	12.51	0.00	6.26
HS3	425	4	212.00	4.00	213.00	0.00	50.12	0.00	25.06
HS17	488	2	487.93	2.00	0.07	0.00	0.01	0.24	0.13
M2	550	3	484.26	3.41	65.74	-0.41	11.95	13.67	12.81
M11	475	2	488.00	2.00	-13.00	0.00	2.74	0.00	1.37
M12	575	4	430.40	3.98	144.60	0.02	25.15	0.44	12.79
M13	238	4	596.28	3.48	-358.28	0.52	150.54	13.07	81.80
							Overall Avg Abs Error (Testing)		20.03
							Weighted Avg Abs Error		14.88

Table B.3. RESULTS FOR OPTIMUM MAXIMUM CONCRETE STRAIN DISTRIBUTION NEURAL NETWORK (NN3a)

Slab #	ActFace ε	ActMid ε	ActEdge ε	NNFace ε	NNMid ε	NNEdge ε	Δ Face	Δ Mid	Δ Edge	AbsErrorFace	AbsErrorMid	AbsErrorEdge	AvgAbsError
TRAINING CASES													
HS5	2600	1300	800	2602.53	1300.98	801.76	-2.53	-0.98	-1.76	0.10	0.08	0.22	0.13
M1	2500	2400	1200	2502.03	2400.79	1203.28	-2.03	-0.79	-3.28	0.08	0.03	0.27	0.13
M4	2970	2760	1710	2952.95	2738.59	1718.99	17.05	21.41	-8.99	0.57	0.78	0.53	0.63
M5	1371	747	260	1392.73	934.57	342.49	-21.73	-187.57	-82.49	1.58	25.11	31.73	19.47
M6	1400	750	250	1404.03	749.20	253.18	-4.03	0.80	-3.18	0.29	0.11	1.27	0.55
M7	1277	979	360	1250.85	805.94	266.22	26.15	173.06	93.78	2.05	17.68	26.05	15.26
M11	1880	940	520	1890.64	938.64	519.25	-10.64	1.36	0.75	0.57	0.14	0.14	0.29
										Overall Average Absolute Error(Training)			5.21
TEST CASES													
M2	1580	1520	1500	2177.24	1562.10	974.20	-597.24	-42.10	525.80	37.80	2.77	35.05	25.21
M12	2455	2030	1420	2248.44	1172.05	713.16	206.56	857.95	706.84	8.41	42.26	49.78	33.49
										Overall Average Absolute Error(Testing)			29.35
										Weighted Average Absolute Error			22.10

Table B.4. RESULTS FOR OPTIMUM CONCRETE STRAIN DISTRIBUTION AT COLUMN FACE NEURAL NETWORK (NN3b)

Slab #	Actu @25%	Actu @50%	Actu @75%	Actu @LB	NNs @25%	NNs @50%	NNs @75%	NNs @LB	Δ 25%	Δ 50%	Δ 75%	Δ LB	AbsError25%	AbsError50%	AbsError75%	AbsErrorLB	AvgAbsError
TRAINING CASES																	
HS8	600	750	1250	2400	603.28	753.98	1252.30	2400.00	-3.28	-3.98	-2.30	0.00	0.55	0.53	0.18	0.00	0.34
M2	110	250	300	210	200.00	300.00	485.06	501.62	-90.00	-50.00	-185.06	-291.62	81.82	20.00	61.69	138.87	75.64
M4	300	500	1100	1500	297.61	497.67	1089.29	1500.33	2.39	2.33	10.71	-0.33	0.80	0.47	0.97	0.02	0.56
M5	250	500	750	1000	318.72	578.28	852.30	1008.14	-68.72	-78.28	-102.30	-8.14	27.49	15.66	13.64	0.81	14.40
M6	200	300	850	1000	200.00	300.00	854.65	999.40	0.00	0.00	-4.65	0.60	0.00	0.00	0.55	0.06	0.15
M7	800	1200	1350	1000	731.34	1100.93	1254.39	992.45	68.66	99.07	95.61	7.55	8.58	8.26	7.08	0.75	6.17
M9	200	400	900	1700	200.00	396.80	894.40	1698.26	0.00	3.20	5.64	1.74	0.00	0.80	0.62	0.10	0.38
M10	500	1000	1500	1800	498.75	999.58	1497.87	1800.55	1.25	0.42	2.13	-0.55	0.25	0.04	0.14	0.03	0.12
														Overall Avg Error (Training)			12.22
TEST CASES																	
HS14	300	600	800	1800	254.99	652.40	1200.81	1399.21	45.01	-52.40	-400.81	230.79	15.00	8.73	50.09	14.42	22.06
M1	290	600	900	500	372.34	623.36	616.09	510.78	-82.34	-23.36	283.31	-10.78	28.39	3.89	31.49	3.36	16.78
														Overall Avg Error (Testing)			19.42
														Weighted Average Absolute E			17.28

