THE APPLICATION OF NEURAL NETWORK TECHNIQUES TO THE ANALYSIS OF REINFORCED CONCRETE BEAM-COLUMN JOINTS SUBJECTED TO AXIAL LOAD AND BI-AXIAL BENDING

A Thesis Submitted for the Degree of

DOCTOR OF PHILOSOPHY

of the

UNIVERSITY OF EDINBURGH

by

MANSOUR NASSER JADID

B.S.C.E Degree in Civil Engineering

UNIVERSITY OF WASHINGTON, Seattle Washington, USA in 1981

M.S.C.E Degree in Structural Engineering

 PENNSYLVANIA STATE UNIVERSITY, State College, USA in 1987

Department of Civil Engineering and Building Science

May 1994
DECLARATION

This Thesis is composed by me on the basis of my own work, conducted in the Department of Civil Engineering and Building Science, University of Edinburgh.
I DEDICATE THIS THESIS TO MY FATHER

NASSER IBRAHIM JADID
ACKNOWLEDGEMENTS

People do not believe that life can be different until they experience it differently, and studying under the supervision of Dr. Daniel R. Fairbairn was an excellent and fruitful experience. The author expresses his sincere debt of gratitude to his supervisor Dr. Fairbairn for his constant supervision, his open door policy and encouragement during the study. Also, the author would like to thank Dr. Rodney Royles for giving him the opportunity to work with him as a tutor for undergraduate students in the analysis of finite element computational methods. Thanks is extended to Professor Michael Forde for his effective policy in the Department of Civil Engineering and Building Science.

The scholarship of the King Faisal University is gratefully acknowledged and sincere thanks is owed to the H. E. Principal of King Faisal University, Professor Mohammad Al-Kahtani, for his personal support and encouragement during the study. Thanks is also due to Mr. Abdulrahman Al-Hamdan at the Ministry of Education for his enormous help.

The author also expresses his appreciation to his parents, brothers and sisters. A special thanks is expressed to his father for his constant help, both morally and financially, during his studies.

The author wishes to express his deepest gratitude to his wife Fawzeyyah for her continued support and encouragement during his studies. Thanks are also extended to the author’s children Ziyad, Sousen, Maysoon and Nawaf for their inspiration.

Finally and ultimately, the author praises Allah (God) for the patience and support, without which the making of this thesis could not have been possible.
The application of neural networks in the form of parameter predictions to the behaviour and strength of beam-column joints under axial load and biaxial bending has been studied. Computation algorithms in the form of numerical analysis were performed on the beam-column joints to simulate the existing experimental data. A systematic approach was provided by implementing neural networks in the form of prediction by backpropagation algorithms.

The objective of this study was to demonstrate a concept and methodology, rather than to build a full-scale knowledge-based system model, by incorporating most of the fundamental aspects of a neural network to solve the complex non-linear mapping of a beam-column joint. In general, it should be possible to identify certain parameters and allow the neural network to develop the model, thus accounting for the observed behaviour without relying on a particular algorithm but depending entirely on the manipulation of numerical data.

The aim of this study was to view available experimental data on beam-column joint parameters from different angles and establish a concept and methodology that would provide rapid and economic benefits to experimental research. The focus of this study is to reconstruct previous experimental work by evaluating several parameters and establish valid mathematical relationships based on neural networks which are in agreement with relationships based on the experimental results.

The computational methodology considered for the analysis of the beam-column joints has been formulated by adopting three stages to establish a procedure to implement the concept and methodology proposed. The procedure is demonstrated
by the evaluation of the ultimate flexural strength of the reinforced concrete members, the moment-curvature relationship and the shear strength of the beam-column joint. The results indicate that the implementation of trained and tested neural networks with a supervised training technique is a step forward in the simulation of experimental testing. The study also demonstrates the capability of neural networks to represent n-dimensional space and track each individual characteristic in that separate space. The essential requirement in this approach is the selection of feasible and appropriate training and testing data. The exploration of neural networks within the context of numerical analysis is a fundamental technique which looks beyond the traditional experimental programme and can assist future experimental testing.

The procedure is not limited to reinforced concrete structures, and the methodology can be applied to other areas of study. Research into neural network application has potential advantages for a wide variety of engineering applications, particularly those which seek to use design processes in innovative ways.
TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................................................................................... i
ABSTRACT ........................................................................................................................ ii
TABLE OF CONTENTS ........................................................................................................ iv
LIST OF FIGURES ................................................................................................................. xiv
LIST OF TABLES .................................................................................................................. xviii
NOTATIONS ........................................................................................................................ xxii

CHAPTER 1 ............................................................................................................................ 1

INTRODUCTION .................................................................................................................... 1

1.1. GENERAL INTRODUCTION .................................................................................. 1
1.2. BEAM-COLUMN JOINT ..................................................................................... 3
1.2.1. Literature Review ............................................................................................. 5
1.3. COMPUTATIONAL TECHNOLOGY PARADIGMS ............................................ 8
1.3.1. Symbolic Manipulation ..................................................................................... 9
1.3.2. Sequential Programming Instruction ............................................................ 10
1.3.3. Parallel Distributed Processing ....................................................................... 10
1.4. SCOPE OF THE RESEARCH ............................................................................ 12
1.5. METHODOLOGY AND OUTLINE OF RESEARCH ..................................... 14
CHAPTER 2...................................................................................................... 21

SYMBOLIC LOGIC BASE LEARNING............................................................... 21

2.1. ARTIFICIAL INTELLIGENCE: GENERAL VIEW ........................................... 21

2.2. EXPERT SYSTEMS : AN OVERVIEW ....................................................... 23

2.3. MEANING OF LEARNING........................................................................... 25

2.4. MACHINE LEARNING BACKGROUND.................................................... 28

2.5. MACHINE LEARNING STRATEGIES...................................................... 31

2.5.1. Rote Learning .......................................................................................... 32

2.5.1.1. Learning by Being Programmed ........................................................... 32

2.5.1.2. Learning by Memorisation .................................................................. 33

2.5.2. Learning by Instruction or Learning by Being Told ............................... 33

2.5.3. Learning by Analogy .............................................................................. 33

2.5.4. Learning by Induction ............................................................................ 34

2.5.5. Learning from Examples - Supervised Learning .................................... 35

2.5.6. Learning by Observation and Discovery - Unsupervised Learning ....... 36

2.5.7. Learning by Deduction ........................................................................... 36

2.6. INDUCTIVE LEARNING ALGORITHMS ................................................ 36

2.6.1. Concept Learning System (CLS) Algorithm ......................................... 37

2.6.1.1. Quinlan's ID3 Algorithm .................................................................... 38

2.6.1.1.1. Expert-Ease .................................................................................... 39

2.6.1.1.2. EX-TRAN 7 .................................................................................. 40

2.6.1.1.3. RuleMaster .................................................................................... 40

2.6.1.1.4. C4 ................................................................................................. 41
CHAPTER 3

PARALLEL DISTRIBUTED PROCESSING BASE LEARNING

3.1. INTRODUCTION

3.2. HISTORICAL BACKGROUND

3.3. BIOLOGICAL NEURAL NETWORKS

3.3.1. Brain Structure

3.3.2. Function of the Brain

3.4. LEARNING ALGORITHMS TECHNIQUES

3.4.1. Supervised Learning

3.4.2. Unsupervised Learning

3.4.3. Reinforcement Learning
3.5. NEURAL NETWORK TRANSFER FUNCTIONS .................................. 58
3.6. NEURAL NETWORK ARCHITECTURE ........................................... 59
3.7. PERCEPTRON NETWORK .......................................................... 61
3.7.1. Basic Rule ................................................................................ 62
3.8. ADALINE AND MADALINE NETWORKS .................................... 63
3.8.1. Adaline Rule and Training ....................................................... 65
3.9. HOPFIELD NETWORK ................................................................. 66
3.9.1. Learning and Energy Concept ................................................ 67
3.10. BACKPROPAGATION ALGORITHM .......................................... 70
3.10.1. General Overview ................................................................. 70
3.11. BASIC MATHEMATICAL FORMULATION .................................. 73
3.11.1. Output Layer Weights Adjustment ....................................... 74
3.11.2. Hidden Layer Weight Adjustment ........................................ 76
3.12. KOHONEN'S SELF-ORGANISING NETWORK .......................... 77
3.12.1. Network Structure ............................................................... 77
3.12.2. Kohonen Training Rule ......................................................... 78
3.13. NEURAL NETWORKS DOMAINS APPLICATION ........................ 83
3.14. NEURAL NETWORKS IN CIVIL ENGINEERING .......................... 86

CHAPTER 4 .................................................................................. 89

BEAM-COLUMN JOINTS DIGITAL REPRESENTATIONS ...................... 89
4.1. INTRODUCTION .......................................................................... 89
4.2. HARDWARE AND SOFTWARE .................................................. 91
4.3. REVIEW OF PREVIOUS EXPERIMENTAL MODEL ...................... 93
4.4. NUMERICAL COMPUTATION PARADIGM ................................................. 103
4.5. NEURAL NETWORKS DEVELOPMENT TOOLS ...................................... 104
4.5.1. Floating Down Menu ........................................................................ 105
4.5.2. Working Area Interface ................................................................... 106
4.5.3. Fly-Out Boxes .................................................................................. 106
4.6. NEURAL NETWORK PRIMARY REQUIREMENTS .................................... 107
4.6.1. Generation of Data ........................................................................... 107
4.6.2. Scaling of Data .................................................................................. 107
4.6.3. Selection of Network ....................................................................... 108
4.6.3.1. Architecture ................................................................................ 109
4.6.3.2. Topology ..................................................................................... 108
4.6.3.3. Neurodynamics ............................................................................ 109
4.6.4. Training and Testing the Network ..................................................... 111
4.6.5. Network Performance ..................................................................... 111
4.6.6. Network Parameters Optimisation ................................................... 113
4.6.7. Deployment of Trained Network ....................................................... 114
4.6.8. Monitoring and Maintenance of Trained Networks ............................ 114
4.7. COMPUTATIONAL METHODOLOGY CONSIDERATIONS ...................... 116
4.8. APPLICATION OF NEURAL NETWORKS IN PREDICTION .................... 117
4.8.1. Preparatory Stage ........................................................................... 118
4.8.2. Renew Stage .................................................................................... 119
4.8.3. Final Stage ....................................................................................... 121

CHAPTER 5 ............................................................................................... 124
ULTIMATE FLEXURAL STRENGTH OF REINFORCED CONCRETE MEMBERS

5.1. INTRODUCTION .............................................................................. 124

5.2. STEEL REINFORCEMENT PROPERTIES ..................................... 125

5.3. CONCRETE TENSILE STRENGTH ................................................ 126

5.3.1. Prediction of Concrete Strength Versus Modulus of Rupture .......... 127

5.4. RECTANGULAR SECTIONS UNDER FLEXURAL STRESSES .......... 130

5.4.1. Concrete Modulus of Elasticity .................................................. 131

5.4.1.1. Prediction of Relationship between Concrete Strength and .... .... 133

5.4.2. Concrete Strain Characteristics at Maximum Stress ..................... 138

5.4.3. Concrete Strain Characteristics at Ultimate Stress ....................... 139

5.4.4. Concrete Stress-Strain Relationship ............................................. 140

5.4.4.1. Prediction of Stress Versus Strain ......................................... 147

5.5. STRESS-BLOCK COMPUTATION FOR FLEXURAL MEMBERS .......... 158

5.5.1. Stress-Block Computation Adopted by Nirjar for Flexural Members .................................................................................................................. 159

5.5.2. Stress-Block Computation Adopted by ACI Code for Flexural Members .................................................................................................................. 163

5.5.3. Stress-Block Computation at Preparatory Stage for Flexural Members .................................................................................................................. 168
CHAPTER 5

5.5.4. Stress-Block Computation at Renew Stage for Flexural Members.... 178
5.6. CONFINED PROPERTIES OF FLEXURAL MEMBERS.............. 182
5.7. CONCRETE BIAXIAL STRESS-STRAIN RELATIONSHIP............. 186

CHAPTER 6

THE MOMENT-CURVATURE RELATIONSHIP OF A FLEXURAL
MEMBER...................................................................................................... 189

6.1. INTRODUCTION............................................................................. 189
6.2. RADIUS OF CURVATURE............................................................. 191
6.3. RELATIONSHIP BETWEEN MOMENT AND CURVATURE...... 195
6.3.1. Cracking Moment and Curvature............................................... 198
6.3.2. Yield Moment and Curvature..................................................... 202
6.3.3. Preparatory Stage for Yield Moment and Curvature............... 207
6.3.4. Moment and Curvature Relationship at Ultimate................. 214
6.3.5. Preparatory Stage for Ultimate Moment and Curvature......... 221
6.3.6. Renew Stage for Yield Moment and Curvature ..................... 227
6.3.7. Renew Stage for Ultimate Moment and Curvature ............... 238
6.4. MOMENT-CURVATURE RELATIONSHIP FOR THE
CONFINED CROSS-SECTION.......................................................... 246

CHAPTER 7

SHEAR BEHAVIOUR AND STRENGTH CHARACTERISTICS.............. 249

7.1. INTRODUCTION............................................................................. 249
7.2. SHEAR STRESSES CONCEPT..................................................... 251
7.3. FUNDAMENTAL SHEAR TRANSFER FAILURE......................... 254
7.3.1. Concrete Failure Criteria ................................................................. 254
7.3.1.1. Mohr's Circle Failure Criteria ...................................................... 255
7.3.1.2. The ACI 318 Code Requirements between 1963 and 1989 .......... 257
7.3.1.3. Shear Strength by Existing Data .................................................. 260
7.3.1.4. Shear Strength by Regan Method ................................................. 261
7.3.1.5. Batchelor and Kwun Approach ................................................... 263
7.3.1.6. Approximation Analytical Approach .......................................... 263
7.3.1.7. Fracture Mechanics Approach ................................................... 264
7.3.2. Reinforced Concrete Shear Transfer Mechanisms Failure .......... 265
7.3.2.1. Shear Resistance of Un-cracked Concrete Stress ....................... 265
7.3.2.2. Interface Shear Transfer Stress .................................................. 265
7.3.2.3. Shear Transfer in Reinforced Concrete by Lateral Reinforcement ... 267
7.3.2.4. Arch Action Stress .................................................................. 267
7.3.2.5. Dowel Action Stress .................................................................. 267
7.4. SHEAR CRACKING STRESS AT PREPARATORY STAGE .......... 270
7.5. SHEAR CRACKING STRESS AT THE RENEW STAGE ................. 275
7.6. THE STRENGTH AND BEHAVIOUR OF BEAM-COLUMN JOINTS ..... 280
7.6.1. Axial Force and Shear Strength of a Joint .................................. 280
7.6.2. ACI-ASCE Committee 352 (1976) ................................................ 282
7.6.2.1. Axial or Compression Load Design Requirements .................... 285
7.6.2.2. Shear Design Requirements ...................................................... 286
7.6.2.3. Design the of Reinforcement .................................................... 288
7.6.3. ACI-ASCE Committee 352 (1985) .........................................................291
7.6.3.1. Revised Axial or Compression Load Design Requirements ........291
7.6.3.2. Revised Shear Design Requirements .............................................292
7.6.3.3. Revised Design for the Development length of Reinforcement ....293

7.7. PREPARATORY STAGE FOR MODEL SPECIMEN
EVALUATION .................................................................................. 294

7.8. PREPARATORY AND RENEW STAGES FOR SHEAR
FORCES .......................................................................................... 301

7.9. SERVICEABILITY CRITERION FOR DIAGONAL
CRACKING .................................................................................. 308

CHAPTER 8 .......................................................................................... 309

EVALUATION FOR THE FINAL PREDICTIVE STAGE ................. 309

8.1. INTRODUCTION .............................................................................. 309

8.2. EVALUATION FOR THE FINAL STAGE ...................................... 313

8.2.1. Final Stage for Ultimate Moment Computation ....................... 313

8.2.2. Moment-Curvature Relationship at the Final Stage ............... 322

8.2.2.1. Moment-Curvature Relationship at Yield ............................... 322

8.2.2.2. Moment-Curvature Relationship at Ultimate ....................... 329

8.2.3. Shear Forces at the Final Stage .................................................. 336

8.3. ADJUSTMENTS FOR THE SHEAR STRENGTH OF THE
MODEL .......................................................................................... 340

CHAPTER 9 .......................................................................................... 347

SUMMARY AND CONCLUSION .......................................................... 347
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Type of Beam-Column Joints</td>
</tr>
<tr>
<td>1.2</td>
<td>Computational Technology Paradigms</td>
</tr>
<tr>
<td>1.3</td>
<td>Strategies for Implementations of Predictive Stages</td>
</tr>
<tr>
<td>2.1</td>
<td>Main Components of an Expert System</td>
</tr>
<tr>
<td>2.2</td>
<td>Machine Learning Strategies</td>
</tr>
<tr>
<td>2.3</td>
<td>Learning Algorithms</td>
</tr>
<tr>
<td>2.4</td>
<td>Commercial Tools</td>
</tr>
<tr>
<td>3.1</td>
<td>Artificial Intelligence Main Paradigm</td>
</tr>
<tr>
<td>3.2</td>
<td>The Biological Nerve Cell (Neuron)</td>
</tr>
<tr>
<td>3.3</td>
<td>Non-Linear Functions</td>
</tr>
<tr>
<td>3.4</td>
<td>Perceptron Network Model</td>
</tr>
<tr>
<td>3.5</td>
<td>ADALINE Network</td>
</tr>
<tr>
<td>3.6</td>
<td>Hopfield Network</td>
</tr>
<tr>
<td>3.7</td>
<td>Fundamental Processing Element</td>
</tr>
<tr>
<td>3.8</td>
<td>Backpropagation Network</td>
</tr>
<tr>
<td>3.9</td>
<td>Kohonen Self-Organising Network</td>
</tr>
<tr>
<td>3.10</td>
<td>Concept of Neighbourhood</td>
</tr>
<tr>
<td>4.1</td>
<td>Corner Beam-Column Joint Details</td>
</tr>
<tr>
<td>4.2</td>
<td>Neural Network Strategies Implementation</td>
</tr>
<tr>
<td>5.1</td>
<td>Concrete Strength Versus Tensile Concrete Networks</td>
</tr>
</tbody>
</table>
5.2 Concrete Cylinder Strength Versus Modulus of Rupture .......... 130
5.3 Concrete Cylinder Strength Versus Modulus of Elasticity ...... 135
5.4 Variations of Concrete Cylinder Strength with Modulus of Elasticity ................................................................. 136
5.5 Concrete Strain at Maximum Stress by Nirjar [1] ............... 139
5.6 Concrete Strain at Ultimate Stress by Nirjar [1] ................. 140
5.7 Typical Stress-Strain Relationship ................................. 142
5.8 Stress-Strain Relationship according to Hognestad [85] .... 143
5.9 Stress Strain Curves for $f'_c = 20N/mm^2$ ...................... 156
5.10 Stress Strain Curves for $f'_c = 30N/mm^2$ ...................... 156
5.11 Stress Strain Curves for $f'_c = 35N/mm^2$ ...................... 157
5.12 Stress Strain Curves for $f'_c = 40N/mm^2$ ...................... 157
5.13 Stress-Strain Distribution for a Typical Cross-Section of a RC Beam ................................................................. 160
5.14 Stress-Strain Distribution ........................................ 163
5.15 Strain Conditions .................................................. 165
5.16 Trained and Tested Networks for Ultimate Moment at Preparatory Stage ................................................................. 170
5.17 Trained and Tested Networks for Ultimate Moment at Renew Stage ................................................................. 179
5.18 Confined Cross-Section for Flexural Strength .................. 184
6.1 Flexural Member Deformation .................................... 192
6.2 Mathematical Representation of the Radius of Curvature .... 194
6.3 Idealised Load-Deflection and Moment-Curvature Relationships ................................................................. 196
6.4 Elastic Stress Distribution Condition at Yield of the
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5</td>
<td>Cracking Moment and Curvature</td>
<td>201</td>
</tr>
<tr>
<td>6.6</td>
<td>Yield Moment and Curvature</td>
<td>206</td>
</tr>
<tr>
<td>6.7</td>
<td>Curvature and Moment at Yield for the Preparatory Stage</td>
<td>213</td>
</tr>
<tr>
<td>6.8</td>
<td>Ultimate Stress Distribution at Ultimate Stage</td>
<td>214</td>
</tr>
<tr>
<td>6.9</td>
<td>An Idealised Tri-linear Steel Stress-Strain Relationship</td>
<td>215</td>
</tr>
<tr>
<td>6.10</td>
<td>Curvature and Moment at Ultimate for the Preparatory Stage</td>
<td>222</td>
</tr>
<tr>
<td>6.11</td>
<td>Stress-Strain Relationship for Unconfined Concrete in Compression</td>
<td>229</td>
</tr>
<tr>
<td>6.12</td>
<td>Curvature and Moment at Yield for the Renew Stage</td>
<td>237</td>
</tr>
<tr>
<td>6.13</td>
<td>Curvature and Moment at Ultimate for the Renew Stage</td>
<td>240</td>
</tr>
<tr>
<td>7.1</td>
<td>Behaviour of Homogenous Rectangular Beam</td>
<td>253</td>
</tr>
<tr>
<td>7.2</td>
<td>Failure Envelopes</td>
<td>257</td>
</tr>
<tr>
<td>7.3</td>
<td>Reinforced Concrete Shear Transfer Mechanism Failure</td>
<td>269</td>
</tr>
<tr>
<td>7.4</td>
<td>Shear Cracking Stress by Neural Networks at Preparatory Stage</td>
<td>272</td>
</tr>
<tr>
<td>7.5</td>
<td>Shear Cracking Stress by Neural Networks at Renew Stage</td>
<td>277</td>
</tr>
<tr>
<td>7.6</td>
<td>Beam-Column Joints Under Loading For Type I</td>
<td>284</td>
</tr>
<tr>
<td>7.7</td>
<td>Beam-Column Joint Requirements</td>
<td>290</td>
</tr>
<tr>
<td>7.8</td>
<td>Shear Strength Parameters by Neural Networks for the Preparatory Stage</td>
<td>300</td>
</tr>
<tr>
<td>7.9</td>
<td>Possible Combinations for Joint Shear Determination at Preparatory and Renew Stages</td>
<td>302</td>
</tr>
<tr>
<td>7.10</td>
<td>Joint Shear Force Strength at Preparatory Stage</td>
<td>306</td>
</tr>
<tr>
<td>7.11</td>
<td>Joint Shear Force Strength at Renew Stage</td>
<td>307</td>
</tr>
<tr>
<td>8.1</td>
<td>Trained and Tested Networks for Ultimate Moment</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>8.2</td>
<td>Curvature and Moment Networks at Yield for the Final Stage</td>
<td>328</td>
</tr>
<tr>
<td>8.3</td>
<td>Curvature and Moment Networks at Ultimate for the Final Stage</td>
<td>335</td>
</tr>
<tr>
<td>8.4</td>
<td>Trained and Tested Networks for Shear Force at Final Stage</td>
<td>339</td>
</tr>
<tr>
<td>8.5</td>
<td>Adjustment Networks for the Model Specimen</td>
<td>346</td>
</tr>
<tr>
<td>9.1</td>
<td>Author's View Point of Past, Current and Future Research Trends</td>
<td>353</td>
</tr>
<tr>
<td>A.1</td>
<td>Simple Backpropagation Network</td>
<td>378</td>
</tr>
<tr>
<td>C.1</td>
<td>Stress-strain Relationship Networks for $f_c = 20 \text{ N/mm}^2$</td>
<td>389</td>
</tr>
<tr>
<td>C.2</td>
<td>Stress-strain Relationship Networks for $f_c = 30 \text{ N/mm}^2$</td>
<td>390</td>
</tr>
<tr>
<td>C.3</td>
<td>Stress-strain Relationship Networks for $f_c = 35 \text{ N/mm}^2$</td>
<td>391</td>
</tr>
<tr>
<td>C.4</td>
<td>Stress-strain Relationship Networks for $f_c = 40 \text{ N/mm}^2$</td>
<td>392</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Neural Network Market [65]</td>
</tr>
<tr>
<td>3.2</td>
<td>USA Companies Designing Neural Networks [65]</td>
</tr>
<tr>
<td>4.1</td>
<td>Experimental Specimens Adopted by Nirjar [1]</td>
</tr>
<tr>
<td>5.1</td>
<td>Modulus of Rupture</td>
</tr>
<tr>
<td>5.2</td>
<td>Concrete Modulus of Elasticity</td>
</tr>
<tr>
<td>5.3</td>
<td>Parameters Used in Stress-Strain Relationship for Preparatory Stage</td>
</tr>
<tr>
<td>5.4</td>
<td>Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_c = 20 \text{ N/mm}^2$</td>
</tr>
<tr>
<td>5.5</td>
<td>Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_c = 25 \text{ N/mm}^2$</td>
</tr>
<tr>
<td>5.6</td>
<td>Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_c = 30 \text{ N/mm}^2$</td>
</tr>
<tr>
<td>5.7</td>
<td>Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_c = 35 \text{ N/mm}^2$</td>
</tr>
<tr>
<td>5.8</td>
<td>Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_c = 40 \text{ N/mm}^2$</td>
</tr>
<tr>
<td>5.9</td>
<td>Comparison of Ultimate Moments Produced by Nirjar [1] and by Neural Networks for $f'_c = 20 \text{ N/mm}^2$</td>
</tr>
<tr>
<td>5.10</td>
<td>Comparison of Ultimate Moment Produced by Nirjar [1]</td>
</tr>
</tbody>
</table>
and by Neural Networks for $f'_c = 25N / mm^2$.

(Not seen by network). ................................................................. 173

5.11 Comparison of Ultimate Moment Produced by Nirjar [1]

and by Neural Networks for $f'_c = 30N / mm^2$ ......................... 174

5.12 Comparison of Ultimate Moments Produced by Nirjar [1]

and by Neural Networks for $f'_c = 35N / mm^2$.

(Not seen by network). ................................................................. 175

5.13 Comparison of Ultimate Moments Produced by Nirjar [1]

and by Neural Networks for $f'_c = 40N / mm^2$.

(Not seen by network). ................................................................. 176

5.14 Comparison of Ultimate Moment by Nirjar [1]

and by Neural Networks for $f'_c = 45N / mm^2$ ......................... 177

5.15 Comparison of Percent Errors Due to Moment

Computations for Renew Stage................................................. 180

5.16. Comparison of Percent Errors Due to Moment

Computations for Renew Stage................................................. 181

6.1 Comparisons of Cracking Moments..................................... 200

6.2 Comparisons of Moments and Curvatures at Yield by Nirjar [1]

and by Neural Networks for the Preparatory Stage.................... 209

6.3 Comparisons of Moments and Curvatures at Ultimate by Nirjar [1]

and by Neural Networks for the Preparatory Stage.................... 223

6.4 Comparison of Moments and Curvatures at Yield for the

Renew Stage in Terms of Neural Networks................................. 232

6.5 Comparison of Moments and Curvatures at Ultimate by Nirjar [1]

and by Neural Networks for the Renew Stage............................ 241

7.1 Network Parameters for Shear Cracking Stress
at Preparatory Stage................................................................. 271

7.2. Comparisons of Concrete Shear Cracking by Several Investigators and Neural Networks at Preparatory Stage. .................. 273

7.3 Network Parameters for Shear Cracking Stress at Renew Stage. ........................................................................... 276

7.4 Comparisons of Concrete Shear Cracking by Several Investigators and Neural Networks at Renew Stage. .................. 278

7.5 Preparatory Stage Comparisons between the Shear Strength of the Model Specimens by Nirjar [1] and Neural Networks........ 295

7.6 Shear Force Strength for the Model Specimens at Preparatory and Renew Stages by Neural Networks. ......................... 304

8.1 Variations of Parameters Used in the Computational Process... 312

8.2 Comparison of Ultimate Moment Produced by Preparatory and Final Stages. $f_c' = 20 \, N/mm^2$. .............................. 315

8.3 Comparison of Ultimate Moment Produced by Preparatory and Final Stages. $f_c' = 25 \, N/mm^2$. .............................. 316

8.4 Comparison of Ultimate Moment Produced by Preparatory and Final Stage. $f_c' = 30 \, N/mm^2$ ............................... 317

8.5 Comparison of Ultimate Moment Produced by Preparatory and Final Stages. $f_c' = 35 \, N/mm^2$. ............................... 318

8.6 Comparison of Ultimate Moment Produced by Preparatory and Final Stages. $f_c' = 40 \, N/mm^2$ ............................... 319

8.7 Comparison of Ultimate Moment Produced by Preparatory and Final Stages. $f_c' = 45 \, N/mm^2$. ............................... 320

8.8 Comparison of Moments and Curvatures at Yield by the Preparatory and the Final Stages............................................. 324
8.9 Comparison of Moments and Curvatures at Ultimate by Preparatory and the Final Stages .......................................................... 331

8.10 Shear Force Strength for the Model Specimens at Final Stage by Neural Networks ................................................................. 337

8.11 Adjustments for Shear Strength of the Model Specimens at Preparatory Stage to Final Stage by Neural Networks .................. 342
Special Notations for Neural Networks Output.

\[ x_{out,Stage}^{PE,Con} \]
\[ y_{out,Stage}^{PE,Con} \]

where:

- \( x \) = predicted value within the neural networks.
- \( y \) = scaled value by neural networks.
- \( PE \) = number of processing element in the network.
- \( Con \) = condition.
- \( out \) = neural network output.
- \( stage \) = stage where the network produced output.