Table B.5. RESULTS FOR OPTIMUM STEEL STRAIN DISTRIBUTION NEURAL NETWORK (NN4)

TRAINING CASES

Slab #	Act YR	Actε@ColFace	Actε@MidSlab	Actε@Edge	NN YR	NNε@ColFace	NNε@MidSlab	NNε@Edge	Δ YR	Δ ε@ColFace	Δ ε@MidSlab	Δ ε@Edge	AbsErrorYR	AbsErrorColFace	AbsErrorMidSlab	AbsErrorEdge	AvgAbsError
HB3	252	2920	720	330	269.40	2789.15	946.22	396.52	-17.40	130.85	-226.22	-66.52	6.91	4.48	31.42	20.16	15.74
HB9	344	2990	1850	450	283.67	3004.44	1629.82	454.82	60.33	-14.44	220.18	-4.82	17.54	0.48	11.90	1.03	7.74
M1	350	2850	2750	1250	321.77	3171.43	2477.21	1269.45	28.23	-321.43	272.79	-19.45	8.07	11.28	9.92	1.56	7.71
M5	200	2710	1708	1145	251.06	3205.05	1816.28	1237.02	-51.06	-495.05	-108.28	-92.02	25.53	18.27	6.34	8.04	14.54
M6	425	3650	2730	2090	478.22	3728.13	2481.93	1972.76	-53.22	-78.13	248.07	117.24	12.52	2.14	9.09	5.61	7.34
M7	480	5200	2850	1430	244.60	3173.24	1779.89	1212.85	235.40	2026.76	1070.11	217.15	49.04	38.98	37.55	15.19	35.19
M8	580	5000	3250	2000	553.04	4669.07	3335.78	1983.19	26.96	330.93	-85.78	16.81	4.65	6.62	2.64	0.84	3.69
M13	600	3970	2250	1960	452.18	3655.44	2201.01	1942.42	147.82	314.56	48.99	17.58	24.64	7.92	2.18	0.90	8.91
M14	200	2530	890	1130	206.66	2988.30	1408.15	1042.92	-6.66	-458.30	-518.15	87.08	3.33	18.11	57.99	7.71	21.79
															Overall Average Error (Training)		13.63

TEST CASES

Slab #	Act YR	Actε@ColFace	Actε@MidSlab	Actε@Edge	NN YR	NNε@ColFace	NNε@MidSlab	NNε@Edge	Δ YR	Δ ε@ColFace	Δ ε@MidSlab	Δ ε@Edge	AbsErrorYR	AbsErrorColFace	AbsErrorMidSlab	AbsErrorEdge	AvgAbsError
M2	365	3230	2030	1700	366.24	3595.91	2682.71	1542.25	18.76	-365.91	-652.71	157.75	4.87	11.33	32.15	9.28	14.41
M11	550	3730	2860	1710	483.92	4344.34	2714.40	1848.91	66.08	-614.34	145.60	-138.91	12.02	16.47	5.09	8.12	10.42
M12	550	5590	3520	1240	590.43	4705.54	3068.39	2090.00	-40.43	884.46	451.61	-850.00	7.35	15.82	12.83	68.55	26.14
															Overall Average Error (Testing)		14.91
															Weighted Average Absolute Error		14.52