Example:

\[ x_{78,y}^{out,r} \] = predicted curvature within the neural network for PE output number 78 at yield condition for the renew stage.

\[ y_{78,y}^{out,r} \] = scaled curvatures for PE output number 78 at yield condition for the renew stage.

- \( a_\theta \) = ADALINE bias of value equal to 1.
- \( a_b \) = arm lever of the beam.
- \( a_c \) = inclined stress cracking position.
\(a_i\) = ADALINE input processing element of \(i\).

\(a_s\) = shear span from first concentrated load to beam reaction.

\(a_z\) = distance between points of zero and maximum moments.

\(ad\) = distance from the compression face of member to the centroid of compression steel.

\(b\) = width of the cross-section.

\(b_c\) = width of the column.

\(b_E\) = outside to outside width of column for confined cross-section.

\(b_{ij}\) = outside to outside of ties or column bars if no ties are used for unconfined cross-section.

\(b_i\) = input signals for the backpropagation network.

\(b_j\) = effective joint width and depends on the lateral widths of the beams framing into the column as well as the lateral width of the column. In this study \(b_j = 0.5(b + b_c)\).

\(d\) = effective depth of the cross-section.

\(d_b\) = bar diameter.

\(d_{ce}\) = column effective depth.

\(d_{ja}\) = ADALINE desired output.

\(d_{jb}\) = desired signals of the backpropagation network.

\(d_{jp}\) = perceptron desired output.

\(d_{out}\) = desired output.

\(d_{qk}\) = backpropagation desired output.

\(e_{qk}\) = backpropagation error signal for one processing element in the output layer.

\(f\) = concrete stress at a distance 'y' above the neutral axis as expressed by the equation.

\(f_0\) = concrete maximum stress.

\(f'_c\) = cylinder compressive strength of concrete.

\(f''_c\) = compressive strength of bound concrete analogous to \(f'_c\).
$f_h = \text{stress developed by a standard hook.}$  

$f_g = \text{general stress at a point.}$  

$f_t = \text{tensile strength of concrete expressed as the modulus of rupture.}$  

$f_T = \text{flexural stress due to bending.}$  

$f_u = \text{ultimate steel stress.}$  

$f_y = \text{yield stress of the main reinforcement.}$  

$f_y' = \text{yield stress of compression reinforcement.}$  

$f_{av} = \text{average concrete compressive stress for unbound concrete.}$  

$f_{av}' = \text{average concrete compressive stress for bound concrete.}$  

$f_{cw} = \text{characteristic concrete strength.}$  

$f_y' = \text{yield stress of the lateral reinforcement.}$  

$f_{sbal} = \text{yield stress of the main reinforcement for balanced condition.}$  

$f_c^c = \text{average confined compressive stress of the concrete.}$  

$f_c^{\max} = \text{maximum principal compressive stress.}$  

$f_t^{\max} = \text{maximum principal tensile stress.}$  

$h_c = \text{depth of column in the direction of load being considered.}$  

$h_i = \text{Hopfield input processing element.}$  

$h_j = \text{Hopfield output processing element.}$  

$h_k = \text{Kohonen neighbourhood horizontal distance.}$  

$h_{PE} = \text{upper bound for the number of processing elements in the hidden layer.}$  

$i = \text{refer to input layer in backpropagation.}$  

$i_{PE} = \text{number of processing elements in the input data.}$  

$j = \text{refer to hidden layer in backpropagation.}$  

$k = \text{refer to the output layer in backpropagation}$  

$k_r = \text{factor of safety, between 0.20 to 0.25.}$
\( k_{\text{win}} \) = Kohonen winning processing element.

\( n \) = modular ratio equals to \( \frac{E_s}{E_c} \).

\( n_{oPE} \) = number of patterns in training set.

\( n_q \) = number of processing elements in the output layer.

\( o \) = backpropagation processing element in input layer.

\( o_{PE} \) = number of processing elements in the output layer.

\( p_i \) = input signals in the perceptron network, 1 or 0.

\( p^* \) = binding ratio parameter to allow for the effect of confinement.

\( q \) = backpropagation processing element in the output layer.

\( q^* \) = parameter for the effectiveness of the lateral reinforcement.

\( r_b \) = radius of bar diameter.

\( r_l \) = lateral reinforcement ratio of steel.

\( s_0 \) = longitudinal spacing of the links at which the lateral reinforcement was not effective in confining the concrete.

\( s_a \) = ADALINE summation.

\( s_h \) = spacing of tie reinforcement measured along column bars.

\( s_{jh} \) = summation of Hopfield processing element.

\( s_{jp} \) = perceptron summation.

\( s_r \) = spacing of the lateral reinforcement.

\( t_c^k \) = Kohonen iteration of current training.

\( v \) = shear stress.

\( v_b \) = volume of bound concrete.

\( v_c \) = concrete shear stress.

\( v_f \) = average shear stress at shear interface for the crack area.

\( v_k \) = Kohonen neighbourhood vertical distance.

\( v_s \) = volume of stirrups.
\( \nu_u \) = ultimate shear stress.

\( w_c \) = concrete density.

\( w^{a}_{ji} \) = ADALINE weight adjusted from input \( i \).

\( w^{b}_{ji} \) = backpropagation weight adjustment between the input and output signals.

\( w^h_{ji} \) = Hopfield weight adjustment.

\( w^{new,p}_{ji} \) = perceptron new weight after adjustment.

\( w^{old,p}_{ji} \) = perceptron old weight before adjustment.

\( w^{new,k}_{ij} \) = Kohonen new weight adjustment.

\( w^{old,k}_{ij} \) = Kohonen old weight adjustment.

\( x_d \) = distance of the neutral axis from the compression face.

\( x_{ud} \) = distance of the neutral axis from compression face at ultimate.

\( y \) = distance from the element to the neutral axis.

\( y_b \) = distance from the centroid of \( A_{cc} \) to the neutral axis.

\( y_{ja} \) = ADALINE network output.

\( y_j \) = output signals of the backpropagation network.

\( y_{pp} \) = output signals of the perceptron network.

\( y_{out} \) = network output.

\( y_{qk} \) = backpropagation actual output for one processing element in particular layer.

\( y_d \) = depth to the resultant concrete compressive force from the neutral axis.

\( z \) = factor which depends on the concrete cylinder strength, \( f_c' \).

\( A_b \) = area of bound concrete under compression \(( A_b = b'x' ) \).

\( A_c \) = area of concrete under compression \(( A_c = bx ) \).

\( A_g \) = gross area of cross-section.

\( A_s \) = area of tension steel reinforcement.
\( A_v \) = total area of lateral reinforcement.

\( A_s \) = area of compression steel reinforcement.

\( A_{cc} \) = cross-area of the section at the plane that passes through the centroid of element 'I'.

\( A_{ch} \) = area of rectangular core measured to the outside of the hoop or tie.

\( A_{ov} \) = area of one leg of a link.

\( A_{sc} \) = area of compression reinforcement in the column.

\( A_{sh} \) = total area of all hoop and extra cross-tie leg crossing mid-depth of the section, within the spacing \( s_h \), in the direction considered.

\( B \) = the greater area of \((0.7x')\) or \((0.7b')\).

\( C_1, C_2 \) = material constants for biaxial stresses state.

\( C_r \) = cohesive constant \( \approx 0.44(f_{cu})^{0.67} \).

\( C_u \) = ultimate compressive force.

\( D \) = diameter of the compression steel.

\( D_i^h \) = Euclidean distance.

\( E \) = Modulus of Elasticity of the cross-section.

\( E_c \) = Modulus of Elasticity of concrete.

\( E_h \) = Hopfield Energy.

\( E_s \) = Modulus of Elasticity of steel.

\( E_T \) = Tangent Modulus of Elasticity of concrete.

\( H \) = total depth of the cross-section.

\( H_c \) = total height of the column.

\( I \) = moment of inertia of the gross cross section equal to \( b \frac{H^3}{12} \), where its centroidal axis is in the plane of bending.

\( K \) = vector of Kohonen input data.

\( L_d \) = development length for a straight bar.
$L_h$  = width, measured to outside of tie reinforcement in the
direction perpendicular to that of shear force being considered.

$L_{dh}$  = development length for a hook bar measured from the critical
section to the outside end of the hook measured.

$M$  = applied moment at the cross-section.

$M_{cr}$  = cracking moment at the cross-section.

$M_y$  = yield moment at the cross-section.

$M_u$  = ultimate moment at the cross-section.

$N_{PE}$  = number of rows in training data.

$N_k$  = Kohonen neighbourhood distance.

$N_{k0}$  = Kohonen initial neighbourhood distance.

$P_a$  = applied axial load.

$P_b$  = axial load capacity at simultaneous assumed ultimate strain of
concrete and yielding of tension steel.

$P_{cr}$  = the load at cracking condition.

$P_u$  = column ultimate strength.

$R$  = range between 5 and 10.

$R_c$  = radius of curvature.

$RMS$  = Root-Mean-Square.

$S$  = section modulus equal to $b\frac{H^2}{6}$.

$S^k_N$  = Kohonen neighbourhood size.

$T^{k}_{train}$  = Kohonen total number of training carried out.

$V_c$  = shear force of concrete.

$V_f$  = shear force along the shear interface.

$V_s$  = shear force provided by stirrups.

$V_n$  = nominal shear strength of a joint.
\( V_u \) = ultimate shear force at the cross-section.

\( W^k \) = Kohonen weights vector.

\( W^k_{\text{win}} \) = Kohonen winner weights vector.

\( \alpha \) = ratio of the principle stress in the orthogonal direction to the principle stress in the direction considered.

\( \alpha_p \) = perceptron learning rate, usually between 0 and 1.

\( \alpha_k \) = Kohonen learning constant.

\( \alpha^k_{\text{inc}} \) = Kohonen an acceptable decrease rate of constant.

\( \alpha^k_{\text{ini}} \) = Kohonen learning constant that decreased to reach zero.

\( \beta \) = 1.4 for joint Type I.

\( \beta_1 \) = constant, varying as \( f_c' \), and equal to 0.85 for \( f_c' = 30 \text{ N/mm}^2 \).

\( \varepsilon \) = concrete strain in the direction of concrete stress, \( f \).

\( \varepsilon_0 \) = concrete strain in the concrete at maximum stress.

\( \varepsilon_c \) = compressive strain in the concrete.

\( \varepsilon_g \) = general strain at a point.

\( \varepsilon_h \) = strain hardening.

\( \varepsilon_{20c} \) = concrete compressive strain when the concrete stress has reduced to 0.2\( f_c' \).

\( \varepsilon_{\text{cm}} \) = concrete compressive strain in the extreme fibres.

\( \varepsilon_u \) = ultimate concrete strain for the unconfined section.

\( \varepsilon_u^c \) = ultimate concrete strain for the confined section.

\( \phi \) = curvature at cross-section with appropriate subscript (rotation per unit length of member).

\( \phi_f \) = the equivalent coefficient of friction which it is determined by tests. It is analogous to the cohesion of soils.

\( \phi_s \) = factor of safety equal to 0.85.

\( \gamma \) = factor to account for cross-section confinement.

\( \tau \) = general shear stress.
\( \theta_{\text{max}} \) = the maximum angle for the occurrence of maximum and minimum principal stresses measured in radians.

\( \eta \) = learning rate or learning coefficient. Usually a constant value between 0 and 1 is used to modify the learning process.

\( \mu \) = Poisson's ratio.

\( \mu_f \) = coefficient of friction, between 0.8 and 1.0.

\( \theta_{\text{max}} \) = angle between the diagonal tension and the horizontal.

\( \rho \) = tensile reinforcement ratio of steel.

\( \rho' \) = reinforcement ratio of compression steel.

\( \rho_b \) = reinforcement ratio of tension steel at balanced condition.

\( \rho_c \) = column longitudinal reinforcement ratio.

\( \rho_{\text{min}} \) = minimum reinforcement ratio.

\( \rho_v \) = lateral reinforcement ratio equals \((A_v/bd)\).

\( \sigma \) = general normal stress.

\( \sum p_i \) = summation over all patterns in the training set.

\( \sum_{oPE} \) = summation over all output processing elements.

\( \Delta \) = deflection of cross-section with appropriate subscript.

\( \Delta E_j \) = difference in Hopfield energy.

\( \Delta w_{ij}^k \) = Kohonen difference in updating weights.

\( \Delta w_{ji}^q \) = Widrow-Hoff delta rule.

\( \Delta w_{qp,k}^{s} \) = backpropagation adjusted weight between the \( q^{th} \), processing element in the particular layer and \( p^{th} \), processing element in the particular layer.

\( \psi \) = factor to account for development of reinforcement.
1.1. GENERAL INTRODUCTION

Beam-column joints are fundamental elements in Reinforced Concrete buildings due to their ability to transform the forces from the floor systems to the columns. The nature of stresses in the joint is very complex resulting from the effect of several forces acting in multi-directions including axial loads, bending moments, shear and torsion. To understand the complex state of the stress distributions, more experimental work is required to accompany the analytical methods, verified by numerical analysis and computational technology such as neural networks. The analysis and behaviour of beam-column intersections have been the subject of discussion for several years and only limited research data is available on the effects of axial compression and biaxial loading of a joint. This subject is considered to be one of the more difficult problems encountered in reinforced concrete theory, since only a limited number of tests and analytical results have been published. Current Codes of Practice for the beam-column joints rely on empirical interpolation of experimental results and evaluation by analytical methods.

Publications produced over the years on the design and analysis of beam-column joints in reinforced concrete structures deal mainly with the anchorage requirements of the beam-column reinforcing bars and the acceptable level of shear within the joint. The anchorage of reinforcing bars is important for design consideration to provide continuity and safety in reinforced concrete structures. Conditions should be
imposed to satisfy the design and detailing criteria which avoid splicing of reinforcement in the joint and check the confinement of the concrete. With the introduction of new materials, economic sections, and larger diameter high tensile steel reinforcement, it has become important to look beyond the particular requirements of anchorage. Attention must now be paid to design requirements such as areas of steel, high strength material, and the performance of such joints in construction to avoid potential problems in the future.

Research on beam-column joints for the past few decades has concentrated on experimental investigations and the development of analytical methods, and empirical formulae. This research has generally involved experiments on model joints which simulate typical structural members in order to understand their behaviour and provide a means of implementing a design process.

The aim of author's study was to view previous experimental studies of beam-column joint parameters from a different angle to establish a concept and methodology that would provide rapid and economical benefits for future experimental research. The study attempts to re-construct the experimental work previously performed by Nirjar [1] to evaluate various parameters and to provide a method which would assist future experimental work. The computer evolution prompted the author to search for a more efficient way of approaching experimental testing. This study is an attempt to guide experimental testing by numerical manipulation through the conceptual implementation of artificial intelligence in its symbolic form using the powerful predictions provided by neural networks. The method used have a solid mathematical foundation which depends on the availability of large sets of data for which humans cannot recognise patterns for their internal representation. Neural networks have the ability to recognise this internal representation of data by separating the data into multi-dimensional space and maintaining tracking of that space through the process of variable mapping by non-
linear equations. The concept and methodology proposed for this study can be transformed to other engineering fields, where large amounts of data are available, by using numerical methods and finite element programming techniques.

The computational methodology presented in this study for the beam-column joint was formulated with the emphasis on adopting three stages to establish a design procedure.

1.2. BEAM-COLUMN JOINT

The simulation of beam-column joints by experimental models is well documented, various models have been proposed by several investigations in which each differs slightly from the other. The main objective in each case was to simulate the beam-column joint behaviour by providing realistic support conditions. The test set up is built according to the proposed experimental procedure and parameters introduced such as section geometry, the quality of the concrete and steel, and the type of detailing.

Three types of joint exist in reinforced concrete structures namely; interior, exterior and corner joints, as shown in Figure 1.1. Figure 1.1 (d) shows the beam-column corner joints investigated in this study. ACI-ASCE Committee 352 [2,3] recommended for beam-column joints in monolithic reinforced concrete structures that the joint should be designed for axial force, bending moment, reinforcement development, shear, torsion, and secondary effects (shrinkage and creep, etc.). The main assumptions adopted by the ACI-ACSE Committee 352 [2] for the design of a joint were concerned about requirements of beam anchorage, column bars, and the fact that the concrete and lateral reinforcement in the joint act together to resist the level of shear stress in the joint. A full in-depth analysis and design procedure based on the recommendations of ACI-ASCE 352 Committee [2] was carried out by Wang and Salmon [4]. A second report has issued by the ACI-ASCE Committee [3] to take
Figure 1.1. Type of Beam-Column Joints.
account of new test results and a better understanding of the joint strength and behaviour. Nilson and Winter [5] then provided a new detailed design procedure for joint reinforcement on the basis of the second ACI-ASCE Committee [3] report. The primary investigation in the author's study was confined to the recommendations of both ACI-ASCE [2,3] Committees for Type I joints resisting gravity and normal wind load forces.

The majority of the investigations undertaken from 1980 were devoted to studying the behaviour of beam-column joints in seismic zones. Little attention was given to the characteristics of the use of a reinforced concrete beam-column joints. ACI-ASCE Committee 352 [2,3] recommended Type II joint for use in seismic zone but this type was not considered in the author's study.

1.2.1. Literature Review

In general, mass investigations into beam-column joints have been concerned with detailing within the joints, consideration of the column lateral reinforcement and continuation of the main reinforcement in the beams through the joints.

A study of detailing was undertaken by Taylor and Clarke [6] who investigated the influence of detailing on the efficiency of beam-column joints and slab-column joints. The study was based on four experimental investigations which included pile-caps, small nibs, concrete beam-column joints and column-slab connections. The general conclusion concentrated on the occurrence of cracks within the joint as a serviceability criterion. Details of reinforcement in beam-column corner joints using light weight concrete were provided by Mayfield, Kong, Bennison and Davies [7] for 48 test specimens. The study presented results for 12 types of corner-joints with values for ultimate strength, cracks as well as stiffness. The influence of detailing on the behaviour of a closed joint was less than an open corner due the effect of the applied moment.
A comprehensive experimental and analytical study into the strength and behaviour of the beam-column corner was undertaken by Nirjar [1]. The test programme consisted of 34 specimens to evaluate the ultimate flexural strength of the beam members, load-deformation and moment-curvature relationships as well as the effect of shear considerations with bond. Analytical and empirical relationships were established at various stages based on the experimental results. The work of Nirjar [1] was extended in this study to evaluate an experimental procedure by numerical analysis with the integration of neural networks as a predication algorithm.

A rational method for designing a beam-column joint subjected to bending moments was undertaken by Nilsson and Losberg [8] to satisfy the essential requirements of strength, cracking, ductility and simplification of the construction methods. The investigation was carried out on 78 beam-column joints subjected to single direction monotonic loading. Three types of reinforcement were proposed, at right-angles, obtuse and acute angles, depending on the moment. Ductility considerations were investigated by Roy and Sozen [9] by testing 60 axially loaded prisms, of dimensions of 5" by 5" and by 25". Load deformation characteristics were determined for each specimen. The results indicated that the load capacity of concrete under axial load is not increased by the presence of lateral reinforcement but that improvement in the ductility of the concrete is achieved. The rotational capacity of the hinge regions of reinforced concrete beams was investigated by Mattock [10], using 37 beams with variable parameters. The rotational capacity of the hinging regions was obtained by evaluating the moment-curvature relationship corresponding to the gradient condition. This relationship was established on the basis of strain compatibility and equilibrium of the forces. An analytical study supported by experimental findings for the ultimate strength and deformation of plastic hinges was completed by Chan [11]. This study concluded that the lateral reinforcement has an effect on the stress-strain relationship and encourages the development of larger plastic hinges in under-reinforced sections. A finite element analysis was carried out
by Suidan and Schnobrich [12] which predicted the behaviour of reinforced concrete members and incorporated the elastic-plastic behaviour of concrete and steel as well as the cracking of concrete in the tension zone. The analysis produced results which were in agreement with observed experimental results.

A general review of the literature for test programmes of reinforced concrete beam-column connections up to 1981 was summarised by Meinheit and Jirsa [13]. The summary indicated concern in the construction details to maintain ductility and check anchorage requirements in an exterior connection due to the unbalanced moment produced by the shear. The study also concentrated on the influence of the shear capacity of the beam-column joint and concluded that cross-section was a major factor for joint strength.

A numerical model for the analysis of the anchorage of the beam reinforcement in exterior column-beam connections was reported by Ueda, Lin and Hawkins [14] based on predictions of 22 test models. The study provided the development of a computer program that predicted the axially loaded end displacement characteristics of the exterior column-beam connection for adequately anchored reinforcement in the beam. The results were found in to be in agreement with those obtained from experimental models.

The effect of detailing on the behaviour of reinforced concrete beam-column joints was investigated experimentally by Scott [15] using results from 15 reinforced concrete external beam-column joints. The detailing used was classified into three categories of bar viz: bent up, bent down or U-shaped. Behaviour of the connection mechanism was observed by placing gauges on the reinforcement which provided strain distributions along the beam and column bars.

In summary, the work carried out for to assess the behaviour and strength of beam-column joints has been mainly experimental using analytical verifications
based on fundamental theoretical formulation. These investigations have contributed to an understanding of the fundamental mechanism of joint behaviour and provided guidance for the issue of Code recommendations. Little consideration has been given to the use of recent technological advances in both hardware and software for the analysis and behaviour of beam-column connections. This prompted the author to search for alternative strategies by exploring the benefits of numerical analysis and neural network technology to study the behaviour and assess the strength of beam-column joints. The potential use of this technique is demonstrated throughout the thesis.

1.3. COMPUTATIONAL TECHNOLOGY PARADIGMS

Computational technology has influenced the direction of progress in many disciplines, and has promoted competition between research institutions in their search for optimum solutions. The evolution of sophisticated computer has provided the forum for Civil Engineering to initiate ideas and test them instantly, also check results throughout the driving discovery process. The integration of computational technology in the academic and professional fields should be implemented without the need to ask *why* but always to ask *how*.

The introduction of the *Alpha*® Reduced Instruction Set Computers (RISC) processor by Digital Equipment Corporation (DEC) and the *Pentium*® processor by Intel® has shifted dependence on the main frame to personal computers. The challenge of the *Power PC*® with a RISC processor by main rivals of Intel®, mainly with the cooperation of IBM®, Apple® and Motorola®, has shifted the balance in the favour of microprocessor industry. The introduction of the Operating System/2 (OS/2) by IBM® has provided multi-tasking, as well as multi-user interfaces, which have changed the way the Personal Computer (PC) performs. The Microsoft New Technology (NT®) operating system has introduced significant changes in the way we approach personal computers. The NT® with its two platforms, one for individual
use and the other for Server/Client, has compressed twenty years of UNIX®
manipulation of the Graphical User Interface (GUI) environment in to less than five
years and provided individuals with an advanced operating system which was
previously only available in the academic research arena. Microsoft has
revolutionised the operating system by introducing an advanced operating system. It
is now the civil engineer's responsibility to benefit from this technological advance.
The personal computer has progressed from a primitive tool to a highly technological
instrument which can manipulate complex mathematical equations and provide
control by direct manipulation of visual representations of user data.

In Structural Engineering profitable and competitive structures have been
designed during the past decade. To keep up with current in advances research,
computational technologies are being used including sequential programming
(programming languages, mathematical programming tools), symbolic logic base,
and parallel distributed processing such as neural networks. Figure 1.2 presents the
computational technology paradigms that represent the algorithm methods that are
included in this study. The intention of this research is to employ the best emerging
technological procedures of computational paradigms since they will dominate
research in the coming years. These procedures involve the following:

1. Symbolic manipulation based on logic.

2. Sequential programming instruction based on numerical base.

3. Parallel distributed processing based on classification and association base.

1.3.1. Symbolic Manipulation

Symbolic logic base is founded on logic to capture human intelligence and code
it into the machine. The heuristic which is rule of thumb is the basis for artificial
intelligence to simulate human thinking. The expert knowledge was encoded into the
computer in a form of *IF ... THEN* rules. PROLOG and LISP are the main programming languages used extensively by the AI community in their programming.

1.3.2. Sequential Programming Instruction

Traditional computational technology provides a flexible means of achieving reasonable and appropriate results. This traditional technique usually includes numerical computing which, through simplifying assumptions, introduces an iteration process which simplifies the complexity of implicit functions. The basis for computational technology was founded on the programming instruction sequence which was introduced by von Neumann. The algorithms use sequential programming such as BASIC, FORTRAN, C, PASCAL, COBOL and other languages.

1.3.3. Parallel Distributed Processing

Parallel distributed processing researchers have dreamed of imitating the human brain's function of making decisions and drawing conclusions. Their main goal is to develop computer programs that emulate human intelligence. The neural network is primarily a system that can simulate some type of human nervous system and has the ability to learn and adapt according to the information presented. It has the capability to represent numerous data in *n-dimensional* space and track each individual characteristic in that separate space. Its advantage is its potential to manipulate data through its internal representation which is primarily a mathematically complex non-linear function. The disadvantage is the difficulty in presenting these complex relationships in the form of graphical presentation.

The combination of the symbolic manipulation base, parallel distributed processing base and numerical manipulation base shown in Figure 1.2, provides a hybrid knowledge-numeric-neural base which is ultimately the ideal knowledge-based system for providing adequate solutions that neither system can alone solve. This is
Figure 1.2. Computational Technology Paradigms.
achieved by combining the best features of the three approaches but is beyond the scope of this study.

This research attempts to explore these technological advances in the field of structural engineering by concentrating on the analysis of beam-column joints and providing procedures that are similar to that has been adopted experimentally. An attempt is made to demonstrate the potential of all the approaches mentioned previously by analysing the structural components and obtaining reliable and efficient results by manipulation of the computational technology. A symbolic manipulation logic approach was used in this research mainly to apply the principle of learning from experience and examples. The procedure involves reviewing information for the beam-column joint and evaluating the recent research and new Code of Practice. The numerical base of a sequential algorithm is implemented by using Mathcad 4 [16] software and FORTRAN programming where it is necessary. Parallel distributed processing is implemented by incorporating neural network technology to learn from a set of examples that has both input and output data.

1.4. SCOPE OF THE RESEARCH

Experimental investigations on small and full scale models of reinforced concrete structures provide valuable information on the behaviour of these structures. The experimental models are the basis for introducing future improvements in the construction and design procedures. New findings from experimental research also provide a fundamental understanding of the structural behaviour and material properties that are essential for changing or updating the Code of Practice. Experimental investigations require the setting-up of specimen models and the placement of strain gauges both inside and on the surface with constant recording of measurement by electrical gauges. The investigations specify fixed dimensional structural shapes and select materials with different properties. Reinforced Concrete experimental investigations require the selection of concrete components composed
of sand, cement/water ratio, aggregate size and types of sand. Various mixes are adopted to produce a specific concrete strength which are measured by cylinder tests at a specified age. The experimental procedure usually includes loading conditions, the placement of strain gauges in the concrete, erection of reinforcement and measurement of deflections. The reinforcement is selected using different steel strengths and diameter sizes with specified yield properties. The number and placement of the reinforcement plays a major part in the experimental investigation along with the type and degree of loading. The experimental process can last for several months with testing and re-testing and collection of the data. Problems arise due to the failure of instrumentations, misalignment of loading, misplacements of gauges in the models and the laboratory limitations. The test specimen and programmes are expensive and require excessive time to monitor, check and collect data. The cost of fabrication and instrumentation in recent years has grown exponentially with consequent economic implications. Therefore an alternative method is required to overcome these disadvantages without losing the value of the experimental investigation.

To overcome these difficulties and simplify the experimental procedures, alternative methods can be adopted to minimise the time required to perform tests, to reduce the number of specimens tested and to increase the specimens analytically. This can be achieved by implementing technological data manipulation to provide instant, fast and reliable results which ultimately will be cost-effective. An attempt has therefore been made to bridge the gap between the reliability of experimental findings and the efficiency of computation of algorithms by implementing neural networks as an aid to experimental testing by utilising the numerical results.

The main objective of this study is to develop concept and methodology, rather than to build a full expert system, by applying neural network techniques in the form
of predicting parameters that are usually obtained by experimental work. In order to achieve this objective the study was divided into three main stages:

1. **Preparatory Stage**

2. **Renew Stage**

3. **Final Stage**

Figure 1.3 shows the strategies adopted in this study to implement neural network in the form of prediction. The preparatory stage follows the procedure adopted by Nirjar [1]. The renew stage updates the equations and information based on recent research, literature review findings and updates in Codes of Practice. The renew stage consists of integrations of the experimental and analytical data for the preparatory stage together with findings from more advanced study. The renew stage is an advance of the preparatory stage which demonstrates adaptations for future experimental models. The final stage combines the knowledge obtained from the preparatory stage with adaptations for new conditions set by the Code of Practice and research findings. The data used in this stage is independent of the experiment or analytical results and is based only on numerical analysis with the integration of neural networks. The objective of this study is to demonstrate the relevant application of neural networks to predict patterns of information for a specific task.

1.5. **METHODOLOGY AND OUTLINE OF RESEARCH**

The experimental models evaluated were limited to reinforced concrete beam-column corner joints under static loading only. The reinforcement and structural properties were included in the analysis but the effects of shrinkage, creep and temperatures are excluded from the analysis. Preliminary work included numerical evaluation of 34 beam-column corner joint specimens investigated by Nirjar [1]. The computation analysis investigated the varying influence of axial load, longitudinal column reinforcement, tension and compression reinforcement in the beams,
transverse reinforcement in the beams, concrete strength and transverse reinforcement in the joint. The procedure adopted in this research is based entirely on numerical methods, analytical evaluations and experiment results with the use of symbolic logic and neural network technology. The method used involved the prediction of various parameters by neural networks through the knowledge of previous experimental findings, theoretical formulation, analytical evaluation and numerical analysis according to the Code of Practice. The procedure is cost-effective, reliable and will provide alternative solutions for future experimental studies.

Although artificial intelligence and parallel distributed processing are still largely at the experimental stage, it was the aim of this research to look beyond conventional methods for conducting experimental research and present an alternative method based on the improvement of software and hardware in recent years. The numerical methods were linked with these new technologies to enhance performance and provide reliable results. A summary of the three stages proposed are described below. A detailed deployment of the preparatory and renew stages are given in chapter 4, 5, 6 and 7 and of the final stage in chapter 8.

1. **Preparatory stage:** this stage involved collecting information from all the work investigated by Nirjar [1]. It consists of an extensive literature review, equation validations, and a comparison of results obtained by Nirjar [1]. The study employed numerical computation by programming appropriate mathematical equations in FORTRAN and using Mathcad® 4 where appropriate to obtain feedback and instant results for the numerical procedure. Intensive numerical computation was carried out at this stage to provide information on particular cross-sections and parameters required for computations concerning the beam-column joint behaviour. Relevant data comprising the stress-strain relationship, the behaviour of the flexural members for confined and unconfined concrete, the
moment curvature relationship, the shear strength and behaviour and the biaxial stress relationship of reinforced concrete were collected where appropriate.

2. **Renew stage**: this stage utilises the idea of symbolic logic that is based entirely on learning from experience. It makes use of new experimental data, recent research conclusions and new code of practice recommendations to enhance the data. A neural network is then carried out to predict equations that are based on both the collective data of the preparatory stage and updated information. At this stage, comparisons between experimental results and those predicted by neural network were made to assess the percentage errors.

3. **Final stage**: this stage incorporates new evaluations based on the recommendations of the Code of Practice and integrates relevant research works. The main function of the final stage is the ability to perform additional experimental tests using new data that was not obtained previously nor evaluated analytically. It relies on the combination of numerical analysis and neural networks and can also be used as a preliminary investigation.

The study is therefore presented in the following nine chapters:

1. Chapter 1 is a general introduction with a historical background of the study of beam-column joints. Three essential computational paradigms are explained in detail and in particular their role in structural engineering.

2. Chapter 2 is devoted to a review of a general artificial intelligence learning technique that is based on the symbolic logic base which includes expert systems and machine learning in particular. Expert systems have played a major part in the advancement of research in engineering science. Its application is particularly relevant to the medical field or to problems where an excessive number of manipulations of symbolical data is required. Machine learning has overcome difficulties in the development of expert systems. The techniques
adopted in machine learning are essential to understand the way the machine is programmed. Methods of learning are presented, and in particular learning from examples and experience is described in detail.

3. Chapter 3 explores the parallel distributed processing approach in detail. The main emphasis is on the neural network architecture and algorithms are presented together with mathematical formulations for several architectures. The biological neural networks are described to provide a foundation for the artificial neural networks. A basic mathematical formulation for the backpropagation algorithm was established for this study. Learning algorithm techniques are explained with particular emphasis on the supervised learning method.

4. Chapter 4 composes the digital and analogical representation of information relevant to the problem investigated in this thesis. The role of FORTRAN and the mathematical programming tool are explained and their implementation in the thesis is highlighted. A general review of the NeuralWorks® [17] Professional tool is completed and aspects for implementation in the author's study are stressed. Neural Network strategies for implementation of the tool are also indicated. A review of the previous experimental model is made with emphasis on the behaviour of the beam-column joint investigated by Nirjar [1] using established analytical and empirical relationships. Three stages are considered to investigate the parameters, equations and relationships adopted by the neural networks.

5. Chapter 5: factors affecting the flexural strength of reinforced concrete members are considered and the ultimate moment of the cross-section evaluated using several methods. The process of simulating the experimental work is established and the formulation of training and test data files to implement neural networks for the analysis of the ultimate flexural strength of reinforced concrete members
are found. Several parameters including the concrete tensile strength, modulus of elasticity, the stress strain relationship are considered and an ultimate moment calculation predicated by the neural networks is proposed. Mathematical equations are established at each stage and comparisons with experimental results are indicated. The advancement of research based on publications and Codes of Practice is illustrated by converting the theoretical and experimental results to a form accepted by neural networks. New mathematical equations proposed by neural networks are established.

6. The idealised load-deflection and moment-curvature relationships for a flexure member are examined in chapter 6 and emphasis is placed on the three load stages. Three methods of calculating the moment-curvature relationship are discussed and the role of neural networks is examined by providing comparisons between results.

7. Chapter 7 deals with the shear strength and behaviour of the joints as evaluated by Nirjar [1] and proposed methods and recommendations of the ACI-ASCE Committee 352 [2,3] are presented. The stresses and forces due to shear are computed by neural networks and values of percentage errors are also provided.

8. Chapter 8 presents the final stage following an investigation of the preparatory stage and renew stage in chapters 5, 6 and 7. The final stage is a result of examining and learning from the previous two stages.

9. Chapter 9 includes a general summary of the research work already carried out and future work foreseen by the author is also discussed. Conclusions on the benefits of neural works and its limitations are highlighted. Recommendations for future research are suggested and the prospect of having data visualisation and virtual reality in the future is examined.
Research work already undertaken has explored the potential application of neural network techniques to the analysis of beam-column joints by demonstrating the effectiveness of the backpropagation algorithm. Conventional computer programs have traditionally addressed problems for which solutions are already formulated explicitly. Artificial neural networks on the other hand, are computing systems which address problems analytically and the solutions for which have not been formulated explicitly. The merit and the flexibility of these techniques are demonstrated throughout the thesis.
Figure 1.3. Strategies for Implementations of Predictive Stages.
CHAPTER 2

SYMBOLIC LOGIC BASE LEARNING

2.1. ARTIFICIAL INTELLIGENCE: GENERAL VIEW

Speculation about creating a "thinking machine" dates back to the nineteen century. It was more than forty years later with the introduction of the digital computer that interest in Artificial Intelligence (AI) emerged with great expectation and over-enthusiasm, the scientific community thought that an imminent move from the conventional algorithmic method to symbolic reasoning was about to happen. The AI field is a sub-field of computer science which links with linguistic and psychology ingredients, concern about the study of human intelligence and behaviour in order to design computer machines that emulate human intelligence and thinking. This field is concerned with an understanding of human intelligence behaviour and how it can be transformed in an intelligent way. It also deals with the problems of modelling human intelligence in an effective way. AI mainly focuses on the mechanism through which human behaviour improves with time. Early work on AI explored ways of dealing with the problem of modelling human intelligence.

The main goal of AI is to understand the way in which humans perform complex functions such as language understanding, learning capability and reasoning. This requires the projection of intelligence behaviour on machines. The heuristic - rules of thumb - is the basis for AI programming technique which exhibits an increasingly complex codes to simulate human thinking. The growth of AI is occurring in many fields for example, the development of expert systems in the field of medical
diagnosis, the implementation of natural language processing, computer vision, speech recognition, theorem proving, engineering and many other disciplines. After two decades of institution research the commercial trend has developed throughout every branch and field of study.

Human intelligence is difficult to define but easy to recognise. However a machine can determine whether it is capable of thought, as proposed by Turning [18] in 1963. The test involves an interrogator and the machine which is to be evaluated. The role of the interrogator is to determine which is the machine and which is the person. Communication between them is carried out in separate rooms and the questions are communicated by typing. The following definition was suggested by Barr and Feigenbaum [19]:

"Artificial Intelligence (AI) is the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior - understanding language, learning, reasoning, solving problems, and so on."

Rich [20], proposed the following definition for AI:

"Artificial Intelligence (A.I) is the study of how to make computers do things at which, at the moment, people are better."

A simple definition was provided by Bock [21] in which he describes AI as:

"Artificial intelligence is the ability of human-man machine (an automation) to emulate or simulate
understanding human learning is essential in AI research to develop intelligent systems. Efforts in applying AI problem solving have led to the development of expert systems. AI technology offers the potential transfer of human intelligence functions to machines. Interest in AI spread worldwide in 1982 after the Japanese launched a ten-year intensive program to produce a computer system based on the knowledge-information processing system (KIPS). They achieved their long-term aims by building a sophisticated and new kind of computer. The project is now well known as the 'fifth generation' computer. Today's most successful and visible implementation of AI is the development of expert systems with practical applications in many fields.

2.2. EXPERT SYSTEMS: AN OVERVIEW

The expert system (also known as knowledge-based system) field is a branch of AI which is concerned with the implementation and codification of human intelligence knowledge into sophisticated computer programs which simulate human performance expertise in solving problems within the expert-specific domain of knowledge. Expert systems are special programs that reason with symbolic information and use the heuristic inference engine to provide a high performance. Expert systems fall into the category of computer programs that can advise, consult, categorise, diagnose, plan, forecast, recognise, learn and provide explanations for their complex reasoning. Also, expert systems are invented and implemented to act as advisers, consultants and assist in solving the human need's problems. This type of systems technology attempts to use, replicate, emulate and manipulate knowledge to solve efficiently and effectively problems which usually require experts in a specific domain. The most useful feature of expert systems is the high-level expertise they provide for problem solving. Expert systems are intended as expert
adviser to the non-specialist for solving problems. It is also possible for the knowledge of many experts to be placed in one system. One of the best characteristics of an expert system is the use of large bodies of domain knowledge which contain procedures and facts from several human experts. Figure 2.1 shows the main components of a simple expert system with a knowledge-base that contains a database of facts and rules, and a symbolic inference engine with a reasoning mechanism which selects applied facts and rules in a controlled way.

![Diagram of expert system components](image)

**Figure 2.1. Main Components of an Expert System.**

The function of the inference mechanism is to compare information it receives with knowledge found in the database and make deduction according to logical rules implemented within the system. The knowledge base, on the other hand, contains rules that manipulate the facts. These two components are separate distinct parts of an expert system, but they complement each other. Forward and backward chaining are the most widely accepted techniques for controlling the search strategies in the inference engine. Forward chaining, known as bottom-up or data driven, depends on the system for reaching a conclusion, while backward chaining, a top down approach, is goal directed whereby the system works back from a conclusion to reach a goal.
The inference engine attempts to reach a conclusion by two different approaches. The depth-first search follows successive nodes before attempting to search an alternative branch, while breadth-first search crosses nodes to reach a conclusion.

The process of building an expert system is known as knowledge engineering. A knowledge engineer interviews an expert, or experts, to transfer their knowledge to the computer system. A knowledge engineer usually has special expertise in multidisciplinary areas involved in building the expert system by interviewing the experts and representing and organising the knowledge accordance to the software. The role of the knowledge engineer is to interpret between the computer and the expert by coding their knowledge into accepted and appropriate knowledge representation. There are several ways of representing knowledge and the most commonly used knowledge representation techniques available are production rules, semantic networks, frames and first-order-logic. A combination of two or more of these knowledge representations is known as a hybrid system. There are two main tasks for the knowledge engineer, first to acquire or elicit knowledge from the expert to build a knowledge base - this process of collecting data is known as knowledge acquisition or elicitation. The second task is knowledge representation, where the information (facts and rules) is stored in the computer or more precisely into the knowledge base. The performance of an expert system is measured by the manner in which it explains the results to the user.

The application of an expert system is usually in the area where the problem is ill-defined such as those encountered in structural design where require intuition and experience. Expert systems technology today has moved from the prototype in research institutions to the production applications and tremendous changes in the way we approach problem-solving.

2.3. MEANING OF LEARNING
Learning is a broad definition which covers a wide range of information and meaning and can be recognised as a form of problem-solving. It can be viewed as an essential component of understanding, visualising, memorising, planning, organising and debugging. Learning is a dynamic process of permanent change that results from activity, observation, memorisation, induction, deduction and training. Learning causes changes in behaviour with constant activity and recalling. Learning ability without doubt is an ingredient of human intelligence. Researchers have paid much attention to the learning process since as early as the invention of the digital computer.

Understanding human learning is the key for AI research to implement the learning machine process. The objective is not only to understand how people learn, but to provide a basic and meaningful process for transferring human learning to the machine without loss of knowledge and information. The lack of computers meant that AI research had to provide a fundamental methodology to transfer the rich knowledge that humans possess to the computer.

Learning is often described as the improvement of achievement with experience - although this is a rather general definition, a more precise definition is difficult to establish. The ability of the human to receive a tremendous amount of information permits us to adapt to the changing environment, to develop a great variety of skills, and to acquire expertise in vast number of specific domains. The ability of the human to learn and acquire new knowledge is so remarkable that they are often capable of learning from information supplied from multiple sources and expressed in a variety of forms. Learning processes include the fundamental gaining of new declarative knowledge, the development of skills through instructions, practise, and the organisation of new knowledge into a general acceptable format. Learning may be categorised as the possibility of generating new ideas and concepts from old ones. Over the years many researchers have attempted to define the meaning of 'learning'.
but the one on which most AI researchers agree is the definition by Simon [22] of Carnegie-Mellon University:

"Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time."

Minsky [23] with his more general view of learning defines it as:

"Learning is making useful changes in our minds."

Where Michalski [24] defines learning as:

"Learning is constructing or modifying representations of what is being experienced."

Scott [25] suggested a revised definition of learning:

"It is any process in which a system builds a retrievable representation of its past interactions with its environment."

While Forsyth [26], an advocate of the evolution approach, defines learning as:

"When a computer system improves its performance at a given task over time, without re-programming, it can be said to have learned something."

The idea of defining and presenting the word 'learning' is essential and important for any learning technique to be implemented on a machine. All types of symbolic manipulations, in one way or another, try to emulate and simulate human
intelligence thinking by programming the intelligence capability into the machine in an efficient and flexible way.

Understanding the internal human learning ability is likely to lead to more ambitious and robust machine learning capability.

2.4. MACHINE LEARNING BACKGROUND

The basis of machine learning capability lies in the way the knowledge elicitation and representation is coded into the machine. McCarthy [27] gave a simple but powerful definition for machine learning:

"In order for a program to be capable of learning something it must first be capable of being told it."

Michie [28] of the University of Edinburgh, well known for his contribution to machine learning gave the following definition:

"By 'machine learning' we mean the modification or construction by program of stored information structures, so that the machine-deliverable information becomes one of the following:

- More accurate
- Large in amount
- Cheaper to obtain."

Forsyth [29] defines machine learning and the requirements of the learning algorithm as follows:

"By machine learning I refer to any automatic improvement in the performance of a computer over
time, as a result of experience. Thus a learning algorithm seeks to do one or more of the following:

- cover a wider range of problems
- deliver more accurate solutions
- obtain answers more cheaply
- simplify codified knowledge."

The field of machine learning has received much attention and developed to play in the recent years particularly in the past decade. One of the main impediments to build an expert system is the acquisition of knowledge, since interviewing an expert is a difficult and tedious process. Usually the knowledge engineer and the expert must work together in the context of solving particular problems. The expert encounters difficulty in explaining how a certain conclusion was reached. At same time the knowledge engineer has difficulty in coding and transforming the expert knowledge into a suitable computer code. It is widely acknowledged by the artificial intelligence community that the knowledge acquisition is a major "bottleneck" of building and constructing an expert system. Feigenbaum [30] recognised this problem and described it as:

"There are many important problems of knowledge representation, utilisation and acquisition that must be solved, but the acquisition problem is the most critical "bottleneck" problem."

Waterman [31] has experienced the difficulty of interviewing an expert and describes the problem as:

"The knowledge Acquisition Process:
This interaction consists of a prolonged series of intense, systematic interviews, usually extending over a period of many months."

Building an expert system is a difficult process, characterised by using a large body of domain knowledge, which results in time consuming especially at the stage of debugging the knowledge base. Recognition of the AI discipline in the mid 1950s opened the opportunity to research in machine learning. Machine learning provides an alternative way of building an expert system to overcome the knowledge elicitation bottleneck, thus avoiding potential problems and eliminating the knowledge engineer's task of interviewing the expert by designing a machine in such a way to learn its heuristics.

AI research communities for many years have explored ways of enabling computers to learn and extract concepts and relations from a set of database. In response to the difficulties encountered encoding large amounts of knowledge, AI researchers have turned their attention to machine learning as an alternative means of overcoming the difficulty of knowledge elicitation. Machine learning researchers attempted to instruct the computer by exploring alternative learning mechanisms to reduce the burden of hand programming. AI research for many years has explored ways of enabling computers to learn and extract concepts and relations from collective databases through use of a friendly interface session with the computer. Machine learning is acknowledged as a difficult and complex field, in which progress is very slow, and over the past decade some general approaches have emerged. However, the most active area of research in the AI field is machine learning.

The main objective of machine learning is to implement symbolic-logic learning capability into the machine in order to acquire new knowledge, learn new skills, and improve with practice. A further objective is for automated knowledge elicitation to improve computer system performance over time which will eventually result in a
more experienced machine. This will produce a new type of knowledge-based system. The incorporation of learning mechanisms into the system provides the potential for alleviating the knowledge elicitation process, improving the system performance and shortening the debugging and system maintenance.

The first international workshop on machine learning was held at Carnegie-Mellon University in Pittsburgh, Pennsylvania, in July 1980. This workshop laid the foundation and direction for research in this field. Several important papers were presented at this workshop, and from these a new book entitled "Machine learning: An Artificial Intelligence Approach" by Michalski, Carbonell and Mitchell [32] was published. This book is considered by AI researchers to be the "BIBLE" of machine learning. An additional workshop was held in June 22-24, 1983 at Allerton House, Monticello, Illinios, and from this workshop Volume 2 [33] published the same authors. A third workshop was held in June 1985, at Skytop, Pennsylvania, and a new journal went to press in 1986 under the name 'Machine Learning' providing a substantial amount of information for the AI community. In 1988, a Machine Learning International Conference was held at the University of Michigan, Ann Arbor. Similar workshops and conferences have been held in Europe, with the first workshop on machine learning held under the auspices of the European COST13 project.

2.5. MACHINE LEARNING STRATEGIES

A classification of machine learning strategies can be arranged according to the amount of inference the learning system performs while processing the provided information. Intensive research is under way on machine learning by the AI community. Current research on machine learning however is diverse in nature within the research groups or organisations, making the exchange and sharing of information difficult namely due to the use of different notations, procedures and terminology. Moreover, it is also difficult for new researchers to join a research
group. Figure 2.2 shows the five main machine learning taxonomies which are based on underlying learning strategies.

Figure 2.2. Machine Learning Strategies.

2.5.1. Rote Learning

This is the simplest and easiest kind of machine learning since it stores new information, facts and experience without generalisation, [34]. It is the most elementary form of learning where no inference or other transformation of knowledge is carried out between the teacher and the learner, since the information is acquired and not processed. This is a straightforward and direct implantation of new knowledge by data recording relying entirely on memorisation. The knowledge is transformed from the teacher and accepted and memorised by the learner. There are two types of learning knowledge acquisition methods available:

2.5.1.1. Learning by Being Programmed

The learner does not get involved as the data is programmed directly into the computer.
2.5.1.2. **Learning by Memorisation**

The information is processed without any inference. It is a function entirely as primitive database system.

2.5.2. **Learning by Instruction or Learning by Being Told**

The knowledge is acquired from the teacher or from other sources such as textbooks. This form of acquiring knowledge is similar to the educational system [34,35]. This is a general strategy of human learning where knowledge is acquired through teachers, books, publications and many other sources. The knowledge is generally provided in a form which is recognised and accepted by the computer. For example, solving algebraic equations of unknowns, the teacher might tell the student to place the unknowns on one side and the knowns on the other side. The strategy requires some inference where the student is still left to proceed with his own strategy. The machine task for this strategy is to build a system that can accept instruction and advise and perform knowledge effectively for different tasks.

2.5.3. **Learning by Analogy**

This strategy requires knowledge acquisition in a specific domain to transfer or extend to a similar task in another domain [34,35]. This technique is based on recalling from previous experience and mapping it into a new domain. The concept is based on the transformation of previous extracted knowledge and applying it to solve similar situations. This strategy involves both deduction and induction. Analogy reasoning is a robust requirement from the system which relates new ideas and concepts to the old by modifying already existing concepts. More inference is required from the learner than for rote learning or learning by instruction. The strategy is more aggressive as the systems try to find a case or solution that is relevant to the current situation. For example, learning to drive a bus when a person already knows how to drive a car. The process involves transferring previous
knowledge to the new domain. Learning by being reminded is another form of learning by analogy. The process of analogy involves exploring previous experience and applying it in future in a flexible manner.

2.5.4. Learning by Induction

Induction is the process of extracting a general concept from the study of specific instances, i.e., formulating a general law that governs particular cases [34,35,36]. It is a bottom-up approach where general rules are obtained from a set of examples. An inductive learning strategy is based on performing an inductive inference on the information provided to the learner. Inference learning is a process of generalising from examples, and moving from a particular to a more general concept. The inductive inference technique is an alternative means of articulating knowledge.

Inductive learning can be used to refine an existing knowledge base. Another application of inductive learning is to generate a meaningful classification of a given set of data and organise the collection of rules. The requirements for obtaining reliable induced rules based mainly on the quality of the following components are:

- Examples; a set of examples or training sets that form the basis for the induction learning process. The training set consists primarily of a database where the solution for a particular case is known. Good examples will provide good rules whereas inadequate examples may result in poor rules.

- Attributes; provide the characteristic of each example to establish comparison and distinction between the set training. The attributes can be either discrete or continuous numbers.

- Classes; usually, each class can be distinguished or classified to represent specific decisions required by the expert.
There are two basic forms of inductive learning: learning from examples (concept acquisition), and learning by observation and discovery (descriptive generalisation or concept formulation without a teacher).

2.5.5. Learning from Examples - Supervised Learning

This strategy has the longest history and has been the most heavily investigated by the artificial intelligence researchers [35]. This type of learning is the most frequently studied and most popular form of inductive learning. Knowledge acquired through a given set of examples is called a training set and the concept is generalised from the description of the examples. The rule of the learner is to induce a general concept description from a set of positive and negative examples. The set of training examples is introduced as an input to the learning algorithm, where the rule is derived. More inference is required from the learner than when learning from analogy. This type of learning may be subcategorised according to the source of the examples:

- Teacher is the source; the teacher knows the concept and generates examples to help the learner.
- Learner itself is the source; the learner knows the basic knowledge but does not know the concept to acquire.
- Environment is the source; the learner depends on an uncontrolled observation where the process of generation occurs randomly.

Also, learning from examples may be classified according to the examples available to the learner:

- Only positive examples; the concept to be learned relies upon a prior knowledge where minimal generalisation is required.
Positive and negative examples; these are the typical types of example used for generalisation. The positive examples provide generalisations and the negative examples prevent over generalisation.

2.5.6. Learning by Observation and Discovery - Unsupervised Learning

This is an unsupervised type of learning where learning is acquired without the supervision of a teacher [35], and knowledge is acquired without help from anyone else. Generally, the learner has the ability to bring together all forms of learning, and to generate, test and validate new ideas and theories. The process may perform a large amount of deduction of new theories, finding patterns in the information classification and recognising certain phenomena between a group of classes. This type of learning requires a large amount of inference from the learner to approach a conclusion. It is a strategy that collects facts and information (observations) and tries to group them into a general description (law). A sub-classification for learning by observation is available according to the degree of interaction with the external environment. These are passive observation and active experimentation.

2.5.7. Learning by Deduction

This is a top-down approach where reasoning varies from the general to the specific [37]. The learner has the ability to reason and infer from previous general knowledge, and can draw conclusions and find out what knowledge is required from what it already available. The process of deduction involves generating rules from which it is possible to deduce facts about specific instances. Explanation-Based learning is a new form of learning that has emerged in the last few years as a promising approach for building better knowledge-based systems. This type of learning solves problems initially by explaining, then generalises by retaining the examples that suit the explanation.

2.6. INDUCTIVE LEARNING ALGORITHMS
The emergence of machine learning has provided several algorithms for developing and constructing knowledge bases. These algorithms are based on mathematical theories, where their primary functions are to build knowledge bases from a set of training examples, and induce decision trees or production rules. A satisfactory number of programs have been developed based on induction learning, particularly learning from examples. Machine learning algorithm developments have useful applications in a wide range of engineering fields. Figure 2.3 shows the most frequent use and available inductive learning algorithms.

**Figure 2.3. Learning Algorithms.**

### 2.6.1. Concept Learning System (CLS) Algorithm

The CLS an algorithm, originally developed by Hunt, Marin and Stone [38] in 1966, led to the construction and development of new knowledge based systems of decision trees from a set of training examples of cases with known classes. The CLS is a mechanism for construction of a decision tree that attempts to minimise the cost of an object classification. Generally, decision trees provide computational efficiency means by realising the dependence of structural variables and their relations on other variables. A decision tree simply applies a recursive structure that expresses a sequential approach. The leaves of the decision tree represent the class names. The basic structure of the CLS is the implementation of the Top-Down
Induction of Decision Trees (TDIDT) which can be used extensively over a range of applications and gives impressive results. The primary role of the induction rule is the formulation of (IF...THEN...) rules from a set of training examples which define the attributes and specify classes that can be mapped to the decision tree. A number of algorithms use this technique, but the one most recognised by the AI research community is the algorithm developed by Quinlan [36].

2.6.1.1. Quinlan's ID3 Algorithm

The inductive learning algorithm proposed by Quinlan [36] is based on the statistical theory of information. Quinlan's ID3 (Interactive Dichotomizer 3) is a descendent of the CLS family and uses backward reasoning to induce a decision tree from a set of training examples. The procedure employs the powerful divide and conquer technique. The nodes of the decision tree represent the attributes and the branches correspond to the values. The main benefit of ID3 is the construction of a knowledge base that results in a reasonable decision tree without the need for intense interviews with the expert over a period of many months. The advantage of this algorithm is that it provides a practical method for knowledge acquisition than which is easier for the expert to present a set of examples instead of describing decision rules. It requires a complete set of training examples to induce a reliable decision tree. The merit of this algorithm lies in its simplicity and it computational efficiency. However, this algorithm always produces deterministic rules and it is not robust particularly where data is uncertain or incomplete. Usually, the algorithm clashes or eliminates when confronted with two contrary examples. The selection of training sets and attributes play a major role in the quality of information that the algorithm provides, so careful attention is recommended to select the composition of examples and its attributes. The performance of this algorithm can be tested and evaluated by several examples and by comparing results with the expert. The algorithm has been implemented for different applications in industry such as the British Petroleum's
system for gas-oil separation as well as in the fault diagnosis system for printed circuit board. Several commercially expert system tools based on ID3 algorithm are on the market, (Figure 2.4) and it has been used mainly in the knowledge acquisition domain. The next section contains brief descriptions of the commercial tools that are widely used in different applications.

![Figure 2.4. Commercial Tools.](image)

### 2.6.1.1.1. Expert-Ease

Expert-Ease is the smallest and simplest inductive tool that can be used and evaluated, using a spreadsheet format to construct a knowledge base from examples. It generates procedures for problem solving using the decision tree to represent the procedure, was developed by Michie, University of Edinburgh, and is a PASCAL based tool [39]. Launched in 1983 on an IBM-PC platform, it is available in different versions. The rules induced are in the form of a decision tree based on logical induction, and can be displayed in graphical form. The tool provides the user
with a menu but lacks an explanation facility. The maximum number of training sets is of the order of 250-350 examples, with an attribute of 31, either integer or logical. Expert-Ease is marketed by Human Edge Software and Expert Systems Inc., USA. One of its drawbacks is the limit of 128K of memory, also it does not handle probabilities.

2.6.1.1.2. EX-TRAN 7

Ex-tran 7 is very sophisticated software, programmed in FORTRAN 77, and marketed by Intelligent Terminal, Glasgow. The tool contains two parts, an Analog Concept Learning Translator (ACLT) and rule a driver (DRIVER), [40]. The ACLT represents the main engine and is responsible for checking and managing the examples supplied by the user. The DRIVER runs the rules generated by the ACLT. The induced rules, converted into FORTRAN code, can be linked to a FORTRAN program and compiled. The editor feature of EX-TRAN uses decision trees to represent rules from a set of examples and the inference engine provides global backward chaining and local forward chaining. ACLT translates the satisfactory rules into FORTRAN code for use in other program. National Aeronautics and Space Administration [41], NASA, has used this software intensively for deriving decision rules from the large amount of test data obtained every time a space shuttle's main engine undergoes a test fire.

2.6.1.1.3. RuleMaster

RuleMaster is a general purpose expert system domain-independent development tool, developed by Michie [42], University of Edinburgh, using induction rules as well as being rule-based. This inductive learning system provides and allows for effective knowledge elicitation. It is written in 'C' and can be transported to other machine platforms, such as the Unix work station. RuleMaster automatically induces decision rules from a set of examples supplied by the user. The examples are entered and coded into a sheet similar to the spreadsheet format.
The RuleMaster environment provides a friendly user interface with pull-down menus and can be accessible to a non programmer. The induced decision tree takes the form of \( \text{IF} \ldots \text{THEN} \ldots \text{ELSE} \) from the logic of examples. RuleMaster deals with uncertainty by implementing the fuzzy logic capability to evaluate input and arrive at a conclusion. Rulemaster contains three main components:

1. **Rulemaker**: an induction algorithm which uses the knowledge as a set of examples and transforms them into a decision tree or set of rules. The basic feature of Rulemaker is the induction of rules from sets of examples provided by the expert. The knowledge is supplied in a declarative form.

2. **Radial**: is a procedural language similar to the PASCAL structure format. The knowledge is presented and entered as a form of rules. The Radial language applies two main reasoning mechanisms, a forward chaining and a backward chaining.

3. **User interface**: sophisticated pull-down-menus which include editors, file management, explanation facilities and utility menus.

2.6.1.1.4. **C4**

Quinlan's C4 [43] is an inductive rule's tool successor of ID3. It is an enhanced form of ID3 inductive tool which it builds decision trees from a large collection of examples and provides an improved output. The induction tool, can generate explicit rules and give a measure of the strength of each individual rule. This is an essential criterion for deciding whether such a rule can be ignored or used in further work. There is a significant performance improvement in this algorithm compared to that for ID3. This tool is programmed in 'C' and provides an optional facility for transfer decision trees to a set of production rules. The advantage of this tool is its capability of dealing with continuous as well as discrete attribute values. C4 tool has been
implemented in several domains, mainly in the diagnosis of thyroid disease, where it was used to synthesize accurate rules from a large collection of data.

2.6.1.1.5. ASSISTANT and ASSISTANT 86

ASSISTANT [44] is an inductive learner based on ID3 and programmed by a Yugoslav group. Its performance improved when the decision tree was pruned by the program. The rules induced by ASSISTANT are in the form of decision trees. ASSISTANT 86 [44] is an enhanced form of the previous program, programmed in PASCAL language and implemented on an IBM-PC which can be used to generate decision rules. It contains a user-friendly facility interface for interactive use and displays the decision tree in a graphical form. It was used to construct knowledge bases in the iron and steel industry, and also implemented in building medical diagnosis expert knowledge-bases.

2.6.1.1.6. Super-Expert

Rules are created automatically by Super-Expert [44] from a set of training examples where the forward and backward chaining is part of this tool using induction rules to reach a conclusion.

2.6.2. Darwinian Principles

The second group of inductive learning algorithms was proposed by Forsyth [45] and Michalski [46]. These are based on the evolution concept by using Darwinian principles to generate decision trees. The commercial tools based on this algorithm are shown in Figure 2.4.

2.6.2.1. AQ Algorithm

The AQ [46] algorithm follows the top-down approach for constructing rules. These rules are in the form of discriminant descriptive. It accepts two types of preclassified training example.
2.6.2.1.1. AQ11

This is an inductive program tool developed by Michalski and others. The rules are generated by AQ11 [46] in a language called Variable-Value Logic calculus (VL1) which follows the top-down approach for reaching a conclusion and uses a propositional style description. These rules start in a general form and become more and more specific. Successful results have been reported using this tool in a wide variety of applications. A successful result was reported by Michalski and Chilausky [47] using this algorithm in a soybean disease experimental analysis where two sets of rules were obtained, one set from the expert and the other induced from a set of examples. It was observed that the rules induced by the set of examples behaved better than the rules induced by the expert. The training set of examples used 290 cases of known soybean plant disease. The AQ11 produced a set of rules containing fifteen types of different soybean for the diagnosis process. This tool can perform analyses using numerical and logical attributes.

2.6.2.1.2. BEAGLE

Biological Evolutionary Algorithm Generating Logical Expression (BEAGLE) [48]: is a rule induction system tool based on the Darwinian approach to pattern recognition. This software package produces decision-rules from a database. The theoretical foundation was developed by Forsyth [48] and produced by Warm Boot Ltd in Nottingham. It accepts examples in the form of a database and produces two distinct output files, a decision rule file from the provided examples and a file that express the rules in FORTRAN or PASCAL languages. The rules induced are Boolean Expression represented by a tree structure. The software package is programmed in PASCAL and consists of six parts. HERB program (Heuristic Evolutionary Rule Breeder) contains three input files, and LEAF (Logical Evaluator And Forecaster) is a rule expressed in a subprogram. STEM (Signature Table Evaluation Module) program combines different rules generated by HERB. SEED
(Selectively Extracts Example Data) is a preliminary program that can be implemented for external use suitable for BEAGLE format. ROOT (Rule-Oriented Optimisation Tester) is a batch tester designed for the programmer's own use before any learning process starts. PLUM (Procedural Language Utility Maker) is a utility package with the primarily task of transferring BEAGLE code into an executable format such as FORTRAN, C or BASIC languages. BEAGLE is a special-purpose tool which uses a genetic induction strategy to predict and classify. Forsyth claims that his method provides better inductive processing than statistical data analysis or any other alternative inductive system.

2.6.3. Rough Sets Theory Algorithm

The third group of inductive learning algorithms was developed in the Computer Science Department of the University of Regina, Canada, and is based on the theory of rough sets which is a mathematical tool dealing with the occurrence of uncertainty in solving problems related to machine learning. It is use as a basis to analyse data for rule acquisition.

2.6.3.1. PLA Algorithm

PLA is the Probabilistic Learning Algorithm [49]. This algorithm is inductive and its derivation is based on the probabilistic model of the theory of rough sets. The algorithm produces decision rules from non-deterministic information. The rules obtained generate a decision tree with the capability of assigning factors to each rule. The products of this algorithm are explained in more detail in section 2.7.

2.7. GENERAL VIEW OF MACHINE LEARNING IN CIVIL ENGINEERING

Interest in machine learning in the field of civil engineering dates as far back as 1966 when Spillers [50] published a paper entitled "Artificial Intelligence and Structural Design". It was an attempt to demonstrate the most elementary learning
capability in structural design. His primary interest was to apply to structural design. His main consideration was to demonstrate the possibility of using examples to generate rules. As he suggested [50]:

"Examples play an important part in teaching and learning in humans and it is natural to ask what part they should play in the formal adaptive system. Because students are taught using examples, should computers be taught through the use of examples, and how? Again, the answer is not now available but perhaps examples may be used as complete sets of independent vectors are now used to represent functions."

This shows an early attempt by Spillers [50] to use examples as a means of generating rules, and confirms the fact that what is available today in machine learning applications is merely an illustration of what was said before.

The application of inductive learning systems in civil engineering is at an experimental stage and a very few published papers are available. Learning from bridge structural failure was proposed by Stone [51], whereby a hierarchical knowledge-base could be developed to predicate bridge failure from a study of historical data. The method of learning employed was based on an algorithm developed by Norris, Pilsworth and Baldwin [52]. The roots of this learning technique emerged from the medical diagnosis field where the relation between the symptoms and the disease is observed for a number of patient case histories. The learning process involves two phases: a *discrimination* analysis that is known as the serial approach and the *connectivity* algorithm which adopts a parallel approach. In the discrimination case, a search is carried out for a single feature, while the connectivity algorithm search is for a group of features. The presence or absence of
such feature or features, gives evidence that an element or elements belong to one or another class. In case of uncertainty encounter, the method of support logic is used.

In the United States, one of the most active researchers in this field is Professor Arciszewski of Wayne State University, Michigan, who has published several papers [53] on the possibility of applying machine learning to the solution of civil engineering problems. These documents have provided a forum for further research in this field. However, further research is required to develop new concepts, utilise alternative procedures and to standardise methodology in the use of such ideas in the field of civil engineering. The methodology of inductive systems used by Arciszewski is based on the rough sets theory.

At the University of Sydney, research in machine learning has been conducted in the area of characterised design from a set of learning examples. McLaughlin and Gero [54] applied the induction concept algorithm to abstract rules from the relationships of performance and decisions. The training set of examples is a set of characterised designs used to induce an optimal solution to the design problem. The ID3 algorithm developed by Quinlan was used in this work.

2.7.1. Inductive Systems Tools

Civil engineering inductive system tools are computer software packages that use the learning from examples to extract a system of decision rules. These tools are new, and they have a potential use for different types of application in civil engineering. They are referred to as the Black Box which relates input (stimuli) to output (responses). Engineers are mainly concerned about the most effective way of extracting decision rules or production rules without getting involved in the computational efforts required in developing such a tool. New procedures will eventually be required to develop certain control criteria and methodology in order to minimise the time required to extract decision rules from a set of training examples.
The application of inductive systems to the area of civil engineering is at an experimental stage but the results demand future research in this area. According to Arciszewski [53], there are at least five potential applications of inductive systems in civil engineering which have already been identified:

1. Knowledge acquisition, or extracting decision rules from examples; this involves generating very complex decision rules with a large number of attributes. Traditional methods for acquiring knowledge are time consuming and expensive.

2. Inductive problem-solving; for use in complex engineering problems where traditional analytical methods have failed to provide adequate solutions. This method extracts decision rules from a set of training examples to find an unexpected rule or missing link in a particular problem.

3. Inductive shallow modelling; this type of inductive learning is based on rational observed behaviour and valid in a specific defined domain. The inductive tool, ANLYST, shown in Figure 2.4 has been used for this approach by Arciszewski [55], who identified relationships between attributes describing cold-rolled steel beams under bending and attributes describing deformations of the magnitude of what had been obtained experimentally.

4. Inductive learning about domains; i.e. decision rules extracted gradually from a set of training examples for a specific domain. This method consists of three stages:

   • Provide examples to the systems as input: training sets.
   
   • Inductive learning process: the process of induction which generalise from specifics to general.
   
   • Inducing decision rules: the conclusion which is reached after a particular problem is subjected to a process of induction.
5. Conceptual design or learning expert system: the inductive systems have the ability to follow a dynamic changing conditions by modifying its own decision rules. Conceptual design has the capability of producing innovative concepts, and ideas, producing standards, and establishing well-known concepts. Arciszewski, Mustafa and Ziarko [56], used this BRZDY1 tool, Figure 2.4, to acquire heuristic rules and to provide concept acquisition based on inductive learning from examples that induce decision rules represented by means of qualitative variables. BRAZDY1 is an inductive tool package under experimentation for use in structural design.

2.7.2. Machine Learning: Final Remarks

Machine learning is a new and powerful method for exploring and developing the deficiencies of traditional expert systems for knowledge acquisition. Machine learning is at an early stage of research and most of the available literature lacks standardisation, methodologies, and procedures. Learning from examples has been covered and investigated more than any other learning strategies. Reducing the random search in the space of problem solving is the key to making the machine behave intelligently. Several algorithms have been developed for extracting rules from examples. Quinlan's ID3 algorithm is the most accepted method used by artificial intelligence researchers, and few commercially available tools produced use this learning algorithm.

In the area of civil engineering few papers have been published and most of the inductive tools are under experimentation. Stone [51] laid down a methodology for designing a bridge based on past failures and learned from that experience. It is a mixture of learning from examples, by providing data in a the form of training sets, and as the same time applying the analogy approach indirectly. Arciszewski provided a forum for further research in machine learning that is related to conceptual design in structural analysis. His work was based on the rough sets theory. McLaughlin
and Gero of the University of Sydney [54] used the ID3 algorithm for design conceptual knowledge acquisition and the design of characteristics, including learning from examples for optimal design solutions.
CHAPTER 3

PARALLEL DISTRIBUTED PROCESSING BASE LEARNING

3.1. INTRODUCTION

The emergence of the electronic computer attracted researchers from a wide variety of disciplines and its development accelerated rapidly in many fields. The wide area of applications with varying background resulted in the publication of different terminology and equations that varied accordingly. The aim of this chapter is to present the concept of parallel distributed processing base learning in artificial neural networks and outline the theories, methodologies and architectures of selected algorithms, also to present consistent terminology, simplified equations and emphasise the backpropagation algorithm which is essential to the author's research.

Conventional methods based on von Neumann programming instruction have established the basis for all computation. However, this method is computationally intensive and serial in implementation so that it does not parallel the human way of thinking and abstraction.

The symbolic based learning approach reduces the computation process in the presence of huge amount of data and has a weakness in knowledge representation. Symbolic logic base implementation is extremely difficult, particularly in the area of intensive mathematical computation.

Neural network fields provide a computational paradigm challenge which has resulted from four decades of intensive research and investment. The function of the
human brain contains many features that can be simulated in a machine to perform certain tasks that are difficult to achieve by conventional methods and symbolic approach. One of the primary intentions of the distributed parallel community is to design new hardware and software on a computer which can simulate human thinking.

3.2. HISTORICAL BACKGROUND

The principal goal of artificial intelligence researches is to develop computer programs which emulate some form of human intelligence. This field of study is concerned with understanding human intelligence behaviour, how it can be transformed in an intelligent way, and also deals with the problem of modelling human intelligence in an effective way. Figure 3.1 shows how AI researchers have historically pursued two different approaches in an attempt to understand human behaviour and thinking.

![Artificial Intelligence Main Paradigm](image)

**Figure 3.1. Artificial Intelligence Main Paradigm.**

The first paradigm, based on Parallel Distributed Processing (PDP) which is also known as the artificial neural network, connectionism, neurocomputation and probably many other definitions, imitates the human brain's functions in making
decisions and drawing conclusions. A large part of the biological detail of the nervous system was eliminated in modelling the brain's activity in order to simplify the brain function. McCulloch and Pitts in 1943 [57] first attempted to compute a neuron-like threshold by deriving a theorem related to a model of the neuron system. Their theorem lacks the learning capability of the neuron system, but was adopted by many AI researchers involved in modern neural computing. In 1949, Hebb [58] proposed a learning rule theory based on updating the synaptic strengths between two neurons when both of them are active. Rosenblatt [59], combining the ideas of Hebb, McCulloch and Pitts, introduced in 1958 his perceptron model to calculate logical functions by modifying the connections between the synapses. Minsky and Papert [60] however demonstrated in 1969 the limitations of the perceptron as a linear classifier. This was a set back for the parallel distributed community and as result most researchers turned to the second paradigm, namely symbolic logic which was discussed in chapter 2. In 1982 Hopfield [61], a well-known physicist, introduced his Hopfield network. Rumelhart, Hinton, and Williams [62] laid the mathematical foundation of the backpropagation network by presenting a clear and concise learning algorithm for the multilayer artificial neural network.

3.3. BIOLOGICAL NEURAL NETWORKS

3.3.1. Brain Structure

One of the most important fascinations of the human brain is its capacity for learning, as well as for processing huge amounts of information and controlling the body movements. The human nervous system has a communication and control network which is located in the brain. The brain plays the major role among all our organs in monitoring, abstracting, visualising, deciding and controlling behaviour. Although the human brain is not fully understood, experimentation and observation of human behaviour have provided an insight into the mystery of the brain cells' functions.
Psychologists have attempted over the years to create new mathematical models of the human functional brain, whilst the computer scientists strive to construct and build a neural network that mimics the mystery of the brain activity. The characteristics of understanding brain functions motivated scientists and mathematicians in testing their theory by modelling the neuron and its connections to provide a better understanding of the human brain's function. They deliberately to develop computer programs which specifically mimic human learning behaviour.

Artificial neural networks are inspired by human biological neural networks whereby they capture the brain function manipulation to approach a specific problem by applying certain rules to achieve reasonable results. The study of artificial neural networks is founded on a semi-empirical base to model the behaviour of the biological nerve cell structure. The processing element in the artificial network is analogous to the nerve cell in the human brain. The brain is composed of dense nerve cells which are highly interconnected and estimated to total 100 billion neurons of different types which are constantly sending and receiving messages. These nerve cells are fundamental elements to the central nervous system and determine any action which is taken. Each nerve cell is connected to 10,000 nerve cells, perhaps more, resulting in an immensely complicated network. Figure 3.2 shows a simplification of the brain structure which is made up of three main parts: the cell body (Nucleus), the dendrites, and the axon.
The nucleus is in the middle of each nerve cell and receives information from other cells before determining the transmission of the message. The basic decision depends on the amount of electrical charge that the cell poses. The dendrites, which carries the signal in, are fine branches surrounding the nerve cell and connected to other nerve cells. Each individual nerve cell contains a single axon and has many branches. The axon, which carries the signal away, is an extension of the nerve cell and is electrically active. It is of slender shape and is a one-way conduit from one nerve cell to the axon terminals. The axon has the capability of electrochemically transmitting information that concerns the state of the nerve cell. The axon terminals are gaps where the electrical charges convert to chemical charges and they terminate at the synapse which is where the nerve cell to cell chemical reaction takes place.

3.3.2. Function of the Brain

The small nerve fibres act as telephone wires which carry signals throughout the human body. These fibres branch out from the brain nerve cells and are connected to
the rest of the body by the spinal cord. Information about the surroundings is monitored by several organs such as the eyes, nose, tongue etc. From these messages the brain receives all the information required to decide what action to take.

The brain regulates and coordinates all the voluntary and involuntary movements of the human body. Messages travel from one nerve cell to another by two means, an electrical means and a chemical means, at varying rates of 10 to 120 metres per second. Electrical messages move from one end of the long nerve cell to another without touching by means of a microscopic synapse gap. These electrical messages transform into chemical signals at the synapse gap where a sending cell meets a receiving one. The role of the synapse is important in the communication that takes place between the nerve cells. In theory, messages are ferried across the synapse by a chemical molecule substance, called a neurotransmitter, which ensures the translation of the electrical signals to chemical signals. The synapses are usually found between the dendrites and the axon, between two dendrites, between a cell nerve and axon or between two axons. The synapse activity involves the weighting of incoming signals and producing a summation which then fires if it is greater or equal to a threshold for the neuron. A process of weight adjustment is then carried out in the nervous system which is a sign of the learning mechanism progress. Finally, the threshold functions complete the energy of the incoming signals over the time and space.

3.4. LEARNING ALGORITHMS TECHNIQUES

Neural networks' technology is founded on a sound rigorous mathematical foundation that can be evaluated and compared to other general scientific fields. The technology is based on a sophisticated mathematical approach which uses differential equations, matrices and linear algebra.

Learning is a fundamental property for neural networks to generate their own rules based on information provided. The adaptation is achieved through the learning
rule that adjust weights in the connections between processing elements in response to the input data. This adaptability is the ability of the network to adjust its internal algorithm to satisfy the desired response.

Neural networks are trained by several training techniques which have developed from a common root and share many characteristics. The training algorithm can be classified into three main groups, supervised learning, unsupervised learning and reinforcement learning.

3.4.1. Supervised Learning

For supervised learning, the training data set is presented to the network with input patterns and desired patterns. The function of the algorithm is to modify the network internal states through the adjustment of the weights to obtain output patterns that are acceptable or consistent with the desired output patterns. The main objective of the learning procedures is to ultimately minimise the error between the network output and the desired output and hence arrive at an acceptable solution. This type of learning is required to train the network for several sets of data before it becomes operational. Supervised learning has achieved a good reputation for producing acceptable results in several practical applications. This learning rule is classified as:

- Hebbian learning rule: this learning algorithm proposed by Hebb [58], which does not include a mathematical formulation, is considered to be one of the most influential learning rule developed which is based on physiological and psychological research. The basic interpretation of this algorithm is that if two connected processing elements are activated at the same time, the connection between them is strengthened otherwise the connection is not modified.
• Perceptron learning rule [59]: this forms the basis for the modern algorithm used today and is limited to binary outputs. The learning algorithm is very simple in a form of error correction where the event set is cycled through repeatedly to arrive at the desired output.

• Widrow-Hoff [63] learning rule: this accepts continuous output data. The error between the network output and the desired output is used to modify the internal connections through the weight adjustment within the network.

• Generalised delta learning rule [62]: is an extension of the Widrow-Hoff [63] learning rule and is widely implemented, especially in the backpropagation algorithm. The algorithm depends on propagating an error signal from the output processing element backwards to the input processing element through the internal layer (hidden layer). This algorithm involves the adjustment of the weights so as to minimise the square of the difference between the network output and the desired output. The process of the event set is cycled repeatedly until an acceptably small error is reached.

3.4.2. Unsupervised Learning

Unsupervised, sometimes called self-supervised, learning is where the network is trained through its input data without the desired output, or where the network organise its internal state to model features found only in the input training data. The network internally adjusts its weights and monitors its performance without external influence. It searches for regularities in the input data and makes adaptations according to the network function. The advantage of this type of learning is its
capability to perform when there is a redundancy in the input patterns. The rule is classified as:

- Self-organising learning rule: developed by Kohonen [64] Helsinki Technical University, Finland, and which performs classification without desired output and predicts results based entirely on the training input data. The procedure is biologically motivated where the processing elements compete for the opportunity of learning from the input signals. The network has no external influence to adjust its weights. Instead, it reaches a conclusion and settles down as it approaches a stable response. The learning process is a competition between the various processing elements in the competitive layer due to the pattern of the input layer.

3.4.3. Reinforcement Learning

Reinforcement learning is the situation where less information is provided at the target in the form of right or wrong. This type of learning is sometimes known as learning with critic and is a special form of supervised learning because some information is provided at the desired output.

3.5. NEURAL NETWORK TRANSFER FUNCTIONS

Artificial neural networks commonly apply a transfer functions 'threshold' which specifies how the processing elements will scale its response to the incoming signals, and then produces the processing element's activation. The processing element will output a signal if the activation is strong enough, that is it passes certain threshold criteria. The output is usually of the form (0,1, -1, +1). The primary transfer functions used are:
1. Non-linear functions (Sigmoid, Figure 3.3(a) or hyperbolic tangent function, Figure 3.3(b)): most popular function used today and essential in the author's research.

2. Threshold logic function, considered to be in a form of binary of either (0 or 1)

3. Radial basis function.


5. Linear function

A wide range of complicated functions are available and can be used for various special purposes.

![Figure 3.3. Non-Linear Functions.](image)

### 3.6. NEURAL NETWORK ARCHITECTURE

Recent advances in hardware and software have influenced progress in many research fields. In the past five years, neural networks have progressed from an idea for academic research to engineering applications in almost every discipline. The first international conference on neural network, held in San Diego 1987, resulted in a renewed interest in neural network research. Within a short period of time, many
publications on neural networks had appeared both in conferences and in Journals. Interest in this field has led to the development of hardware accelerators and software simulators and there are today many types of learning algorithms, software and hardwares.

The main objective of neural network architecture is to use a weight adjustment as a medium for representing and manipulating information to perform complex tasks. Their essential role has been demonstrated by solving complex non-linear relationships even when the input data information is incomplete, noisy or less precise. The fields of application include, classification, predictions, data filtering, data association, optimisation and conceptualisation.

There are approximately 36 neural networks now available in the market and their primary focus is on adaptation and learning. The most popular neural networks are;

1. Perceptron network (Perceptron referred to as nodes)
2. ADALINE and MADALINE networks
3. Hopfield network
4. Backpropagation network
5. Kohonen network or Self-organising network or Self-Organising Map (SOM)
6. Counterpropagation network
7. Adaptive Resonance Theory (ART)
8. Bidirectional Associative Memories (BAMs)
9. Brain-State in-a-Box
10. Probabilistic Neural Network
11. Functional-link Networks (FLN)
Only networks essential to research have been presented in this thesis along with their learning algorithms and architecture.

3.7. PERCEPTRON NETWORK

This is the earliest and first serious attempt at network development by Rosenblatt at Cornell University in the mid 1950's [59], and constitutes an important step in the development of neural networks. It is noted for an elementary pattern recognition including feature extraction and classification. The network generally consists of three principal layers. The signal input layer is called retina, in analogy to human vision, and acts as input buffer. The second layer represents a feature detector and consists of associate or predicate cells. Each of these cells is connected to the input buffer and to the third layer which consists of decision cells. These decision cells represent the perceptron layer output where they receive their input from the associate cells. The perceptron employs a supervised learning technique in reaching a conclusion, by adjusting the weights between the feature detector layer and the perceptron output layer.

Despite the perceptron's limitations as a linear classifier, which is a procedure for finding the straight line that separate classes, the theory has been extensively studied and used. The perceptron laid the foundation for many other forms of neural networks and demonstrated the important logical starting point for research in neural networks. Figure 3.4(a) shows a simplified model for a single processing element which is the fundamental block for any network, while Figure 3.4(b) shows a perceptron network consisting of three layers which are used for feature detection.
3.7.1. Basic Rule

The operation of the perceptron network requires the output to process the element $y_{jp}$ determined by the weighted sum of the perceptron input signals, (a set of input signals' $p_1, p_2, p_3, \ldots p_p$ applied through a set of associated weights $w_{ji}^{new,p}$, $w_{j2}^{new,p}$, $w_{j3}^{new,p}$, $\ldots w_{jN}^{new,p}$) which then become the input to the next layer. Mathematically the summation is represented as:

$$s_{jp} = \sum_{i=1}^{p} P_i W_{ji}^{new,p}$$

The summation $s_{jp}$ is then compared with specific value, the threshold, such that:

$$\begin{align*}
\text{if } s_{jp} > 0, \quad & \text{then } y_{jp} = 1 \\
\text{if } s_{jp} \leq 0, \quad & \text{then } y_{jp} = 0
\end{align*}$$

The perceptron learning rule is modified according to the following relationship:

$$w_{ji}^{new,p} = w_{ji}^{old,p} + P_i \alpha_p (d_{jp} - y_{jp})$$

where:
\[(d_{jp} - y_{jp}) = \begin{cases} 
1 & \text{if } d_{jp} = 1 \text{ and } y_{jp} = 0 \\
0 & \text{if } d_{jp} = y_{jp} \\
-1 & \text{if } d_{jp} = 0 \text{ and } y_{jp} = 1 
\end{cases} \]

\[s_{jp} = \text{perceptron summation.}\]

\[p_i = \text{input signals in the perceptron network, 1 or 0.}\]

\[d_{jp} = \text{perceptron desired output.}\]

\[y_{jp} = \text{output signals of the perceptron network.}\]

\[w_{ji}^{\text{new},p} = \text{perceptron new weight after adjustment.}\]

\[w_{ji}^{\text{old},p} = \text{perceptron old weight before adjustment.}\]

\[\alpha_p = \text{perceptron learning rate, usually between 0 and 1.}\]

The activation function used in the perceptron is linear. This caused a set back in the development of the neural computational technology and discouraged several researchers including Minsky and Papert [60].

### 3.8. ADALINE AND MADALINE NETWORKS

The **ADaptive LINear Element (ADALINE)** network, invented by Widrow and Hoff in 1960 [63], resembles a single biological nerve cell with two or three layers of processing elements and a single output processing element. Their contribution is considered to be one of the most important in neural network technology and was largely due to their engineering background.

The input processing elements can take on values of plus or minus one with a bias which always takes the value of one. The input data is adjusted by weights, which are initialised with random values, and the summation of the input data with their weights produces a linear output. The learning algorithm is called the Widrow
Hoff delta rule or Least Mean Squares (LMS), and is an improvement on the perceptron as it is faster and more accurate. Initially, the weights are set to random values to prevent the adaptation procedure from trap in non-global minimum. This network has been applied in the real world for example in the telecommunication industry, particularly in the adaptive noise reduction of telephone lines.

The disadvantage of this algorithm is that it uses linear pattern separation. Figure 3.5 shows an Adaline network with input signals of (+1 or -1), where the output resulted in (+1 or -1).

An extension of the Adaline is the MADALINE. It is stands for Multiple ADAptive LInear Element (MADALINE). It is a multiple of Adaline networks in parallel. The Madaline network employs a majority vote rule on the output of the Adaline network. If the Adaline output plus one is more than one half, then the Madaline result is plus one. If on the other hand, more of the adaline output is minus one, the Madaline output result is minus one.
The network is fully connected when the input signals are connected to each Adaline network with their weights. There are no weight adjustments between the Adaline network and the Madaline network. The training of the Madaline networks begins with input data presented as input signals with their weights and a desired output of +1 or -1 at the Madaline output. During the learning process the Madaline network compares its output with the desired output. If the desired output matches the network output, this indicates that there is no learning in process. If they differ, the output close to zero in the wrong direction is used for adaptation. The algorithm guarantees convergence where the weights set of adjustment exists and has the ability to classify the input set of signals.

3.8.1. ADALINE Rule and Training

The ADALINE network performs as a summation device and the weights are adjusted on the inputs as:

\[ s_a = \sum_{i=0}^{a} a_i w_{ji} \]  \hspace{1cm} (3.4)

where:

\( a_0 \) = ADALINE bias of value equal to 1.

\( a_i \) = ADALINE input processing element of \( i \).

\( w_{ji} \) = ADALINE weight adjusted from input \( i \).

\( s_a \) = ADALINE summation.

\( y_{ja} \) = ADALINE network output.

Then the Adaline performs a linear threshold which is;

\[ y_{ja} = \begin{cases} +1 & \text{if } s_a \geq 0 \\ -1 & \text{if } s_a < 0 \end{cases} \]  \hspace{1cm} (3.5)
Usually, the output of the ADALINE network is either (+1 or -1), and the delta rule adjust performs the following;

\[ \Delta w_{ji}^a = \eta a_i (d_{ja} - y_{ja}) \]  

(3.6)

where:

\[ \Delta w_{ji}^a = \text{Widrow-Hoff delta rule.} \]

\[ \eta = \text{learning rate or learning coefficient. Usually a constant value between 0 and 1 is used to modify the learning process.} \]

\[ d_{ja} = \text{ADALINE desired output.} \]

3.9. HOPFIELD NETWORK

This network was developed by Hopfield in 1982 [61] and renewed interest in research which had been halted for many years due to the Minsky and Parpert publications [60]. His work is considered to be the best, simplest and most powerful of these with a theoretical mathematical foundation which can be understood of any other neural networks' model. His work is essentially a collection of studies by others in which he gathers together several publications to create one detailed piece of work. Hopfield’s contribution concentrated on the analog neural network theories which were of particularly interest to scientists and physicists. His first attempt to make neural chip took place in 1984, and by 1987 the AT&T Bell laboratories had announced the development of a successful neural network chip based on his network.

The most important characteristic of the Hopfield network is the feed back mechanism from the output layer to the input layer. The process involves a set of weight coefficients which ensure that the processing elements in the Hopfield network do not connect to themselves, as shown in Figure 3.6. The feed back
mechanisms reconstruct the missing input data and enhance the output data. The Hopfield network is a nonlinear recurrent network that is fully interconnected and the activity usually flows around and within the network until equilibrium is reached. It applies the principles of the energy function to reach a stable state. The process involves the response of the processing elements to the input signal and the modification of the network until it reaches a stable energy minimum. The network learns throughout the supervised learning suggested by the Hebb learning algorithm.

![Hopfield Network Diagram](image)

**Figure 3.6. Hopfield Network.**

### 3.9.1. Learning and Energy Concept

According to Hopfield each processing element is a binary that can take only two binary state values of 1 and 0. Bipolar values of +1 and -1 have been used which provide satisfactory results and simplify the mathematical analysis procedure. The activation function employed is governed by a hard-limit function of threshold of +1
or -1. The input to the processing element is the weighted sum of its inputs calculated from:

\[ s_{jh} = \sum_{i=1}^{n} h_i w_{ji} \]  \hspace{1cm} (3.7)

This summation is evaluated and checked as to whether it is greater or smaller than zero. The output result is +1 if the summation is at least 0. This can be written as:

\[
\begin{bmatrix}
\text{if } s_{jh} \geq 0 & h_j = 1 \\
\text{if } s_{jh} < 0 & h_j = 0
\end{bmatrix}
\]  \hspace{1cm} (3.8)

where:

\[ s_{jh} = \text{summation of Hopfield processing element.} \]
\[ h_i = \text{Hopfield input processing element.} \]
\[ h_j = \text{Hopfield output processing element.} \]
\[ w_{ji}^h = \text{Hopfield weight adjustment.} \]

The Hopfield network model requires symmetry of the weights for the processing elements in order that the function energy exists. Therefore, the weight to processing element 'j' from processing 'i' must have the same values, hence:

\[ w_{ji}^h = w_{ij}^h \]  \hspace{1cm} (3.9)

The Hopfield network passes through a state of energy that can be defined as:

\[ E_h = -\frac{1}{2} \sum_{j} \sum_{i=1}^{n} w_{ji}^h h_j h_i \]  \hspace{1cm} (3.10)

The Hopfield network guarantees convergence irrespective of the network's state of condition by reducing the energy to a lower value. The main property of an
energy function is that it always decreases or stays constant. For a specific processing element, the energy can be rewritten with effect to a certain processing element. The energy for processing \( j \) can be defined as:

\[
E_j = -\frac{1}{2} h_j \sum_{i \neq j} w_{ji} h_i
\]  

(3.11)

During the updating of the processing element \( j \), the difference in energy \( E_j \) is evaluated as:

\[
\Delta E_j = E_j^{new} - E_j^{old}
\]  

(3.12)

Hence the difference in energy becomes:

\[
\Delta E_j = -\frac{1}{2} \Delta h_j \sum_i w_{ji} h_i
\]  

(3.13)

The difference in the processing element is evaluated from:

\[
\Delta h_j = h_j^{new} - h_j^{old}
\]  

(3.14)

The conditions for \( \Delta h_j \), can be obtained from the following conditions:

If the output processing element changes from 1 to 0, the state of \( \Delta h_j \) becomes:

\[
\Delta h_j = -1
\]  

(3.15)

And hence the input summation reduces to:

\[
\sum_i w_{ji} h_i < 0
\]  

(3.16)

Therefore the difference in energy reduces to:

\[
\Delta E_j < 0
\]  

(3.17)

When the output processing element changes from 0 to 1, the state of \( \Delta h_j \) becomes:
\[ \Delta h_j = 1 \]  
(3.18)

And the input summation becomes positive as:

\[ \sum_i w^h_i h_i \geq 0 \]  
(3.19)

And finally, the difference in energy becomes:

\[ \Delta E_j \leq 0 \]  
(3.20)

where:

\[ E_h = \text{Hopfield Energy.} \]

\[ \Delta E_j = \text{difference in Hopfield energy.} \]

Hence converge is reached for both conditions where the difference in energy is less than or equal to zero. Generally, the method seeks for local or global minimum to reach a stable condition.

It should be noted that this procedure is only applicable for a binary Hopfield network. In 1984 Hopfied derived a similar method for continuous networks which allows for processing elements to have continuous values.

3.10. BACKPROPAGATION ALGORITHM

3.10.1. General Overview

Neural networks or artificial neural nets are based on modern research in the field of neurophysiology, a study of the human nervous system and its biological brain neurons. These networks are essentially a computing system consisting of a large number of interconnected processing elements arranged in several layers to provide output signals as result of a serious of input signals. Neural networks are a valuable information processing technology that is more efficient and robust than
conventional programming. It is a learning mechanism which involves a process of automatic weight adjustment between the layers of the individual processing elements to ensure that a stability approximation of the outcome results. The learning process involves the initialisation of all weights to random values to ensure that the network will not memorise or be trapped in a local minimum.

Backpropagation networks are considered to be the most reliable and most applicable of all. One survey has shown that about 80% of all applications used backpropagation due to mathematical designed of learning complex nonlinear relationship even the input data is less precise or noisy. Backpropagation network has the ability to minimise the mean squared error by applying a gradient descent algorithm that follows the gradient error curve downward across all the input patterns. This is mathematically computed by taking the partial derivatives of the error with respect to the weights. A supervised learning technique is essential in the learning process due to the presence of input and output data which ensures the development of the internal model which describes the objective requirements. Figure 3.7 shows a simplified one processing element of the backpropagation network with its summation and activation functions. These play a major role in backpropagation algorithm training.
A backpropagation network always has an input layer, one or more hidden layers and one output layer. Each layer consists of processing elements and every processing element is connected to the preceding and next layer. The processing elements in the input layer represent a set of input data, while the processing elements in the output layer represent the desired output. The hidden processing elements represent feature detectors, or an approximation to complex non-linear mapping between the input and output layers. The backpropagation algorithm involves a forward activation start when a set of input patterns is presented to the network and a backward error activation which begins at the output layer when errors propagate through the intermediate layers toward the input layer. The process of forward and backward activation continues until the error is reduced to an acceptable level or runs for a specified time. Figure 3.8 shows a typical backpropagation network with three layers, each of which is connected to the processing elements in the next layer. More complicated backpropagation networks are available by using more than one hidden layer. This is occasionally essential in the complex problem of non-linearity.
3.11. BASIC MATHEMATICAL FORMULATION

The basic mathematical formulation of the backpropagation network requires each processing element to perform four main steps:

1. Input connections, which are analogous to the synapses, receive information from other processing elements or start with known input data.

2. Summation function which involves the activation of each processing element with its weight.

3. A threshold function, which is a process of converting the summation input activation data to an output activation data by specific function.

4. Output processing elements, which resemble the axon in the human brain, and result from the previous process.
Generally, the output to processing element $y_j$, determined by the weighted sum of the input signals, (a set of input signals $b_1, b_2, \ldots, b_p$ applied through a set of associated weights $w_{ji}^b, w_{j2}^b, w_{j3}^b \ldots w_{jl}^b$) becomes the input to the activation function. Mathematically:

$$s_j = \sum_{i=1}^{n} b_i w_{ji}^b \quad (3.21)$$

$$y_j = f(s_j) \quad (3.22)$$

where:

- $b_i$ = input signals for the backpropagation network.
- $y_j$ = output signals of the backpropagation network.
- $d_{jb}$ = desired signals of the backpropagation network.
- $w_{ji}^b$ = backpropagation weight adjustment between the input and output signals.

The activation functions are non-linear, although there are several functions. For sigmoid activation function, equation (3.22), becomes:

$$y_j = f(s_j) = \frac{1}{1 + e^{-s_j}} \quad (3.23)$$

And for hyperbolic activation tangent function, equation (3.22) becomes:

$$y_j = f(s_j) = \frac{e^{s_j} - e^{-s_j}}{e^{s_j} + e^{-s_j}} \quad (3.24)$$

### 3.11.1. Output Layer Weights Adjustment

The network output layer produces a single real number for each processing element, ($y_{1k}, y_{2k}, \ldots, y_{qk}$). These real numbers are then compared to the desired output ($d_{1k}, d_{2k}, \ldots, d_{qk}$) and specified by the input training set to obtain the error signal. The
The error signal, $e_{qk}$, is a measure of the network performance for one processing element in the output layer, and is computed as:

$$e_{qk} = (d_{qk} - y_{qk}) \quad (3.25)$$

where:

$q = \text{backpropagation processing element in the output layer.}$

$k = \text{refer to the output layer in backpropagation.}$

$e_{qk} = \text{backpropagation error signal for one processing element in the output layer.}$

$d_{qk} = \text{backpropagation desired output.}$

$y_{qk} = \text{backpropagation actual output for one processing element.}$

We then multiply the error signal, $e_{qk}$, obtained from equation (3.25), by an activation function derivative to obtain the error value, $\delta_{qk}$, which can be computed from:

$$\delta_{qk} = f'_k(s_{qk})e_{qk} \quad (3.26)$$

The error value, $\delta_{qk}$, obtained from equation (3.26), is then multiply by $y_{pj}$, the output of one processing element in the hidden layer, to provide the connection weight adjustment, $\Delta w_{qp,k}^b$ (known as the generalised delta rule). This weight adjustment is computed as:

$$\Delta w_{qp,k}^b = \eta f'_k(s_{qk})e_{qk} y_{pj} \quad (3.27)$$

where:
\[ \Delta w^b_{q,p,k} = \text{backpropagation adjusted weight between the } q^{th} \text{ processing element in the output layer and the } p^{th} \text{ processing element in the hidden layer.} \]

\[ y_{pj} = \text{backpropagation output for one processing element in the hidden layer.} \]

### 3.11.2. Hidden Layer Weight Adjustment

The weight adjustments for the hidden layers require a different procedure because of the absence of desired outputs in the hidden layer. The error value, \( \delta_{pj} \), for the hidden layer is generated without the desired outputs. From equation (3.26), calculate each processing element error value in the output layer, \( \delta_{q,k} \). These are used to adjust weights going into the output layer where they propagate to the hidden layer to generate \( \delta_{pj} \) for the hidden layer which is computed as:

\[
\delta_{pj} = f'_j(s_{pj})(\sum_{j=1}^{n} w^b_{kj}\delta_{qk})
\]  

(3.28)

Similarly, the hidden layer is adjusted by:

\[
\Delta w^b_{po,j} = \eta f'_j(s_{pj})(\sum_{j=1}^{n} w^b_{kj}\delta_{qk})y_{oi}
\]  

(3.29)

where:

\[ \Delta w^b_{po,j} = \text{backpropagation adjusted weight between the } p^{th} \text{ processing element in hidden layer and the } o^{th} \text{ element in the input layer.} \]

\[ y_{oi} = \text{backpropagation output for one processing element in the input layer.} \]

\[ i = \text{refer to input layer in backpropagation.} \]
j = refer to hidden layer in backpropagation.

o = backpropagation processing element in input layer.

3.12. KOHONEN'S SELF-ORGANISING NETWORK

Kohonen's self-organising network, usually known as Self-Organising Map (SOM), was developed by Kohonen between 1979 and 1982 [64]. It is one of the most important network architectures, first developed in the early 1960's, between the Adaline and Hopfield networks, but is more biologically oriented. A key characteristic of this network is its ability to modify the internal state of the network to detect features found in the input training data without any external guidance. The network sorts out similarity items into appropriate categories by creating a two-dimensional flat grid feature detector map of the input data pattern. The role of Kohonen's self-organising network is to look for regularities or trends in the input data, learn according to the function of the network, and provide a visual graphical representation of its findings.

3.12.1. Network Structure

The Kohonen basic network has two layers. The first layer is the input layer which is fully connected to a two-dimensional Kohonen layer, known as the competitive layer. This layer can organise a relationship among patterns in the input data without any external influence. The SOM searches for regularities or trends and makes adaptations according to the specific function of the network. The SOM is viewed as a topological map which starts from random to reflect the distribution of the input training data.

The most obvious feature of the network is that the processing elements are not arranged in layers as in the backpropagation network, but in a flat two-dimensional grid. The network was specifically biologically motivated because of the absence of
desired output, whereby the network organises relationships among the input training data. The input and competitive layers are highly interconnected as each input processing element is connected to all the processing elements in the second layer, as shown in Figure 3.9.

3.12.2. Kohonen Training Rule

The learning algorithm is unsupervised so that only input data is applied during training and the weights adjusted so that given processing elements or sets of processing elements in the second layer are activated. One main characteristic of the SOM is that it determines the number of dimensions according to the dimension of input data, so that a two-dimensional problem is a result of two input data and n-dimensional problem is a product of n-input data. In general, the number of network dimensions is influenced by the number of input data. Each processing element computes the product of its weight with the input. Only the processing element with the largest product will produce an output and only that processing element and its
neighbours will adjust their weights. The basic philosophy behind the algorithm is that the processing elements compete for the chance of learning. The processing element with the largest output is declared the winner. The element will have the ability to inhibit its competitors and excite its neighbours by adjusting their weights. The weight of the winner processing is adjusted according to the Kohonen learning rule which makes the adjustment proportional to the difference between the input training data and the value of the pre-adjusted weights. Generally, the winning weight set is closest to the input set. The weights between the input and output processing elements are initially set to small random values prior to starting the learning process. Then the input data is presented to the network.

The Kohonen learning rule involving the input Kohonen processing element is represented as:

$$ K = [k_1, k_2, ..., k_n] $$

(3.30)

The processing elements are connected to the two-dimensional grid layer through weights which are identified by:

$$ W_i^k = [w_{i1}^k, w_{i2}^k, ..., w_{im}^k] $$

(3.31)

The procedure starts in the competitor layer when each of the processing elements in the Kohonen layer measures the *Euclidean* distance of its weights to the incoming input data. The *Euclidean* distance is an operation of measuring the nearest neighbour distance. This procedure involves the calculation of the individual processing element in the competitive layer whose weight is closest to the input data. This processing element will then be declared the winner in the competition since it provides the minimum distance. In the event of two processing elements have the same values, the lowest index is declared the winner. For input data with 'k_r' values and 'w_m' weights the *Euclidean* distance is computed as:
\[ D^k_i = \| K - W^k_i \| \]  
\[ (3.32) \]

\[ D^k_i = \sqrt{\sum_j (k_j - w^k_j)^2} \]  
\[ (3.33) \]

\[ D^k_i = \sqrt{(k_1 - w^k_1)^2 + (k_2 - w^k_2)^2 + \ldots + (k_n - w^k_n)^2} \]  
\[ (3.34) \]

The best value which wins the competition can be found from:

\[ \| K - W^k_{\text{win}} \| = \text{Minimum} (\| K - W^k_i \|) \]  
\[ (3.35) \]

By declaring the winner the next step is to search for the nearest neighbour processing element which is closest to the winner processing element. Then, a process of updating the weights commences, as evaluated from:

\[ w^{\text{new},k}_{ij} = w^{\text{old},k}_{ij} + \Delta w^k_{ij} \]  
\[ (3.36) \]

\[ \Delta w^k_{ij} = \begin{bmatrix} \alpha_k (k_j - w^k_{ij}) \\ 0 \end{bmatrix} \]  
\[ (3.37) \]

Equation (3.37) states that the updating is of equal certain value when the processing element 'i' is in the neighbourhood of a set of processing elements, otherwise it is equal to zero. The learning constant, \( \alpha_k \), in the Kohonen feature detector two-dimensional map varies as the process of updating evolves. Usually, it starts with a large value and decreases through the iteration process. Initial values are set to between 0.2 to 0.5 and decrease linearly to a certain value defined by:

\[ \alpha^k_i = \alpha^i_{\text{init}} \left(1 - \frac{t^k}{T_{\text{total}}} \right) \]  
\[ (3.38) \]

where:

\( \alpha_k \) = Kohonen learning constant.

\( w^{\text{new},k}_{ij} \) = Kohonen new weight adjustment.
\[ w_{ij}^{old,k} = \text{Kohonen old weight adjustment.} \]

\[ D_i^k = \text{Euclidean distance.} \]

\[ \Delta w_{ij}^k = \text{Kohonen difference in updating weights.} \]

\[ K = \text{vector of Kohonen input data.} \]

\[ W_i^k = \text{Kohonen weights vector.} \]

\[ W_{win}^k = \text{Kohonen winner weights vector.} \]

\[ \alpha_{t_c}^k = \text{Kohonen an acceptable decrease learning rate of constant.} \]

\[ \alpha_{ini}^k = \text{Kohonen learning constant that decreased to reach zero.} \]

\[ T_{total}^k = \text{Kohonen total number of training carried out.} \]

\[ t_{c}^k = \text{Kohonen the iteration of current training.} \]

Figure 3.10 illustrates the concept of a neighbourhood that is close to the processing element that wins the competition.

![Figure 3.10. Concept of Neighbourhood.](image-url)
The neighbourhood size, $S^k$, around the winning element is important for the progress of the learning iteration. The size of the neighbourhood is the processing elements around the winner within a square shape. Figure 3.10 shows three types of neighbourhood sizes and the calculation starts from the element nearest to the winner. The distance from the winning processing element and the neighbourhood square is determined from:

Along the horizontal:

$$\begin{align*}
    h_k &> k_{\text{win}} - N_d^k \\
    h_k &< k_{\text{win}} + N_d^k
\end{align*}$$

(3.39)

Along the vertical:

$$\begin{align*}
    v_k &> k_{\text{win}} - N_d^k \\
    v_k &< k_{\text{win}} + N_d^k
\end{align*}$$

(3.40)

As the progress of training iteration evolves, the neighbourhood distance decreases to certain values satisfied by the following equation:

$$N_d^k = N_{d0}^k (1 - \frac{i^k}{T_{\text{max}}})$$

(3.41)

where:

- $S^k$ = Kohonen neighbourhood size.
- $k_{\text{win}}$ = Kohonen winning processing element.
- $h_k$ = Kohonen neighbourhood horizontal distance.
- $v_k$ = Kohonen neighbourhood vertical distance.
- $N_d^k$ = Kohonen neighbourhood distance.
- $N_{d0}^k$ = Kohonen initial neighbourhood distance.
An area of application essential to a Kohonen network is probabilistic or statistical mapping where the network has the capability of mapping the function of the input signals.

3.13. NEURAL NETWORKS DOMAINS APPLICATION

Advances in software and hardware over the past five years have led to an open market of about $20 millions in 1988 and about $1 billion forecast for 1997. There are more than 300 vendors around the world trying to gain a stake in the market and this will increase in the coming years. The growth of interest in neural networks during the 1980s has been one of the fastest in the history of science. Commercial neural network applications have appeared in many disciplines, and show a great deal of promise in areas where problems are encountered such as in structural design.

Neural networks are one of the most rapidly expanding areas of recent research, attracting a wide variety of disciplines and generating much interest among engineers and scientists. Neural network applications cover a diverse spectrum of interest due to the special learning capabilities which make them applicable to many areas including robotic and control, real-time control, forecasting, classification problems, decision and support, optimisation including structural optimisation, processing monitoring, non-destructive testing, planning and scheduling, and image processing. Research is being conducted in many countries including U.K., Japan, and the U.S.A.

Research in the U.K. is closely linked with activity in other European countries. Several governments and private agencies have launched and support research using neural computing. The Department of Trade and Industry (DTI) has been aware of neural network development and encouraged British Industry to benefit from developments in this field. They launched a campaign which has three primary objectives [65]: 
1. National programme of workshops and seminars

2. Exhibitions programme

3. Campaign information

Private companies which have shown interest in neural networks include British Telecommunication (BT) who have launched a five year project under the name COONEX which focuses on the technology of image processing. Other companies who have launched similar projects include British Aerospace, Atomic Energy Authority as well as insurance companies such as Sun Alliance. A survey conducted by the Advisory COuncil for Science and Technology (ACOST) in the U.K. gave details of the expected growth of the neural network market over the next ten years. These are given in Table 3.1 [65].

Table 3.1. Neural Network Market [65].

<table>
<thead>
<tr>
<th>Year</th>
<th>Millions ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1988</td>
<td>20</td>
</tr>
<tr>
<td>1991</td>
<td>300</td>
</tr>
<tr>
<td>1997</td>
<td>1000</td>
</tr>
</tbody>
</table>

After launching their ten year project of the 'fifth generation' in 1980, by using the symbolic logic base rule, the Japanese launched another project, which can be referred to as the 'sixth generation' project, which bases their software and hardware on the concept of neural computing. The project named New Information Pprocessing (NIP) was launched in 1992 for a period of ten years using neural networks at a cost
about $150 million. Large Japanese companies and government laboratories intend to link their previous work of symbolic logic base with neural computing to discover rules used by an expert logic base system.

In the USA, considered to be the leader in neural network research and development, both government and private sectors play major roles in developments in this field. The Department of Defence provides a huge amount of financial support to promote the development of neural networks. The Defence Advanced Research Project Agency (DARPA) was funded by $390 million over eight years to study the strategic important of neural networks in the defence industry by seeking alternative methods to traditional computing approaches. The study was concentrated on two main aspects: the ability of neural networks to learn and the implementation of a massive parallelism algorithm. Also, Congress has approved a TNA project which detects plastic explosive in sealed luggage at international airports. A survey in the USA showed a strong research and industrial interest in this area, and several neural computing companies have participated in the development of both software and hardware. Table 3.2 gives details of the number of US companies involved in the design of neural networks [65].

Table 3.2. USA Companies Designing Neural Networks [65].

<table>
<thead>
<tr>
<th>Year</th>
<th>Millions ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1989</td>
<td>45</td>
</tr>
<tr>
<td>1990</td>
<td>65</td>
</tr>
<tr>
<td>1991</td>
<td>135</td>
</tr>
</tbody>
</table>
The surge in neural network algorithms and the improvement in the network's algorithms prompted many companies to develop commercially several types of neural networks architecture to solve many problems that are encountered in the industrial and research communities. Neural network development tools are designed on two main platforms, namely software simulators and hardware accelerators.

3.14. NEURAL NETWORKS IN CIVIL ENGINEERING

A considerable interest in the application of neural networks to civil engineering has emerged in recent years, due to the development of non-linear mapping. Generally, the linear function provides a primitive solution for the neural network to learn, but the strength of neural network lies in the non-linearity function that provides an internal representation of the input data.

Artificial neural network architecture has captured the engineering design expertise and made it available to designers. Engineering design requires intuition and human expertise which are not easily captured by computer. The application of neural networks covers wide areas of civil engineering, particularly structural engineering. Structural analysis is a potential area for the use of neural networks to compute structural displacements according to specific loading conditions. The network can be trained to map load-displacement data which results in explicit relationships for computing a stiffness matrix through the learning process.

Hajela and Berke [66] have extensively used neural networks in several different applications such as the analysis and design of truss structures through the use of the minimum weight design concept, the decomposition in optimal structural synthesis, and in structural mechanics. Structural optimisation solves for continuous, pure-discrete, or mixed continuous-discrete design variable problems. The necessary data
is computed by the traditional optimisation problem and usually involves a minimum weights structure for optimum profit.

In Structural Engineering, Vanluchene and Sun [67] have demonstrated the potential use of the backpropagation learning algorithm in three structural problems which include a pattern recognition, a simple concrete beam and a rectangular plate analysis. Their program was written in FORTRAN language and it was demonstrated that a neural network was capable of simulating the knowledge used by a professional Civil Engineer.

Preliminary structural design can also employ neural networks. Swift and Batill [68] have a configurational design for five bar truss structures of minimum weight, also a configurational design for ten bar truss structures of minimum design. The input data is obtained from the analysis of fully stressed design problems, as encountered in the normal structural optimisation situation. Their work was concentrated on mapping a set of design information where the weight contours are plotted as a map contour obtained by a neural network and compared to a map computed by conventional methods. This work provided an insight into how important design space is considered to be in the selection of the design process.

Jadid and Fairbairn [69] have applied neural networks to structural analysis by implementing an adaptive finite element mesh generation. The work focuses on the application of the neural network technique to adaptive re-meshing of an idealised square shape and individual triangle by using triangular elements. The backpropagation learning algorithm was implemented by a supervised training technique which deals with the problem of re-meshing structural elements in a structural analysis. The main objective of the authors' study was to demonstrate how neural networks can be employed to re-mesh structural elements without using numerically intensive computations. One essential requirement of that approach was the selection of feasible and appropriate domain for generating training and test data.
It also demonstrated that the power of neural networks lies in its presentation of \textit{n-dimensional} space and the tracking of each individual characteristics in that separate space. The benefits of applying neural network are that it does not make prior assumptions about mathematical formulation or the algorithm involved, and bases its results on the analytical results that feed into it.

The technology of neural networks shows tremendous promise in an area where conventional programming and logic symbolic failed to provide adequate solutions. Neural networks can be of benefit in the fields of civil engineering and structural engineering in particular due to their valuable information processing technology which can be used to solve problems which are resistant to other computational technology.

Structural engineering design requires human intuition as well as expertise and this is difficult to compute by conventional methods. Usually, a design process involves the determination of certain parameters that the design will meet the requirements of a standard code of practice. The parameters encountered in design are complex and do not readily meet the standard exiting algorithm without further simplification. Consequently, the neural network approach provides an alternative to conventional methods and will produce benefits due to its ability to handle highly complex multidimensional space of the kind usually encountered in structural analysis and design.
4.1. INTRODUCTION

In this neurocomputer era classical computational methods are being superseded, whenever possible, by more economical intelligent methods which must be as efficient as possible. Developments in software simulators, hardware accelerators, neural chips and computer technology offer an opportunity to tackle complex structural engineering problems which were impossible or impractical just a few years ago. In recent years, a popular trend towards using neurocomputer applications has emerged in the engineering field which has resulted in most engineers moving away from traditional methods of analysis of structures to the slightly different approach made available by neural networks. The growing awareness of efficient neurocomputers and the positive development of neural network techniques have produced a greater understanding of the more complicated problems and provided an alternative method of analysing structures. It is too early to state whether neural networks will replace conventional algorithm methods. However experimentation in the approach to engineering problems is essential to discover the potential of its application.

The objective of this study is to compare available experimental results with those evaluated on the basis of neural network techniques. Thus, with a greater knowledge of current information technology, this research approach can be transformed into a more reliable and efficient method of simulating experimental
modelling. This chapter outlines a standard procedure that will be adopted throughout the coming chapters. A brief presentation of the software and hardware that has emerged in recent years is made and the criteria for selection of a neural network are discussed. The previous experimental work carried out by Nirjar [1] is described briefly to establish the foundation upon which this study is based. The role of a numerical computer paradigm specially related to FORTRAN-77 [70], Turbo C++ [71] and the new high-level programming environment in this research project is stressed. A neural network development tool and the primary implementation requirements are explained in detail to provide an understanding of the way the tool operates. It is intended to provide a forum for the next chapters and establish a basis for discussion and explanation of the tool application. Terminologies and concepts used exclusively by the tool are explained to provide a clear and consistent guide for understanding the neural network process. Computational methodology considerations such as the experimental modelling and analytical computations used extensively in the experimental investigations are outlined. The role of the numerical computations paradigm that are implemented throughout this study is discussed as well as its dependency on programming techniques. The role of neural network applications as a predictive method is stressed in relation to the stages required in the programming to ensure reliable results.

The current 'state of the art' of neural networks and computational technology is also presented in this chapter to indicate that a digital representation of information technology is referred to within the computer processor as 1 or 0. The main objective of this chapter and subsequent chapters is to present numerical computations, analytical manipulations, experimental observations, theoretical findings, practical experience generations and neural network technique manipulations in a new type of computational method. This will encourage wider use of the new technology in the future and reduce the burden of analysing complex mathematical relationships. To
gain optimum benefit, the most recent technology and most strategic directions must be examined in the engineering field.

4.2. HARDWARE AND SOFTWARE

Neurocomputers have emerged to represent a new class of computers which run neural networks efficiently. The architecture is different that of conventional machines as a parallelism technique is employed to manipulate data at a faster speed. There are two possible strategies for reducing the amount of training network: neural chips and neural network accelerators.

Neural chips provide the hardware implementation of neural net data structure, whereas neural network accelerators are a dedicated process to speed up calculation. Mark III is a general purpose neural computing system which has eight processors. Each can perform about 8,000 processing elements, and 65,000 processing elements which use partial hardware for neural simulation. Mark IV is an advancement of Mark III which can perform about 250,000 processing elements with tremendous connections of 5 million. ANZA, ANZA PLUS, PARALLON 2X and Delta II floating Point Processor are new on the market. Each has high speed engines for general digital signal processing of algorithms and for neural network simulations. However, this hardware is very expensive and as a result was not used in this research study.

Due to the successful implementation of neural networks in many fields numerous software packages have emerged and selecting the correct development tool for a specific task is difficult. Selecting the appropriate tool is dependent on the work intended and the software capability to reach acceptable results. There are many neural software packages available for use on the Mainframe and Workstations to Personal Computers (PCs). Reviews for the following packages were carried out:

- BrainMaker® from California Scientific Software.
NeuroShell™ from Ward Systems Group.

NeuralWorks® Professional and NeuralWorks® Explorer from NeuralWare Inc.

ExploreNet™ and KnowledgeNet™ from HNC Inc.

AINet™ from AI Ware Inc.

ANSim™ and ANSkit™ from SAIC of San Diego.

ModelWare™ from Teranet IA Inc.

NeuralWorks® Professional II/Plus version 5 [17] was selected since this product was dependent on the ability of the tool to deliver and provide the following:

1. Capability to generate several broad networks with flexibility and permit a customised network to be developed.

2. Access to 31 neural network paradigms (five types of backpropagation, ten kinds of predictive networks, seven associative networks, five classification networks, two conceptualisations, one filtering network, and one optimisation network).

3. Availability of several learning rules and the mechanism to load data files into the RAM memory. This significantly increases the I/O speed and reduces excessive use of the hard disk.

4. Ability to select several network paradigms with the option to choose several transfer functions. This has a tremendous influence on the network results and its operation.

5. Capability and flexibility to convert a trained network into a 'C' code for development and deploy a workable network. The code can be easily linked to a main 'C' code for future deployment.

6. Multi-platform hardware support. The tool supports and runs on several platforms, such as IBM® PC's and compatibles, Macintosh®, Sun® Workstations, NEC EWS/4800, Silicon Graphics® IRS workstations, HP® workstations,
VAX/VMS® and others. This type of common interface across platforms allows switching between platforms without the need to learn new interfaces.

7. A wide variety of diagnostic tools for monitoring the network performance as the network evolves.

4.3. REVIEW OF PREVIOUS EXPERIMENTAL MODEL

The behaviour of a beam-column joint was investigated experimentally and analytically and empirical relationships established by Nirjar [1]. The three-dimensional beam-column corner joint, commonly used in reinforced concrete building frames, provided guidance in the selection of the test specimen dimensions. The corner beam-column joint was subjected to a complex stress distribution due to the effect of biaxial bending forces. Figure 4.1 shows how a beam-column joint was assembled by Nirjar [1]. The beams selected where based on spans common in multi-storey building frames. These usually vary from 3.5 to 16 metres where the column height is varies only between 2.5 and 4.5 metres. A column height of 550 mm with square cross-section dimensions of 100x100 mm was selected. The beam dimensions were 75 mm wide by 125 mm in deep with a cantilever span of 350 mm and a loading position of 300 mm from the column face. The members were designed according to the 1972 CP110, 1972 [72] and checked by the ACI code 318-71 [73] and the first recommendation of the ACI-ASCE Committee 352-76 [2].
Figure 4.1. Corner Beam-Column Joint Details.
Experimental work was carried out to investigate the structural behaviour of cast-in-situ beam-column joints under static loading conditions. The study investigates the relationship between the behaviour of beam-column joints and geometrical shape, amount and size of steel the reinforcement, fixed beam and column cross-sectional dimensions and concrete strength. Tests were carried out on a total of 34 specimens under the following conditions:

1. **Column Loading**: load were applied axially to the column section consisting of 10, 20, 30, 40, 50, and 60 percent of the ultimate strength of the column where the ultimate strength is computed as:

   \[ P_u = 0.85 f_c (A_g - A_{sc}) + A_{sc} f_y \]  

   (4.1)

2. **Column longitudinal reinforcement**: the range of reinforcement percentages was selected to meet the provisions of ACI 318-71 [73]. This provision specifies the minimum reinforcement to be 1% and the maximum to be 6% for a ductile frame. The column reinforcement percentage varied as 4.53, 3.92, 3.14, 2.01 and 1.41 percent with the column tested at 10% and 50% of the ultimate column load.

3. **Lateral reinforcement in the joints**: lateral reinforcement is provided to hold the main beam reinforcement and to provide confinement of the concrete. The ratio of lateral reinforcement, \( r' \), in the joints varied as 0.53, 0.40, 0.35, 0.18, and zero percent in tests at 10% of the ultimate column load.

4. **Beam tensile reinforcement**: tensile reinforcement was provided in the beam to withstand the applied bending moments and reduce the stresses in the section. The beam tensile reinforcement, \( \rho' \), varied from 0.72, 1.28, 2.00, 2.55, and 2.99 percent for tests at 10% and 60% of the ultimate load condition. The tests were
conducted according to the ACI 318-71 code [73] in which the reinforcement ratio \(\rho_b\) for the balanced condition at ultimate load is expressed as:

\[
\rho_b = \frac{0.85\beta_1 f'_c}{f_y} \frac{87000}{87000 + f_y} \quad (4.2)
\]

where:

- \(r_l\) = lateral reinforcement ratio of steel.
- \(P_u\) = column ultimate strength, **Newtons**.
- \(f'_c\) = cylinder compressive strength of concrete, **N / mm\(^2\)**, (4300 psi \(\approx 30 \text{ N/mm}^2\)).
- \(A_g\) = gross area of cross-section, **mm\(^2\)**.
- \(A_{sc}\) = area of compression reinforcement in the column, **mm\(^2\)**.
- \(f_y\) = yield stress of compression reinforcement, **N / mm\(^2\)**.
- \(f_y\) = yield stress of the main reinforcement, **N / mm\(^2\)**.
- \(\beta_1\) = constant, varying as \(f'_c\), and equal to 0.85 for \(f'_c = 30 \text{ N/mm}^2\).
- \(\rho_b\) = reinforcement ratio of tension steel at balanced condition.

The reinforcement ratio limits specified by CP110 [72] between 0.25% and 4% were taken into consideration and the minimum reinforcement ratio of steel was specified by the ACI 318-71 [73] code as:

\[
\rho_{\text{min}} = \frac{200}{f_y} \quad (4.3)
\]

or for SI system:
\[ \rho_{\text{min}} = \frac{1.4}{f_y} \]  

(4.4)

where \( f_y \) is in \( N/mm^2 \).

5. Beam lateral reinforcement: is defined as the ratio of volume of stirrups to volume of bound concrete and varied between 0.005, 0.0074, 0.0148, 0.0167 and 0.0333 for tests at 10% of the ultimate load column.

6. Cylinder compressive strength of concrete: the concrete compressive strength was determined from tests on concrete cylinders and the strength varied between 20, 25, 30, 35, 40 and 45 \( N/mm^2 \).

A detailed description of the specimens tested are shown in Table 4.1 [1]. The specimens were grouped into eight series (NN, NM, NO, NP, NQ, NR, NS, NT). The grouping is based on the following conditions:

1. Series **NN** is based on variations of column loading conditions, from 10% to 60% of the ultimate column load.

2. Series **NM** is based on variations of column longitudinal reinforcement ratio, \( \rho_c \), tested at 10% of the ultimate load loading.

3. Series **NO** is based on variations of column longitudinal reinforcement ratio, \( \rho_c \), tested at 50% of the ultimate column loading.

4. Series **NP** is based on variations in the area of tensile reinforcement in the beams tested at 10% of the column ultimate load.

5. Series **NQ** had the same reinforcement variations as the **NP** series but tested at 60% of the column ultimate load.
6. Series **NR** involved variations in the area of transverse reinforcement in the column and joint tested at 10% of the ultimate load.

7. Series **NS** had variations in the area of lateral reinforcement in the beams using different spacing of stirrups.

8. Series **NT** covered five different concrete grades viz 20, 30, 35, 40 and 45 N/mm$^2$.

The concrete used in all the specimens except series 'NT', in which the concrete strength itself was a variable, had an average strength of 30 N/mm$^2$. Concrete strength was measured from nine standard cylinder tests, with one set of six cylinders tested on the same day as the test and a further three samples tested at 28 days. For all bar sizes in the main steel used in the test, the yield strength of the steel is 304 N/mm$^2$. The yield strength for the lateral reinforcement is 272 N/mm$^2$ for diameters of 4.5 mm and 242 N/mm$^2$ for diameters of 3 mm. All beams tested were under-reinforced and failed by yielding of steel.

The loading arrangements consisted of holding the column by a steel column shoe in order to simulate a pin connection. The beams were loaded using a hydraulic jack placed on a 3-ton load cell. A continuous measurement of load versus end deflection was monitored and recorded with strain gauges placed in the steel and concrete. It should be noted that the beam moment was resisted by column moments both above and below the joint. The loading was applied to the column in stages as a percentage of the column ultimate load, varying from 10% to a maximum of 60% of the column load. The loading was applied axially to the column to ensure that no additional moments were introduced. Each load was held by locking the hydraulic jack simultaneously and then the measurements were recorded. The loading on the beams was applied at a distance of 300 mm from the face of the column by connecting them to same hydraulic jack. Measurements were made of the applied
Table 4.1. Experimental Specimens Adopted by Nirjar [1].

<table>
<thead>
<tr>
<th>Series</th>
<th>Specimen</th>
<th>Concrete Strength</th>
<th>Column Load Level</th>
<th>$f_c$ N/mm$^2$ (%)</th>
<th>Longitudinal Column Reinforcement</th>
<th>Lateral Reinf. in Col. &amp; Joint</th>
<th>Tension Reinforcement in Beams</th>
<th>Lateral Reinforcement in Beams:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No. and Dia. of Bars (mm)</td>
<td>$\rho_c$ (%)</td>
<td>Diameter and Spacing of Stirrups (mm)</td>
<td>No. and Diameter of Bars (mm)</td>
<td>$\rho$ (%)</td>
</tr>
<tr>
<td>$NN_1$</td>
<td>30</td>
<td>10</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3 mm 75 mm</td>
</tr>
<tr>
<td>$NN_2$</td>
<td>30</td>
<td>20</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3 mm 75 mm</td>
</tr>
<tr>
<td>$NN_3$</td>
<td>30</td>
<td>30</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3 mm 75 mm</td>
</tr>
<tr>
<td>$NN_4$</td>
<td>30</td>
<td>40</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3 mm 75 mm</td>
</tr>
<tr>
<td>$NN_5$</td>
<td>30</td>
<td>50</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3 mm 75 mm</td>
</tr>
<tr>
<td></td>
<td>(NN_6)</td>
<td>30</td>
<td>60</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>NM</td>
<td>(NM_7)</td>
<td>30</td>
<td>10</td>
<td>4 12 mm</td>
<td>4.53</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>(NM_8)</td>
<td>30</td>
<td>10</td>
<td>5 10 mm</td>
<td>3.92</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>(NM_9)</td>
<td>30</td>
<td>10</td>
<td>4 10 mm</td>
<td>3.14</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>(NM_{10})</td>
<td>30</td>
<td>10</td>
<td>4 8 mm</td>
<td>2.01</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td>NO</td>
<td>(NO_{11})</td>
<td>30</td>
<td>50</td>
<td>4 12 mm</td>
<td>4.53</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>(NO_{12})</td>
<td>30</td>
<td>50</td>
<td>5 10 mm</td>
<td>3.92</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>(NO_{13})</td>
<td>30</td>
<td>50</td>
<td>4 10 mm</td>
<td>3.14</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>(NO_{14})</td>
<td>30</td>
<td>50</td>
<td>4 8 mm</td>
<td>2.01</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>(NP_{15})</td>
<td>30</td>
<td>10</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 6 mm</td>
<td>0.72</td>
</tr>
</tbody>
</table>
Table 4.1. (continued)

<table>
<thead>
<tr>
<th>NP</th>
<th>NP_{16}</th>
<th>30</th>
<th>10</th>
<th>5 6 mm</th>
<th>1.41</th>
<th>4.5 mm 60 mm</th>
<th>2 10 mm</th>
<th>2.00</th>
<th>3 mm 75 mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP_{17}</td>
<td>30</td>
<td>10</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>4 8 mm</td>
<td>2.55</td>
<td>4.5 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>NP_{18}</td>
<td>30</td>
<td>10</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>3 10 mm</td>
<td>2.99</td>
<td>4.5 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>NQ</td>
<td>NQ_{19}</td>
<td>30</td>
<td>60</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 6 mm</td>
<td>0.72</td>
<td>3.0 mm 75 mm</td>
</tr>
<tr>
<td>NQ_{20}</td>
<td>30</td>
<td>60</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 10 mm</td>
<td>2.00</td>
<td>3.0 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>NQ_{21}</td>
<td>30</td>
<td>60</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>4 8 mm</td>
<td>2.55</td>
<td>4.5 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>NQ_{22}</td>
<td>30</td>
<td>60</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>3 10 mm</td>
<td>2.99</td>
<td>4.5 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>NR</td>
<td>NR_{23}</td>
<td>30</td>
<td>10</td>
<td>4 10 mm</td>
<td>3.41</td>
<td>4.5 mm 80 mm</td>
<td>2 12 mm</td>
<td>2.99</td>
<td>4.5 mm 75 mm</td>
</tr>
<tr>
<td>NR_{24}</td>
<td>30</td>
<td>10</td>
<td>4 10 mm</td>
<td>3.41</td>
<td>3.0 mm 40 mm</td>
<td>2 12 mm</td>
<td>2.99</td>
<td>4.5 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>NR_{25}</td>
<td>30</td>
<td>10</td>
<td>4 10 mm</td>
<td>3.41</td>
<td>4.5 mm 80 mm</td>
<td>2 12 mm</td>
<td>2.99</td>
<td>4.5 mm 75 mm</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.1. (continued)

<table>
<thead>
<tr>
<th></th>
<th>( NR_{26} )</th>
<th>( NS_{27} )</th>
<th>( NS_{28} )</th>
<th>( NS_{29} )</th>
<th>( NS_{30} )</th>
<th>( NT_{31} )</th>
<th>( NT_{32} )</th>
<th>( NT_{33} )</th>
<th>( NT_{34} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>10</td>
<td>4 10 mm</td>
<td>3.41</td>
<td>2 12 mm</td>
<td>2.99</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
</tr>
<tr>
<td>( NS )</td>
<td>6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3.0 mm 50 mm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( NS )</td>
<td>6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3.0 mm 25 mm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( NS )</td>
<td>6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>4.5 mm 50 mm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( NS )</td>
<td>6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>4.5 mm 25 mm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( NT )</td>
<td>50</td>
<td>50</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3.0 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>( NT )</td>
<td>50</td>
<td>50</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3.0 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>( NT )</td>
<td>50</td>
<td>50</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3.0 mm 75 mm</td>
<td></td>
</tr>
<tr>
<td>( NT )</td>
<td>50</td>
<td>50</td>
<td>5 6 mm</td>
<td>1.41</td>
<td>4.5 mm 60 mm</td>
<td>2 8 mm</td>
<td>1.28</td>
<td>3.0 mm 75 mm</td>
<td></td>
</tr>
</tbody>
</table>
axial loads, beam deflections, reinforcement bar strains and concrete strains on all specimens. The measurement techniques were kept the same in each test to provide consistent results.

4.4. NUMERICAL COMPUTATION PARADIGM

During the past decade, there has been extensive progress in the mathematical modelling of both linear and non-linear structural behaviour. The task and achievement of these models are expressed in terms of their ability to imitate real structural behaviour. The main obstacle against the implementation of such models is the requirement of vast computer resources, even for small structures.

Numerical computations involve the translation of the experimental model into mathematical modelling through procedural low-level programming languages (BASIC, C, COBOL and FORTRAN) that required intensive programming and effort in writing and debugging. It is an approach that applies a fixed sequence problem solving task which takes the form of algorithms, with a pre-defined structure of sequential steps, looping structure and branching.

Usually, convergence problems arise in engineering problems due to the nature of non-linearity and this results in a substantial amount of time in fixing the problem to optimise the process to achieve reasonable convergence. The feedback mechanism from experiment results might also lead to revisions of the mathematical model which eventually required re-programming and re-implementation.

The current state of the art in modelling a beam-column joint mathematically is still at very low-level programming and clearly high-level environment tools will minimise most of the low-level programming and allow the engineer to spend his time on the real aspects of the problem. Ideal high-level programming tool environments are on-going for several years which will allow substantial reductions in research time if it is used efficiently. MathCad® 4 [16] is a high-level science and
engineering tool environment which incorporates SmartMath features which makes numeric and symbolic calculations faster and more accurate. SmartMath provides an intelligent interface between the user's problem and the Mathcad's numerical and symbolic capabilities. The role of SmartMath reviews the input and executes a strategy for solving the problem by writing a new set of equations in a popup window. Another important feature of SmartMath is that it comes with symbolic/numerical optimisation and live symbolic modules. This type of environment is very flexible and can probably succeed where low-level programming cannot obtain easily results.

The research work used FORTRAN-77 [70] programming and the Mathcad® 4 [16] tool to generate the appropriate data required by the neural network. The main features of using FORTRAN-77 [70] and Mathcad® 4 [16] are to produce results for selected applications and then pass them to the neural network. Turbo C++ [71] by Borland International Inc., was also used to create the main source code and can be easily interfaced with the FlashCode which is an integrated feature provided by the NeuralWorks® professional [17].

4.5. NEURAL NETWORKS DEVELOPMENT TOOLS

NeuralWorks® Professional II/Plus version 5 [17] is a development tool for neural network implementation on a microcomputer. It is a complete, comprehensive and sophisticated graphics software package with multi-paradigm prototyping for system development. It employs one of the most popular and powerful algorithms to perform parallel distributed processing. It can be used to design, build, train, test and deploy a neural network in many forms for approximating complex non-linear mathematical relationships. The networks are displayed either as network or Hinton diagrams. NeuralWorks® [17] interface is a user-friendly graphical environment and is organised into three main areas: a mix of floating down menus used to control the network; network working area interface; and "fly-out" tool boxes of several icons.
4.5.1. Floating Down Menu

The floating down menus provide a window interface within the network working area that is associated with six main functions:

1. File Menu: provides commands to control the major file operations such as open, new, print, quit and save files. In addition, it can access other directories and save windows.

2. InstaNet Menu: the InstaNet builder provides a selection of 28 major paradigm that can be created with many variations. This menu provides a dialogue box with numerous choices that allows the building of a suitable network which makes the construction of the network extremely simple and visible. To create a backpropagation network or other networks, all you need to do is specify the number of inputs, hidden and output layers with the specific momentum, learning rate, transformation function, training file, and test file.

3. Input/Output (I/O) Menu: commands that affect the input and output data by setting up specific parameters for the data. The I/O parameters dialogue box gives choices on the way the data is presented either sequentially or randomly. The user has a choice either to direct input from file or keyboard. A 'C' code subroutine can be obtained through the FlashCode to deploy the full train network.

4. Instrument Menu: consists of three facilities for building Probe and Instruments combinations: Easy-Probe, InstaNet Probe and an instrument that can be built from scratch. The probe components are: PEs, weights and layers. Instruments extract the data from the probe and present the data in a graphical form. Probes can be attached to selected processing elements or a layer which allows the user to view what is actually happening within the network. A probe is a means of pointing to groups of PEs or connections to extract values from them later. The instruments and probes are essential for understanding and debugging the
network during its training since it provides clear visual monitoring of the network as it evolves.

5. Run Menu: provides commands to run, test, refine and recall the network. The run command can be invoked by specifying the number of iterations that usually runs in thousands. The test command is invoked by specifying the test data to be executed only once and sequentially to ensure that all test data are read.

6. Utilities Menu: provides an access to the global parameters for editing. The dialogue box allows the networks name to be changed and gives a choice between the selection of either Auto-associate networks which require only input data or Hetero associative networks that require the input and desired data.

4.5.2. Working Area Interface

Within the working area interface the network is drawn either as Hinton or Network diagrams. Hinton diagrams display the network graphically as the interconnections matrix. Processing elements are arranged in the X-axis and Y-axis assuming the output processing elements are on the X-axis. The input processings are displayed along the Y-axis after the processing elements are multiplied by the weights. The network diagram is arranged by layers and connected by solid or broken lines. Layers are arranged from the bottom of screen starting from the input layer, one or more hidden layers and one output layer. The connections and the processing elements can be accessed by clicking off the specified processing element or layer where a dialogue box appears for specific changes.

4.5.3. Fly-Out Boxes

The 'fly-out' tool boxes are icons on the left side of the working area that manipulate the network topology. It consists of seven icon palettes to control the
layers and individual processing element. The palettes allow one to add, delete, clone or modify processing elements and connections between them.

4.6. NEURAL NETWORK PRIMARY REQUIREMENTS

The neural network modelling described for this research is implemented on a Personal Computer (PC) with an Intel® 486 Processor and a RAM of 16 MB. The operation of a neural network tool requires the setting up of training and test data for each individual task and correct parameters that the network requires to provide a reasonable and acceptable trained network. Figure 4.2 illustrates the methodology process of implementing a neural network, involving two main stages; **pre-processing** and **post-processing** which require the following strategies:

4.6.1. Generation of Data

The process consists of collecting the required data in one place by generating a FORTRAN algorithm or implementing Mathcad® 4 [16] for each specific task. Then separate the data into two sets, one for training and the other for testing. The testing data is normally taken as 10% of the training data such that the 10th element of each training set is reserved for the testing data which will provide the best picture representations and increase the confidence in the performance of the trained network. Generally, the more training data the better the network will perform.

4.6.2. Scaling of Data

Transform the input data to acceptable values to the network. The network accepts only values from 0 to 1 for the sigmoidal function and -1 to 1 for the hyperbolic tangent function. The input and test sets are normalised by a tool within the neural networks under the MinMax table. The process involves the computation of low and high values of each training example data field in the selected data files and stores them in the MinMax table.
4.6.3. Selection of Network

A backpropagation network is a general purpose network that can be implemented for prediction, classification, system monitoring, filtering and solving other problems. The advantages of the backpropagation network are the use of non-linear regression techniques that attempt to minimise global error, its ability to provide compact distribution representations of complex data and its potential to manipulate multiple-dimensional functions. However, there are also disadvantages for the backpropagation network due to its slow learning, its weakness in solving fundamentally different problems and its difficulty to set good learning parameters. To overcome these short comings variants of backpropagation networks are available that can resolve the inadequacies in backpropagation. These variant networks are discussed in the neurodynamics section.

Three main aspects are essential in selecting a specific network paradigm that gives and dictates the characteristics of a given network. The selection of an appropriate network is based on the three following configurations: architecture, topology and neurodynamics. A simple neural network manual computation is shown in Appendix A.

4.6.3.1. Architecture

The architecture is essential in applying a neural network effectively it requires skills and knowledge of the network paradigm to meet the problem at hand. Another important skill is to know how to tune a network's performance which is an art learned only with practice. The backpropagation and its variant network architecture is selected for this research according to the availability of series pattern pairs, where each pair consists of an input pattern with desired output pattern. The learning technique offered by the backpropagation is supervised learning.

4.6.3.2. Topology
Network topology consists of the number of input and output layers, the processing elements (PE's) they contain, the number of hidden layers and processing elements, their interconnectivity and the properties of geometrical configurations. Usually, one hidden layer is selected for simple problems while more than one hidden layer is selected for complex problems. Within each hidden layer, the fewer the number of processing elements (PE's) the better the network performs. As a general rule, the amount of input data that can be used as an upper bound for the number of PE's in the hidden layer is as follows:

\[
h_{PE} = \frac{N_{row}}{R^* (o_{PE} + i_{PE})}
\]

(4.5)

where:

\( h_{PE} \) = upper bound for the number of processing elements in the hidden layer.

\( N_{row} \) = number of rows in training data.

\( o_{PE} \) = number of processing elements in the output layer.

\( i_{PE} \) = number of processing elements in the input data.

\( R \) = range between 5 and 10.

4.6.3.3. Neurodynamics

Neurodynamics represents the learning rules and transfer activation functions that represent a specific network. The Generalised-Delta-Rule developed by Rumelhart, Hinton and Williams [62] is the most popular learning rule used by backpropagation networks. Two popular extensions of the generalised delta rule are implemented in the NeuralWorks® [17] tool, viz the Cumulative-Delta-Rule which
accumulates weights changes over several examples and the Normalised Cumulative Delta Rule. Variants of the backpropagation networks include:

1. Delta-Bar-Delta rule (DBD); proposed by Jacobs [74] attempts to encourage the speed of convergence by a heuristic route and increases the rate of learning by reducing the learning time for the network. The past values of the gradient are used to infer the local curvature of the error surface.

2. Extended-Delta-Bar-Delta (EDBD); Minai and Williams [75] suggested a type of predictive rule as an enhancement of the Delta-Bar Delta (DBD). The EDBD implements a heuristic which encourages positive learning trends and reduces oscillation.

3. QuickProp and MaxProp: uses a quadratic estimation heuristic in determining direction and the step size and had developed at Carnegie Mellon University in Pittsburgh, USA by Scot Fahlman.

4. The Logicon Projection Network: can be deployed with other learning rules to speed up the actual learning by estimating the optimal direction and optimal step size for weight changes. It is also characterised by its architecture that allows a good initialisation which is substantially speeded up by the process of learning.

5. Fast Learning: introduced by Tarig Samad of Honeywell Inc., and is a modification of the Delta-Bar-Delta. A portion of the estimated error is added to the connection's source processing element of the activation value prior to the connection of the learning process.

The NeuralWorks® [17] tool incorporates five types of activation functions-viz. sigmoid, TanH (hyperbolic tangent), sine, linear, and Digital Neural Network Architecture (DNNA) functions. Selecting a transfer function is entirely determined by the kind of data and the aim of the network. The recommended activation functions are the sigmoid activation function, which is best for learning about
"average" behaviour, and the hyperbolic tangent activation function which is suited for learning about "deviation" from the average.

4.6.4. Training and Testing the Network

Training the network successfully requires many choices and training experiences. After selecting the correct network configuration, which has substantial impact on the network results, the basic training processing involves the presentation of the input data with the desired output. The network then adjusts its internal presentation by carrying out an iteration procedure for correcting the error to produce acceptable results. An iteration process continues until the network converges to acceptable levels, or runs for a specific time. Usually, the number of iterations is specified as the number of learnings in the run menu, or as an acceptable error in the RMS diagnosis tool. Once the network is trained and converges, the test set is presented to the network sequentially only once to increase the confidence of the network performance and accounts for accuracy. During the process of learning the network is monitored graphically and instantaneously by several tools provided by the software to observe the learning process and to adjust any configurations that might arise.

4.6.5. Network Performance

During training, the network performance is monitored by Root-Mean-Square (RMS), weight histogram, and confusion matrix diagnostic instruments provided by the tool to achieve a better understanding of the network performance. The RMS error instrument provides a measure of the output network performance over the number of training iterations. This instrument is created by selection within the network menu. The RMS error calculates each error signal in the output layer, adds them up, then divides these by the total number of processing elements to obtain the average value for the output layer. This results in calculating the square root of that average. Mathematically, the RMS error is computed from [76]:
\[ RMS = \sqrt{\frac{\sum_{pt} \sum_{opt} (d_{out} - y_{out})^2}{n_{pt} n_{opt}}} \]  \hspace{1cm} (4.6)

where:

\[ RMS \] = Root-Mean-Square.

\[ \sum_{pt} \] = summation over all patterns in the training set.

\[ \sum_{opt} \] = summation over all output processing elements.

\[ d_{out} \] = desired output.

\[ y_{out} \] = network output.

\[ n_{pt} \] = number of patterns in training set.

\[ n_{opt} \] = number of processing elements in the output layer.

The weight histogram instrument is used to monitor the network performance and is created by selection within the network menu. The weight histogram provides a normalised histogram of all the variables in the network that changes during the training session.

The confusion matrix is also used to monitor the performance of each network output processing element and compares it to the desired output. This instrument is selected within the backpropagation or InstaNet builder. The x-axis along the confusion matrix provides the network output and the y-axis is the desired output. The interior quadrants are discretised into bins to show the network outputs. A value of one means an excellent correlation between the desired and network output.

One of the main factors that control convergence is the epoch size. The epoch size is the number of data presentations over which weight changes in the network are accumulated. This value is set-up in the dialogue box of the I/O menu before the
network commences its learning. The *epoch size* can be tuned and adjusted to provide better learning procedures by monitoring as it evolves.

The research emphasised the variables used for individual application where appropriate.

### 4.6.6. Network Parameters Optimisation

Once the network is set and trained, there are several key areas where the network performance can be optimised by tuning the *hidden layer size*, *epoch size*, as well as *learning rates*.

- The *hidden layer size* can be adjusted either through a constructive or destructive approach. The constructive approach starts by building the network without the hidden layer where the input layer is connected directly to the output layer. The network weights are then trained until the error stabilises. Adjust these weights and add one hidden processing element and connect to the input and output layers. Resume training until the network was not make any mistakes on the training data. Continue adjusting and adding processing elements until the network performance was not improve, at which time the network starts to memorise the data. The network performance is plotted for each addition of the processing for both the training and test data.

The destructive approach starts with a number of the processing elements in the hidden layer. Train the network for a short time and then perform testing on both the training and test data. Disable one of the processing elements, train the network and test again on training and test data. If the performance of the network improves, then continue to disable another processing element and repeat the process.
- **Epoch size**: adjustments of the epoch size involve selection of an initial value and training the network for a specific number of iterations. Test the network and record the accuracy (the number along the y-axis of the confusion matrix which is known as Pearson's R-coefficients). Repeat the process four or five times and plot the accuracy against the epoch size. The peak value of the epoch size is selected for training the network and represents the frequency of certain dominant components of the underlying noise.

- **Learning rates**: are coefficients which determine the amount of change when the network adjusts the interprocessing elements connecting strength. Manipulating the learning rates depends entirely on the network connectivity. If the connection does not jump layers, then the learning rates for the first hidden layer is twice that of the next hidden layer. If the connection does jump layers, then the learning rates should be reduced by at least 30% of the next hidden layer. Usually, the desired learning rates are which result in smooth RMS error graphs and weight histograms for those each layer. If the RMS error graph jumps, then reduce the learning rates proportionally.

4.6.7. **Deployment of Trained Network**

Once the network is completely trained and tested, the network can be deployed as part of the system application. This can be accomplished by either converting the trained network into a 'C' subroutine provided by the FlashCode and linking it with a main 'C' source code or interactively by entering data through the keyboard and getting results instantaneously. However, for a large number of input and test sets, the interactive is not recommended due to its prolonging of entering data and the possibility of entering the wrong data.

4.6.8. **Monitoring and Maintenance of Trained Networks**

Trained and deployed networks require continuous monitoring and maintenance to check reliability. Developing and debugging the trained, testing and creating
Figure 4.2. Neural Network Strategies Implementation.
maintenance are part of the network development. To conclude, neural network task allows computers to perform tasks that would otherwise require continuous human input and attention.

4.7. COMPUTATIONAL METHODOLOGY CONSIDERATIONS

Real world problems require solutions using different approaches that really depend on the problem at hand and require the most efficient procedure that can achieve reliable solutions. Intensive experimental and analytical studies have been undertaken over the years in several countries which has led to the establishment of reliable reinforced concrete structural analysis, improved design methodology and updated codes of practice. Theoretical analysis in recent years has also been modified and new equations with their relationships have been evaluated which lead to significant advances in the state of knowledge of structural behaviour and analysis.

Theoretical derivations of models of concrete structures are very complex due to the complexity of the structural shapes and the presence of different material properties. A reinforced concrete beam-column joint model addresses this complex behaviour of yielding of the ductile material, brittle fracture of the concrete, localisation of multi-stresses, micro-cracking and different loading conditions, all of which make numerical implementation of such a complex structure very difficult if not possible.

The problem and the different activities encountered when investigating experimental problems involve two main classes of activity: experimental modelling and analytical computation. During the experimental stage, models are built to describe the relevant state of the structure. This is achieved by building a simplified structure of selected material properties and geometrical shapes which is often time-consuming and very costly. On the other hand analytical models involve parameters that can be selected to give the best fit for given real data usually supplied from
experimental models. These analytical computations are adopted to establish a general procedure to derive several parameters for specified conditions and in such cases it is obvious that the neural network approach can be adopted to determine the missing parameters.

The objective of this study is to demonstrate a concept and methodology, rather than to build a full-scale knowledge-based system model, by incorporating most of the fundamental aspects of a neural network in solving the complex non-linear mapping for a beam-column joint. Generally, it may be possible to identify certain parameters and allow the neural network to develop the model and account for the observed behaviour without relying on a particular algorithm but depending entirely on the manipulation of numerical data.

4.8. APPLICATION OF NEURAL NETWORKS IN PREDICTION

Linear analysis in reinforced concrete structures is no longer considered adequate to obtain a realistic and safe design which must deal non-linear analysis and requires the development of an intensive computational algorithm for an acceptable proper solution. Civil engineering confronts massive complex mathematical computation in which parallelism can provide a great potential utilisation of neural network in civil engineering in general and structural engineering in particular. Recent developments in neural networks and computer technology have merged to incorporate advanced predictive capability solutions, which were not available before and simulate humans recognising similarities in patterns to reach suitable solutions. The power of the neural network predictively lies in the flexible and explicit data implementations rather than the representations of algorithmic procedure. The encoded trained network can automatically retrieve valuable information with less time and can be updated for future use without intensive numerical or analytical formulation.
Identifying the potential applications of neural networks in structural engineering is perhaps the most skilful and difficult task facing the engineer and requires correct procedures which is mixture of art and science. This research intends to significantly integrate the scope of symbolic logic base manipulations, numerical algorithm computations and neural network applicability in the assessment of reinforced beam-column joint behaviour. Emphasis is particularly concentrated on shifting away from the complicated numerical routines computation to the manipulation of complex relationships among data by neural network. It is an effective approach to capture the fundamental aspects of the patterns in the data and categorise it as a form of classification which results in feature extraction in a form recognisable by the neural network. Therefore, in order to enhance the efficiency and effectiveness of neural networks in the study of beam-column joints, the research work is formulated on the basis of three stages: preparatory, renew and final stages.

4.8.1. Preparatory Stage

Experimental results are of great importance in improving codes of practice in general and understanding in this case the behaviour of beam-column joints. The preparatory stage basically involves re-evaluating the previous experimental model for all 34 test specimens and also relying entirely on the analytical formation produced by Nirjar [1]. Prior to implementation of the neural network, intensive numerical programming was performed to generate the desired data required by the network. The experimental data was also used as an enhancement to the numerical data. The approach is that generating training and test data is organised according to intended requirements so that certain data are not presented to the network, thus ensuring the capability of the network to recognise and predict the missing desirable data.

The process starts to examine the experiments and analytical results by performing numerical computation for each series in the group. Having identified
the specific group for neural network prediction application, two data files are established as training data and test data. The next step is to select and set-up a specific network for certain tasks and begins training, adjusting, retraining and manipulating network parameters until a final acceptable trained network is obtained. The trained network is then deployed to predicate the missing series of that group by providing only the required input missing data and expecting that the network will provide results close to the desired output. The aim is to verify and observe the ability of the neural network in working out the missing series. For positive response identifications the results can provide considerable information and knowledge because of the capability of the network to predict a result for the series that is normally obtained through experimental procedure.

The problem is then set up to tackle more recent up-to-date research findings, experimental investigations, recent code of practice evaluations as well as new research publications. This enhancement of data will provide a guide for any future experimental work and ultimately reduce the number of specimens required to be tested. The reduction in experimental investigation is an objective to reduce costs and save time.

4.8.2. Renew Stage

The main features that are obtained from the preparatory stage including the investigation of the experiment results associated with the analytical formulation can be of great benefit for further development of new methodology. The general knowledge gained can now be utilised to obtain reliable relationships and function equations to suit a definite purpose. The research can benefit from previous information technology in all its forms and apply it in a constructive way to shape methodology that integrates it into its operation as follows:
• Acquisition of learning from experience and examples as applied in the framework of symbolic logic as described in chapter 2.

• Introducing new codes of practice particularly since 1977 (BS and ACI Codes of practice).

• Updating relevant new research work findings as published in conferences and seminars.

• Assessing recent research publications.

• Storing the new concepts for future work.

• Incorporating new concepts related to the investigations of beam-column joint analysis.

• Evaluation and verification of theoretical discovery and formulation of analytical work findings.

• Ability of handling complex mathematical relations due to the availability of a complex mathematical tool that provides instant and accurate results.

• Better understanding of this technique and assistance in carrying out further research.

• Collection of expertise from many specialists, by means of feedback mechanism which enables the tool to learn from experience and thereby reduces the number of iterations necessary in procedural language to achieve the desired objective.

• Potential of improving the final product as a result of previous information and storing the product for future use.

• Reduction in time to create structural modelling by learning from previous experience and by adjusting to new variations.

• Self-modifying operation according to the presented data.
Minimise paths can be eliminated by storing relevant information about specific information and predicting the solution that is likely to succeed in the current problem.

Useful application especially where the problem is an ill-problem or incomplete.

To permit some structural generalisation.

Considerable potential economic advantages which lie in the significant reduction of the number of specimens tested.

4.8.3. Final Stage

This stage is concerned with the application of neural networks for the new and updated information based on the recommendations of the Code of Practice. It also provides a way to enhance the results of data already obtained. Chapter 8 discusses this stage in detail.

The role of computers in the application of neural networks in previous experimental work and the formulation of new methodology concerning specific investigation is presented. The work is formulated on the basis of three criteria:

1. The feasibility of the methodology has been demonstrated in several examples and applications.

2. Prediction of several variables, equations and relationships through network manipulations which admits a direct monitoring of responses characterised as input data entered.

3. The reduction of the number of test specimens by evaluating fewer specimens and mapping the results to a predicted new situation. This is one of the most important areas for neural network application in terms of economic impact.

The approach undertaken by this study is to couple neural network techniques with conventional methods of analysis to produce a pattern recognition processing
system. It is an attempt to understand and interpret the behaviour of the beam-column joint which requires characteristic knowledge of all parameters, functions and variables that is vital prior to implementation of the neural network. The investigation was carried out in order to observe the ultimate flexural member representations in a new approach, the deformational behaviour by studying the moment curvature relationships and shear with strength and behaviour for beam-column joints for unbound and confined concrete.

Therefore, the research study focuses on the representation of a beam-column joint in the framework of neural network predictions in terms of the following activities:

- Ultimate flexural moments.
- Moment curvature relationships.
- Shear strength and behaviour.

The ultimate flexural moments include the full establishment of the stress-strain relationship prior and post the experiment work. Several stress-strain relationships have been proposed during the past decade which have a tremendous effect on the representation of the stress-block. The effect of variations in concrete strength has been discussed. The strain at maximum stress as well as the strain at ultimate is also presented in several methods, and this has an influence on the way the analysis is formulated. The modulus of elasticity has been presented by several equations in order to provide an understanding of the concrete behaviour. These parameters are very important in establishing the properties and relationships for analysing structures and have an affect on the results. The ultimate flexural strength of reinforced concrete is determined using experimental expressions and new expressions that have been derived since 1977.
The moment-curvature relationship is evaluated by studying the characteristics of the load-deformations and moment-rotations for the beam-column joint. The three main stages of analysis that the moment curvature passes through are investigated to include the cracking, yielding and the ultimate stages. The analysis is also extended to study the moment curvature relation for confined sections.

The nature of the stresses in the joint is very complex due to the biaxial loading and the compressive force on the column. The full and comprehensive review of beam column joints in the ACI-ASCE committee 352-76 [2] has been applied, followed by the updated recommendations in 1985 [3]. The investigation of beam-column joints is studied thoroughly with respect to changes in the Codes of practice, the advancement of research and the new surge of publications. Shear consideration is also part of the investigation that are undertaken both in the flexural members as well as the joints themselves.

The final neural network applications are an attempt to present a new concept of analysing the beam-column joint by providing alternative methodology which entirely depends on numerical data with intensive computational algorithm that benefit from previous experience, new research findings and recent theoretical formulations.
CHAPTER 5

ULTIMATE FLEXURAL STRENGTH OF REINFORCED CONCRETE MEMBERS

5.1. INTRODUCTION

Factors affecting the flexural strength of reinforced concrete members are examined in this chapter which deals with the evaluation of the compressive force and the ultimate moment of a cross-section using several methods. The strength of a reinforced concrete member depends on the strength of the materials from which it is made as well as the properties of the concrete and steel components. The ultimate strength of a cross-section is reached when the steel yields, followed by crushing of concrete. Adopting a strength method is based on the characteristics of the reinforced concrete section is inelastic behaviour at high loads, a non-linear stress-strain curve and an acceptable deformation of the section at larger loads. Several parameters were analysed in an attempt to reach a simplified method of analysis. A typical rectangular cross-section has been assumed for analysis to derive the critical equations.

This chapter analyses several parameters which influence the behaviour of the beam-column joint for unbound concrete and presents necessary relationships by implementing neural network techniques. Also, the investigation is extended to evaluate a procedure to determine the degree of confinement produced by the lateral reinforcement. Fundamental assumptions were applied, particularly that a plane cross-section remains plane before and after bending, the tensile strength of concrete
is negligible, the strains in the concrete and steel are of the same level and finally the strain in the concrete is linearly proportional to distance from the neutral axis. Generally, strain may be categorised into the following four groups: elastic, creep, shrinkage and lateral. The research conducted in this study considers only elastic strain.

The work in this chapter simulates the process of experimental research by learning from initial and specific data and extending the work to predict the results of the research. With the knowledge obtained from new studies and codes of practice the work can be extended to derive relationships obtained by converting the knowledge into sets of examples, implementing the FORTRAN program, and generating adequate training and test data for the neural networks. This chapter focuses on neural network techniques by implementing the backpropagation learning algorithm in its several forms through the supervised training technique. One essential requirement of this approach is the selection of feasible, appropriate training and test data to explore the neural network in representing the ultimate flexural member in unbound and bound concrete. Work done by Nirjar [1] is presented in detail and more recent research equations, adapted to comply with the new code of practice were implemented in the final work.

5.2. STEEL REINFORCEMENT PROPERTIES

The steel reinforcement plays a major part in the concrete behaviour of reinforced members, both in the tension and compression zones, particularly in reducing the dimensions of the section. Their properties are classified on the basis of minimum yield strength, generally referred to as the grade of the steel. The steel grade is thus an indication of the minimum yield strength. The steel reinforcement is available in several shapes viz. bars, welded wire fabric and wires and are produced in accordance with specific standards. The essential properties of steel bars are: yield strength, $f_y$, modulus of elasticity, $E_s$, and ultimate strength, $f_u$. 
5.3. CONCRETE TENSILE STRENGTH

Concrete is a brittle material so that its tensile strength greatly affects the extent and the size of cracking in the concrete member. Experimental investigators over the years have found difficulty in determining the actual tensile strength of the concrete, due to stress concentration at the specimen ends and minor misalignments which result in unacceptable data. A test procedure, known as the split cylinder test, utilises a concrete cylinder dimension, 150 mm in diameter, 300 mm high loaded perpendicular to its longitudinal axis. The tensile strength of concrete, $f_t$, is relatively low, equal to 20% of the compressive strength when considering cracking and deformation of the reinforced concrete members, the tensile strength of concrete is a major property and is represented as the modulus of rupture. Several investigators have related cylinder strength to the concrete tensile strength. According to ACI committee 435 [77] the concrete tensile strength, for normal weight concrete is given by:

$$f_t = 0.627 \sqrt{f_c}$$  \[(5.1)\]

Warwaruck [78], at the University of Illinois, adopted the expression:

$$f_t = \frac{21}{3 + 84/f_c}$$ \[(5.2)\]

And according to Teychenne [79]:

$$f_t = 0.7 \sqrt{f_c}$$ \[(5.3)\]

The Comité Européen du Béton (CEB) [80] adopted the expression:

$$f_t = 0.272 (f_c)^{2/3}$$ \[(5.4)\]

Nirjar [1] proposed the following relationship on the basis of the flexural test to represent the modulus of rupture of concrete:
\[ f_i = 0.65 \sqrt{f_e} \]  

(5.5)

Where \( f_e \) and \( f_i \) are expressed in \( N / mm^2 \).

5.3.1. Prediction of Concrete Strength Versus Modulus of Rupture

The relationship between concrete cylinder strength and modulus of rupture is vital for any future study. The network chosen to establish such a relationship has one processing element, (PE), in the input layer representing the concrete cylinder strength and one PE in the output layer representing the modulus of rupture. The first hidden layer has 12 PE's and the second hidden layer has 9 PE's. Delta-Rule with the sigmoid transfer function is used in the network with a learning rate of 0.3 in the first hidden layer, 0.25 in the second layer and 0.15 in the output layer while maintaining a momentum of 0.5. The network was trained with a \( RMS \) error convergence value of 2.0%. Figure 5.1(a) represents the trained network and Figure 5.1(b) represents the trained and tested network. The full generation code equation predicted by neural networks for the modulus of rupture is shown in Appendix B and the final formulae are produced in the following form:

\[ y_{24}^{out} = x_{24}^{out} \times (7.415) + (-0.982) \]  

(5.6)

\[ x_{24}^{out} = \frac{1.0}{1.0 + e^{-\Sigma x_{24}}} \]  

(5.7)

where:

- \( x_{24}^{out} \) is the modulus of rupture predicted within the neural network.
- \( y_{24}^{out} \) is the modulus of rupture scaled by network output, \( N / mm^2 \).
(a) Trained network.

(b) Trained and Tested Network.

Figure 5.1. Concrete Strength Versus Tensile Concrete Networks.
The results for selected concrete cylinder strengths against the modulus of rupture with the predicted values by neural network are shown in Table 5.1.

Table 5.1. Modulus of Rupture.

<table>
<thead>
<tr>
<th>$f'_{c}$ (N/mm^2)</th>
<th>ACI Committee Equation (5.1)</th>
<th>Warwaruk Equation (5.2)</th>
<th>Teychenne Equation (5.3)</th>
<th>CEB Equation (5.4)</th>
<th>Nirjar Equation (5.5)</th>
<th>Neural Networks Equation (5.6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2.80</td>
<td>2.92</td>
<td>3.13</td>
<td>2.00</td>
<td>2.91</td>
<td>2.8</td>
</tr>
<tr>
<td>25</td>
<td>3.14</td>
<td>3.30</td>
<td>3.50</td>
<td>2.33</td>
<td>3.25</td>
<td>3.2</td>
</tr>
<tr>
<td>30</td>
<td>3.43</td>
<td>3.62</td>
<td>3.83</td>
<td>2.63</td>
<td>3.56</td>
<td>3.6</td>
</tr>
<tr>
<td>35</td>
<td>3.71</td>
<td>3.89</td>
<td>4.14</td>
<td>2.91</td>
<td>3.85</td>
<td>3.8</td>
</tr>
<tr>
<td>40</td>
<td>3.97</td>
<td>4.12</td>
<td>4.43</td>
<td>3.18</td>
<td>4.11</td>
<td>4.1</td>
</tr>
<tr>
<td>45</td>
<td>4.21</td>
<td>4.32</td>
<td>4.70</td>
<td>3.44</td>
<td>4.36</td>
<td>4.2</td>
</tr>
</tbody>
</table>

Figure 5.2 shows the relationship between concrete cylinder strength and modulus of rupture as obtained by previous investigators together with that obtained by neural networks.
5.4. RECTANGULAR SECTIONS UNDER FLEXURAL STRESSES

Extensive experimental testing, theoretical research and analytical procedures have resulted in a rigorous theorem and efficient codes of practice which consequently simplified the understanding of the fundamental structural behaviour of reinforced concrete members.

External loading on structural members produces deformation strains in the beam member which induce flexural stresses. Reinforced concrete beams are heterogeneous in nature, made up of two different materials, concrete and steel. Analysis of a reinforced concrete member is therefore a combination of empirical and rational methods. The ultimate strength analysis of a reinforced concrete section involves the evaluation of the stress-block parameters, which are entirely dependent
on the concrete modulus of elasticity, $E_c$, the concrete strain at maximum stress, $\varepsilon_0$, the concrete strain at ultimate stress, $\varepsilon_u$, and finally the stress-strain relationship, $f-\varepsilon$. Therefore, the four main factors which must be evaluated to obtain the stress-block parameters are:

1. Concrete modulus of elasticity, $E_c$.
2. Concrete strain characteristics at maximum stress, $\varepsilon_0$.
3. Concrete strain characteristics at ultimate stress, $\varepsilon_u$.
4. Concrete stress-strain relationship, $f-\varepsilon$.

These factors are evaluated experimentally by carrying out a series of tests on concrete cylinders and prisms of fixed dimensions. In the absence of experimental test procedures, neural network techniques can produce new methods of evaluating these missing factors by manipulating large amounts of previous test data and theoretical procedures to provide rapid and acceptable solutions.

5.4.1. Concrete Modulus of Elasticity

The concrete modulus of elasticity, $E_c$, and the stress-strain relationship are important properties for design purposes. $E_c$ is taken as the slope of the initial linear portion of the stress-strain curve, is defined as the ratio of compressive stress to compressive strain, and is a measure of the stiffness or resistance of the material to deformation. The concrete modulus of elasticity depends on various factors such as the aggregate type, water/cement ratio, aggregate/cement ratio, the period for curing, the rate of loading and age at loading. A substantial amount of research work has been carried out and several empirical formulae have been proposed to establish methods and procedures to identify the value of the concrete modulus of elasticity and serve as a guide in the absence of experimental results. ACI Committee 435 [77] provided the following expression for the concrete modulus of elasticity:
\[ E_c = 33w_c^{1.5}\sqrt{f_c'} \quad \text{for } 90 < w_c < 155 \]  

(5.8)

For normal and sand stone concrete, equation (5.8) is expressed as:

\[ E_c = 57,000\sqrt{f_c'} \quad \text{for } w_c \approx 145 \]  

(5.9)

where:

\[ w_c = \text{concrete density in pounds per cubic feet, } lb/ft^3. \]
\[ E_c = \text{Modulus of Elasticity of concrete, } lb/in^2. \]
\[ f_c' = \text{cylinder compressive strength of concrete, } lb/in^2. \]

ACI code 318-89 [81] expressed the concrete modulus of elasticity, \( E_c \), in terms of SI equivalent for normal and sand stone as:

\[ E_c = 4.730\sqrt{f_c'} \]  

(5.10)

where:

\( E_c \) is expressed in \( kN/mm^2 \), and \( f_c' \) in \( N/mm^2 \).

In terms of the cube strength, the concrete modulus of elasticity, as given in equation (5.9) can be expressed as:

\[ E_c = 50,000\sqrt{f_{cu}} \]  

(5.11)

where:

\[ f_{cu} = \text{characteristic concrete strength, } lb/in^2. \]

The CEB [80] proposed an expression for \( E_c \) as a characteristic of concrete strength as:

\[ E_c = 70,000\sqrt{f_{cu}} \]  

(5.12)
The expression by CEB [80] can be expressed in the SI system as:

\[ E_c = 6.58 \sqrt{f'_c} \]  

(5.13)

where:

\[ f'_c = 0.78 f_{cu}; \quad E_c \text{ is expressed in } kN/mm^2, \quad f'_c \text{ and } f_{cu} \text{ are expressed in } N/mm^2. \]

Another expression adopted by Saenz [82] in discussing the stress-strain curve of Desayi and Krishnan [83] is:

\[ E_c = \frac{10^3 \sqrt{f'_c}}{1 + 0.006 \sqrt{f'_c}} \]  

(5.14)

If \( E_c \) is expressed in \( kN/mm^2 \) and \( f'_c \) in \( N/mm^2 \), equation (5.14) becomes:

\[ E_c = \frac{8.3 \sqrt{f'_c}}{1 + 0.07 \sqrt{f'_c}} \]  

(5.15)

The expression by Saenz [82] provides a higher value of '\( R = E_c/E_0 \) for concrete with lower strengths, where \( E_0 \) is the secant modulus at strain \( \varepsilon_0 \) in \( kN/mm^2 \).

Nirjar [1] provided a simplified expression for the concrete modulus of elasticity based on his experimental data and adopted the following expression:

\[ E_c = 5.2 \sqrt{f'_c} \]  

(5.16)

5.4.1.1. Prediction of Relationship between Concrete Strength and Modulus of Elasticity

The equations for \( E_c \) predicted by neural network are in the following form:

\[ y_{53}^{out} = x_{53}^{out} \ast (65.082) + (-5.537) \]  

(5.17)

\[ x_{53}^{out} = \frac{1.0}{1.0 + e^{-\Sigma x_{53}}} \]  

(5.18)
\( x_{53}^{\text{out}} \) is the modulus of elasticity predicted within the neural network.

\( y_{53}^{\text{out}} \) is the modulus of elasticity scaled by network output, \( N / \text{mm}^2 \).

Backpropagation network is implemented with the sigmoid transfer function, using the Delta-Rule learning. The network topology consists of one input PE representing the cylinder concrete strength and one output PE with 0.2 learning rate representing the modulus of elasticity. The first hidden layer consists of 30 PE's with a learning rate of 0.6 and the second hidden layer consists of 20 PE's with a learning rate of 0.3. The training has 342 patterns while the test has 42 patterns. The momentum was kept at 0.5 with an epoch size of 10. The architecture, topology and neurodynamics for concrete strength versus modulus of elasticity for the trained and test networks are shown in Figures 5.3(a) and 5.3(b).
Figure 5.3. Concrete Cylinder Strength Versus Modulus of Elasticity.
The curves for concrete cylinder strength against the modulus of elasticity produced by various investigations are shown in Figure 5.4 together with the values for $E_c$ given by neural network. Table 5.2 shows the values for the modulus of elasticity provided by other investigators as well as these predicted by neural network.

![Figure 5.4. Variations of Concrete Cylinder Strength with Modulus of Elasticity.](image-url)
## Table 5.2. Concrete Modulus of Elasticity.

<table>
<thead>
<tr>
<th>$f'_c$</th>
<th>ACI (5.10)</th>
<th>CEB (5.13)</th>
<th>Saenz (5.15)</th>
<th>Nirjar (5.16)</th>
<th>Neural Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>$E_c = 4.73\sqrt{f'_c}$</td>
<td>$E_c = 6.58\sqrt{f'_c}$</td>
<td>$E_c = \frac{8.3\sqrt{f'_c}}{1 + 0.07\sqrt{f'_c}}$</td>
<td>$E_c = 5.2\sqrt{f'_c}$</td>
<td>Neural Networks (5.17)</td>
</tr>
<tr>
<td>20</td>
<td>18.32</td>
<td>25.48</td>
<td>25.29</td>
<td>20.17</td>
<td>20.0</td>
</tr>
<tr>
<td>25</td>
<td>21.15</td>
<td>29.43</td>
<td>28.27</td>
<td>23.26</td>
<td>24.0</td>
</tr>
<tr>
<td>30</td>
<td>23.65</td>
<td>32.90</td>
<td>30.74</td>
<td>26.00</td>
<td>27.0</td>
</tr>
<tr>
<td>35</td>
<td>25.91</td>
<td>36.04</td>
<td>32.86</td>
<td>28.48</td>
<td>30.0</td>
</tr>
<tr>
<td>40</td>
<td>27.98</td>
<td>38.93</td>
<td>34.72</td>
<td>30.76</td>
<td>32.0</td>
</tr>
<tr>
<td>45</td>
<td>31.73</td>
<td>44.14</td>
<td>37.89</td>
<td>34.88</td>
<td>36.0</td>
</tr>
</tbody>
</table>
5.4.2. Concrete Strain Characteristics at Maximum Stress

Experimental studies over the years have demonstrated that eccentric compressive loading on prism specimens provide higher peak points than concentric compression loading for the relationship between stress and strain. This may be due to differences in the rate of load application. Popovics [84] presented a review of stress-strain relationships as well as a state-of-the-art summary, evaluation and analysis of the axial deformation of concrete under loading. The empirical expression adopted by Nirjar [1] provides a good relationship between stress and strain at maximum stress, as shown in Figure 5.5, and is expressed as:

\[ \varepsilon_0 = 0.9 \times 10^{-3} \sqrt{f_0} \]  

(5.19)

where the maximum stress, \( f_0 \), is expressed in \( N / mm^2 \).

The maximum concrete stress is assumed to be the concrete cylinder strength, i.e. \( f_0 = f_c \). In general, strains within the range of 0.002 have obtained from most investigators and therefore used in the author's study.
Figure 5.5. Concrete Strain at Maximum Stress by Nirjar [1].

5.4.3. Concrete Strain Characteristics at Ultimate Stress

Determination of the concrete strain at ultimate stress experimentally was difficult due to the nature of failure of the specimen and the need to define the curve beyond the point of maximum stress. The normal experimental procedure is to test a number of prisms and cylindrical specimens and take readings beyond the point of maximum stress until failure occurs. Many investigators have assumed values for the concrete strain at ultimate based on an analytical procedure. The expression for the concrete strain at ultimate adopted by Nirjar [1] assumes $f_0 = f'_c$, is shown in Figure 5.6, and expressed as:

$$
\epsilon_u = 7.5 \times 10^{-3} \sqrt{f_0}
$$

(5.20)

The value of ultimate strain provided by the ACI code 318-89 relationship [81] is 0.003, and by Hognestad [85], 0.0038. In practice an ultimate strain of 0.003 is
acceptable, and has been adopted for this research work. Utilising equations (5.19) and equation (5.20), the expression for the strain in the concrete at ultimate stress is expressed by Nirjar [1] as:

\[
\frac{\varepsilon_u}{\varepsilon_0} = 8.33 \times (f_0)^{-\frac{1}{3}}
\]  

(5.21)

![Figure 5.6. Concrete Strain at Ultimate Stress by Nirjar [1].](image)

**5.4.4. Concrete Stress-Strain Relationship**

The non-linear behaviour of concrete has been well documented. Investigators have attempted to represent the stress-strain relationship in terms of mathematical curves such as the hyperbolic or cubic parabola. These mathematical relationships represent the ascending part of the stress-strain relationship but have provided an
insight into the non-linearity of the concrete stress-strain relationship. An analysis of the cross-section requires a knowledge of the stress-strain relationship. This relationship influenced by a number of factors and so is difficult to define by only one method. Over the past decade a considerable amount of research has been directed towards the inelastic behaviour of concrete and many theories have been proposed which define the stress-strain relationship in terms of elastic and inelastic behaviour. Other theories have tried to simplify the approach by redefining the area of the stress block in order to simplify the computation of the compressive force and its location. Experimental and analytical work on the behaviour of reinforced concrete sections under load has provided a considerable amount of information and a knowledge of the strain as well as stress pattern is important in understanding the mechanism of behaviour of the cross-section.

The performance of a reinforced concrete members depends to a large extent on the stress-strain relationships both of the concrete and the steel. The stress-strain behaviour of the concrete is determined by its strength, rate of loading, materials properties, age at loading and nature of the loading. Structural members are normally stressed in different directions simultaneously, although uniaxial stresses can be justified in most cases.

A typical stress-strain curve for a concrete cylinder at initial loading shows an approximate linear relationship up to 40% of ultimate cylinder strength $f'_c$. This defines the initial or tangent modulus of elasticity, the form of curvature beyond this point is shown in Figure 5.7, for specimens loaded in compression at 28 days.
Figure 5.7. Typical Stress-Strain Relationship.

The relationship between stress and strain is influenced by a number of factors and this creates the problem of defining the relationship by a specific method. Many investigators have proposed several hypotheses in an attempt to provide a simplified stress-strain relationship. For examples, stress-strain relationships have been proposed in the form of triangles, rectangles as well as trapezoidal shapes. One investigation emphasised the use of new testing techniques to provide a more complete stress-strain relationship over a wide-range of loading and which could handle larger values of strain. Rüsch [86] proposed the use of stress-strain curves obtained from by loading the concrete prisms concentrically in an effort to obtain the stress distribution in the compressive zone in flexure.

The stress-strain relationship proposed by Hognestad [85] assumed a parabolic shape from the origin up to the maximum stress and then a linear stress-strain relationship as shown in Figure 5.8.
The expression involved is in the form:

\[ f = [2\left(\frac{\varepsilon}{\varepsilon_0}\right) - \left(\frac{\varepsilon}{\varepsilon_0}\right)^2]f_0 \quad \text{for } 0 \leq \varepsilon \leq \varepsilon_0 \] (5.22)

and,

\[ f = [1 - 0.15\left(\frac{\varepsilon - \varepsilon_0}{\varepsilon_u - \varepsilon_0}\right)]f_0 \quad \text{for } \varepsilon_0 < \varepsilon \leq \varepsilon_u \] (5.23)

where the strain at maximum stress, \( \varepsilon_0 = 0.002 \) and ultimate strain, \( \varepsilon_u = 0.0038 \).

Sahlin [87] and Smith and Young [88] proposed a stress-strain relationship based on an exponential expression for the behaviour of the concrete under flexural compression. This was expressed as:

\[ f = \left[\left(\frac{\varepsilon}{\varepsilon_0}\right)e^{\left(\frac{1 - \varepsilon}{\varepsilon_0}\right)}\right]f_0 \] (5.24)

The value for \( \varepsilon_0 \) suggested by Sahlin [87] was 0.002. Smith and Young [88] suggested a value for \( \varepsilon_0 \) between 0.017 to 0.002, assuming that the \( f_0 = f_c \).
Tulin and Gerstle [89] in the discussion paper by Desai and Krishnan [83] proposed the following stress-strain relationship based on their own experimental work:

\[
(a+1)f_0 \varepsilon \\
\frac{f}{a+b} = \frac{\varepsilon_0}{a+\left(\frac{\varepsilon}{\varepsilon_0}\right)^b}
\]  

(5.25)

The coefficient values 'a' and 'b' were selected from their experimental work and are adjusted to obtain the best fit for the experimental results. It was found that values of '2' and '3' were selected to fit 'a' and 'b' respectively. The strain at maximum stress was assumed to be \(\varepsilon_0 = 0.002\).

Nirjar [1] modified the expression obtained by Tulin and Grestle [89] for his experimental work and adopted a similar expression in the form:

\[
f = \frac{\varepsilon}{k_1 + k_2 \varepsilon^n}
\]  

(5.26)

where the variables \(k_1\), \(k_2\) and \(n\) are obtained by satisfying the following conditions:

(a) \( f = 0 \) at \( \varepsilon = 0 \); ie at the point of origin.

(b) \( \frac{df}{dc} = E_c \) at \( \varepsilon = 0 \); ie the slope of the stress-strain curve at the origin is the modulus of elasticity.

(c) \( f = f_0 \) at \( \varepsilon = \varepsilon_0 \); ie the maximum stress occurred at \( \frac{df_0}{dc_0} = 0 \).

Nirjar [1] applied these conditions to equation (5.26) to determine the following coefficients:

\[
k_1 = \frac{1}{E_c};
\]
where:

\[ R = \frac{E_c}{E_0} \]

\[ E_0 = \frac{f_0}{\varepsilon_0} \]

and equation (5.26) can be expressed as:

\[ f = \left[ \frac{R \varepsilon}{E_0} \right] f_0 \]

\[ f = \left[ \frac{R \varepsilon}{E_0} \right] f_0 \]

Equation (5.27) as defined by Nirjar [1], represents a more acceptable stress-strain relationship which is more practical to use due to the absence of the definition of the descending portion of the curve. Nirjar [1] then simplified equation (5.27) to the following expression:

\[ f = \left[ \frac{2 \varepsilon}{E_0} \right] f_0 \]

\[ f = \left[ \frac{2 \varepsilon}{E_0} \right] f_0 \]

This modified equation is influenced by the shape of the stress-strain curve. The ratio \( R = 2 \) plays a major role by representing the relationship between the secant modulus and the modulus of elasticity of concrete.

Young [90] proposed three separate stress-strain functions which were independent of experimental beam testing and can be expressed in the following form:
\[
f = \left( E_f \varepsilon_0 - 2 f_c \right) \frac{\varepsilon^3}{\varepsilon_0^3} - \left( 2 E_f \varepsilon_0 - 3 f_c \right) \frac{\varepsilon^2}{\varepsilon_0^2} + E_f \varepsilon 
\]

(5.29)

\[
f = f'_c \frac{\varepsilon}{\varepsilon_0} e^{\left(0 - \frac{\varepsilon}{\varepsilon_0}\right)} 
\]

(5.30)

\[
f = f'_c \sin \frac{\pi \varepsilon}{2 \varepsilon_0} 
\]

(5.31)

The values of the ratio of strain at maximum stress to ultimate strain obtained for equation (5.29):

\[
\frac{\varepsilon_0}{\varepsilon_u} = 0.80 
\]

\[
\frac{\varepsilon_0}{\varepsilon_u} = 0.70 
\text{ for equation (5.30)}
\]

\[
\frac{\varepsilon_0}{\varepsilon_u} = 0.75 
\text{ for equation (5.31)}
\]

where:

\( E_T \) is the concrete tangent modulus of elasticity = 1000 \( f_0 \).

Desayi and Krishnan [83] introduced the following simplified stress-strain relationship:

\[
f = \frac{E_c \varepsilon}{1 + \left( \frac{\varepsilon}{\varepsilon_0} \right)^2} 
\]

(5.32)

in which \( E_c = 2 \frac{f_0}{\varepsilon_0} \)

and \( f_0 = f'_c \).

The values of \( \varepsilon_0 \) and the corresponding stress are important in equation (5.32), in which an average value of 0.002 is used. The authors proposed values of 0.003 for \( \varepsilon_u \), a corresponding stress of \( 7/8 f'_c \).
5.4.4.1. Prediction of Stress Versus Strain

The previous parameter evaluations for the modulus of elasticity, strain at maximum stress, and strain at ultimate of the concrete are necessary to define the state of the stress-strain relationship. This relationship has been presented by several expressions which all define the characteristic behaviour of the stress-strain curve.

The investigation into the stress-strain relationships in this study fall into two main categories, namely the preparatory stage and the renew stage, as mentioned in chapter four. The preparatory stage evaluates the stress-strain relationship obtained experimentally and analytically by Nirjar [1]. Evaluation is achieved by selecting concrete cylinder strengths of 20, 25, 30, 35 and 40 $N/\text{mm}^2$ and training each curve individually for selected points along the curve. With the numerical methods, the curves proposed by Nirjar [1] are used to generate the stress-strain relationship for each concrete strength. Certain points along each curve which are not generated are used to predicate the neural network results. Experimental data by Nirjar [1] is also used as part of the training data to enhance the network performance.

The process of determining the correct neural networks architecture with topology and neurodynamics requires delicate techniques to avoid underfitting, which results in selecting fewer hidden layers and PE's, and to avoid overfitting, which results in more hidden layers and PE's. A balance is required which will produce a neural network that will not memorize the data or lack the ability to generalise. A standard backpropagation network is selected for all the concrete cylinder strengths. These are composed of one PE in the input layer representing the strain, 10 PE's in the hidden layer and one PE in the output layer representing the stress. The Delta-Rule learning rule was implemented with a learning rate of 0.3 for the hidden layer, 0.15 for the output layer and with a momentum of 0.5. A Hyperbolic Tangent transfer function is used which performs effectively with the Delta-Rule. An epoch size of 16 was selected with RMS convergence criteria of
0.3%. Table 5.3 shows the full parameters which were used the preparatory stage to process the network predictions. Table 5.3 also provides the network results performance. The evaluation results are shown in Table 5.4, 5.5, 5.6, 5.7, and 5.8 for concrete strengths of 20, 25, 30, 35 and 40 $N/mm^2$ respectively. The strain values in the shaded area were passed to the trained network to predict the stress-strain relationship. Table 5.4 shows that large errors occur at high concrete strengths in the experimental testing. The results obtained are encouraging for most cases and the

Table 5.3. Parameters Used in Stress-Strain Relationship for Preparatory Stage.

<table>
<thead>
<tr>
<th>$f_c$ $N/mm^2$</th>
<th>Input Data Sets</th>
<th></th>
<th></th>
<th>Network Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Set</td>
<td>Test Set</td>
<td>Cycle Learn</td>
<td>Confusion Matrix</td>
</tr>
<tr>
<td>20</td>
<td>82</td>
<td>12</td>
<td>27757</td>
<td>0.9999</td>
</tr>
<tr>
<td>25</td>
<td>87</td>
<td>12</td>
<td>14118</td>
<td>0.9994</td>
</tr>
<tr>
<td>30</td>
<td>91</td>
<td>14</td>
<td>29383</td>
<td>0.9998</td>
</tr>
<tr>
<td>35</td>
<td>94</td>
<td>13</td>
<td>31889</td>
<td>0.9999</td>
</tr>
<tr>
<td>40</td>
<td>85</td>
<td>12</td>
<td>78619</td>
<td>0.9999</td>
</tr>
</tbody>
</table>

error computation was calculated for the neural network and compared to the experimental and analytical results. This demonstrates the potentiality of adopting such a procedure to minimise the time required to record the test data by relying on specific points on the curve and allowing the neural network to obtain the results.
Table 5.4. Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f' = 20 \text{N/mm}^2$

<table>
<thead>
<tr>
<th>Strain $\varepsilon*10^{-3}$</th>
<th>Nirjar [1]</th>
<th>Neural Networks</th>
<th>Computation of Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Experimental = E</td>
<td>Analytical = A</td>
<td>$f$ ($\text{N/mm}^2$)</td>
</tr>
<tr>
<td>0.45</td>
<td>9.50</td>
<td>9.62</td>
<td>9.99</td>
</tr>
<tr>
<td>0.90</td>
<td>16.00</td>
<td>15.98</td>
<td>17.02</td>
</tr>
<tr>
<td>1.35</td>
<td>19.00</td>
<td>19.05</td>
<td>19.45</td>
</tr>
<tr>
<td>1.80</td>
<td>20.00</td>
<td>19.99</td>
<td>19.96</td>
</tr>
<tr>
<td>2.25</td>
<td>19.25</td>
<td>19.77</td>
<td>19.92</td>
</tr>
<tr>
<td>2.70</td>
<td>17.50</td>
<td>19.05</td>
<td>19.35</td>
</tr>
<tr>
<td>3.15</td>
<td>15.50</td>
<td>18.13</td>
<td>18.24</td>
</tr>
<tr>
<td>3.55</td>
<td>14.50</td>
<td>17.26</td>
<td>17.27</td>
</tr>
</tbody>
</table>
Table 5.5. Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f_c = 25 \text{ N/mm}^2$

<table>
<thead>
<tr>
<th>Nirjar [1]</th>
<th>Neural Networks</th>
<th>Computation of Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Strain</td>
<td>Experimental = E</td>
<td>Analytical = A</td>
</tr>
<tr>
<td>$\epsilon \times 10^{-3}$</td>
<td>$f \ (\text{N/mm}^2)$</td>
<td>$f \ (\text{N/mm}^2)$</td>
</tr>
<tr>
<td>0.50</td>
<td>11.75</td>
<td>12.08</td>
</tr>
<tr>
<td>1.00</td>
<td>20.00</td>
<td>20.21</td>
</tr>
<tr>
<td>1.50</td>
<td>24.00</td>
<td>24.04</td>
</tr>
<tr>
<td>2.00</td>
<td>25.00</td>
<td>25.00</td>
</tr>
<tr>
<td>2.50</td>
<td>24.00</td>
<td>24.47</td>
</tr>
<tr>
<td>3.00</td>
<td>22.50</td>
<td>23.30</td>
</tr>
<tr>
<td>3.35</td>
<td>21.00</td>
<td>22.33</td>
</tr>
</tbody>
</table>
Table 5.6. Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_e = 30\, N/mm^2$

<table>
<thead>
<tr>
<th>Strain</th>
<th>Nirjar [1]: Experimental = E</th>
<th>Nirjar [1]: Analytical = A</th>
<th>Neural Networks: $f'(N/mm^2)$</th>
<th>Computation of Error</th>
<th>%Error = $\text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \times 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon \times 10^{-3}$</td>
<td>$f(N/mm^2)$</td>
<td>$f(N/mm^2)$</td>
<td>$f(N/mm^2)$</td>
<td>N.(w.r.t) E</td>
<td>N.(w.r.t) A</td>
</tr>
<tr>
<td>0.54</td>
<td>14.50</td>
<td>14.43</td>
<td>14.88</td>
<td>2.62</td>
<td>3.84</td>
</tr>
<tr>
<td>1.08</td>
<td>24.50</td>
<td>24.36</td>
<td>25.72</td>
<td>4.98</td>
<td>5.58</td>
</tr>
<tr>
<td>1.60</td>
<td>28.75</td>
<td>28.90</td>
<td>29.29</td>
<td>1.88</td>
<td>1.35</td>
</tr>
<tr>
<td>2.15</td>
<td>30.00</td>
<td>29.99</td>
<td>29.95</td>
<td>0.17</td>
<td>0.13</td>
</tr>
<tr>
<td>2.70</td>
<td>29.00</td>
<td>29.10</td>
<td>29.22</td>
<td>0.76</td>
<td>0.41</td>
</tr>
<tr>
<td>3.00</td>
<td>28.00</td>
<td>28.22</td>
<td>28.26</td>
<td>0.93</td>
<td>0.14</td>
</tr>
<tr>
<td>3.20</td>
<td>27.00</td>
<td>27.55</td>
<td>27.54</td>
<td>2.00</td>
<td>0.04</td>
</tr>
</tbody>
</table>
Table 5.7. Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_c = 35 \text{ N/mm}^2$

<table>
<thead>
<tr>
<th>Strain $\varepsilon \times 10^{-3}$</th>
<th>Nirjar [1]</th>
<th>Neural Networks</th>
<th>Computation of Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Experimental = E</td>
<td>Analytical = A</td>
<td>Network = N</td>
</tr>
<tr>
<td>$f (\text{N/mm}^2)$</td>
<td>$f (\text{N/mm}^2)$</td>
<td>$f (\text{N/mm}^2)$</td>
<td>%Error = Abs(1 - \frac{\text{Network}}{\text{Desired}}) * 100</td>
</tr>
<tr>
<td>0.55</td>
<td>16.50</td>
<td>16.08</td>
<td>16.65</td>
</tr>
<tr>
<td>1.10</td>
<td>28.50</td>
<td>27.73</td>
<td>29.54</td>
</tr>
<tr>
<td>1.65</td>
<td>33.50</td>
<td>33.55</td>
<td>34.19</td>
</tr>
<tr>
<td>2.20</td>
<td>35.00</td>
<td>35.00</td>
<td>34.96</td>
</tr>
<tr>
<td>2.85</td>
<td>34.00</td>
<td>33.71</td>
<td>33.77</td>
</tr>
<tr>
<td>3.10</td>
<td>33.00</td>
<td>32.81</td>
<td>32.80</td>
</tr>
</tbody>
</table>
Table 5.8. Stress-Strain Comparison Between Nirjar [1] and Neural Network for $f'_c = 40 \text{ N/mm}^2$

<table>
<thead>
<tr>
<th>Strain</th>
<th>Experimental = E</th>
<th>Analytical = A</th>
<th>Neural Networks</th>
<th>Computation of Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon \times 10^{-3}$</td>
<td>$f (\text{N/mm}^2)$</td>
<td>$f (\text{N/mm}^2)$</td>
<td>Network = N</td>
<td>%Error = $\text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \times 100$</td>
</tr>
<tr>
<td>0.56</td>
<td>18.00</td>
<td>17.67</td>
<td>18.26</td>
<td>1.44</td>
</tr>
<tr>
<td>1.125</td>
<td>33.00</td>
<td>31.09</td>
<td>32.77</td>
<td>0.70</td>
</tr>
<tr>
<td>1.690</td>
<td>38.50</td>
<td>38.12</td>
<td>38.78</td>
<td>0.73</td>
</tr>
<tr>
<td>2.250</td>
<td>40.00</td>
<td>40.00</td>
<td>39.95</td>
<td>0.13</td>
</tr>
<tr>
<td>2.800</td>
<td>39.25</td>
<td>38.96</td>
<td>39.00</td>
<td>0.10</td>
</tr>
<tr>
<td>3.00</td>
<td>38.25</td>
<td>38.20</td>
<td>38.22</td>
<td>0.08</td>
</tr>
</tbody>
</table>
Once a stress-strain relationship is established from a sample of the experimental and analytical procedure, the relationship is enhanced by studying the theoretical, historical relationships and other experimental investigations. The renew stage is then implemented by updating the relationship. The stress-strain relationships studied earlier by other investigators are vital since they will provide predictive relationships for any further research. The stress-strain study includes four values of cylinder strength, 20, 30, 35 and 40 $N/\text{mm}^2$.

To achieve an adequate comparison, a standard network for all cylinder strengths was selected consisting of one PE in the input layer to represent the strain, 50 PE's in the first hidden layer, 25 PE's in the second hidden layer and one PE in the output layer representing the stress. Each concrete strength has 2157 training and 267 test patterns. The test sets are extracted to provide the best picture representations and to increase confidence in the trained network. This is achieved by extracting 9% of the training set through FORTRAN programming. The training examples were presented to the network randomly to maximise performance and to avoid the possibility of the network memorising the data. The supervised learning was conducted and the Extended-Delta-Bar-Delta was employed for a learning rate of 0.3 for the first hidden layer, 0.25 for the second hidden layer and 0.15 for the output layer with a momentum of 0.4 for all the networks. The sigmoid transfer function, which represents the positive data of the stress-strain relationship, is used. Training was performed until a convergence of 3% was achieved through the RMS instrument. The trained and test networks along with their instruments are given in Appendix C.

The predicted equations for concrete strength of 20 $N/\text{mm}^2$ are:

\begin{align}
y_{78}^{\text{out},20} &= x_{78}^{\text{out},20} \times (33.333) + (-6.667) \\
x_{78}^{\text{out},20} &= \frac{1.0}{1.0 + e^{-\Sigma x_{78}}} 
\end{align}

(5.33) (5.34)
The corresponding equations for the remaining concrete strengths are:

$30 \text{ N/mm}^2$:

$$y_{78}^{out,30} = x_{78}^{out,30} \ast (49.999) + (-10.0) \quad (5.35)$$

$$x_{78}^{out,30} = \frac{1.0}{1.0 + e^{-\sum x_{78}}} \quad (5.36)$$

$35 \text{ N/mm}^2$:

$$y_{78}^{out,35} = x_{78}^{out,35} \ast (58.333) + (-11.667) \quad (5.37)$$

$$x_{78}^{out,35} = \frac{1.0}{1.0 + e^{-\sum x_{78}}} \quad (5.38)$$

$40 \text{ N/mm}^2$:

$$y_{78}^{out,40} = x_{78}^{out,40} \ast (66.666) + (-13.333) \quad (5.39)$$

$$x_{78}^{out,40} = \frac{1.0}{1.0 + e^{-\sum x_{78}}} \quad (5.40)$$

where:

$x_{78}^{out}$ is the stress predicted within the neural network for specific concrete strength.

$y_{78}^{out}$ is the stress scaled by neural network output for specific concrete strength.

Full representations of the above relationships are shown in Figures 5.9, 5.10, 5.11 and 5.12 respectively.
Figure 5.9. Stress Strain Curves for $f_c' = 20N/mm^2$.

Figure 5.10. Stress Strain Curves for $f_c' = 30N/mm^2$. 
Figure 5.11. Stress Strain Curves for $f_c = 35N/mm^2$.

Figure 5.12. Stress Strain Curves for $f_c = 40N/mm^2$. 
5.5. STRESS-BLOCK COMPUTATION FOR FLEXURAL MEMBERS

Extensive research work has been conducted to determine the ultimate moment a cross section can sustain. This moment depends on the assumed stress distribution in the compression zone and the values assumed for the ultimate flexural strain in the concrete. The cross-section reaches its maximum flexural capacity when the product of the total compressive force and the internal lever are at a maximum. A theoretical formulation of the ultimate moment capacity of the cross-section is based on several parameters. The distribution of compressive stress across the section plays a major role in assessing the section's properties at ultimate load. Although the stress-block for flexural members has been computed by several investigators, only three have been presented in detail to cover work completed during the past decade.

According to the steel percentage, the member can behave in three possible modes failure. The first mode, under-reinforced, is where the steel percentage is less than that required to balance the concrete in compression. The second mode, balanced, is where the steel and concrete fails simultaneously. The third mode, over-reinforced, is where the concrete fail before the steel yields.

The concrete stress-block has been evaluated by many investigators over the years. In the 1940's, Whitney [91] proposed an equivalent rectangular stress distribution in an attempt to simplify the calculations based on a parabolic shape for the stress distribution. His work was based on the ultimate analysis. In 1943, Jensen [92] proposed an idealised trapezoidal stress distribution and derived the properties of the trapezoid shape as a function of the cylindrical strength. This approach was based on the ultimate strength of the reinforced concrete beams. Hongestad [85] proposed a rectangular idealised stress distribution as a result of reviewing several proposed idealised stress distributions. Mattock, Kriz and Hognestad [93] presented a clear and concise ultimate strength relationship based on an equivalent stress distribution in the concrete compressive zone. The theory adopted by the ACI Code
318-89 [81] is widely used and implemented in practice. The following analysis presents three possible approaches to evaluate the theoretical ultimate moment. The first approach is based on an 'accurate' stress distribution of the compressive stress derived from the classical method of evaluating compressive stress and the ultimate moment capacity of a cross-section, as adopted by Nirjar [1]. The second method is based on the simplified approach used by ACI code 318-89 [81]. Finally, the third method is based on neural network techniques. This final approach is presented and compared with the two other approaches.

5.5.1. Stress-Block Computation Adopted by Nirjar for Flexural Members

Figure 5.13 shows a rectangular cross-section with typical stress-strain distributions for ultimate conditions. By applying the equations of equilibrium to evaluate the ultimate moment capacity of the cross-section, the ultimate moment is obtained as:

\[
M_u = \int_A f \cdot y dA
\]  

(5.41)

\[
(yd)C_u = \int_0^{x_u d} f \cdot b \cdot y dy
\]  

(5.42)

where:

- \( C_u \) = ultimate compressive force, Newtons.
- \( f \) = concrete stress at a distance 'y' above the neutral axis as expressed by the equation, \( N/mm^2 \).
- \( x_u d \) = distance of the neutral axis from compression face at ultimate, \( mm \).
- \( yd \) = depth to the resultant concrete compressive force from the neutral axis, \( mm \).
\[ b = \text{width of the cross-section, mm.} \]

From Figure 5.13(b), applying strain compatibility:

\[ \frac{y}{x_u d} = \frac{\varepsilon}{\varepsilon_u} \]  

\[ y = \frac{\varepsilon}{\varepsilon_u} (x_u d) \]  

Differentiating with respect to the strain:

\[ dy = \frac{x_u d}{\varepsilon_u} d\varepsilon \]  

Substituting the values of 'y' and 'dy' obtained from equations (5.44) and (5.45) respectively into equation (5.42):

\[ (yd)C_v = \int_0^{x_u d} b * f\left(\frac{\varepsilon}{\varepsilon_u}\right)(x_u d)\left(\frac{x_u d}{\varepsilon_u}\right) d\varepsilon \]  

Thus by integrating and substituting for 'f' as obtained from equation (5.28), the above expression becomes:
\[(yd)C_u = 2b(x_u d)^2 f_0 \left( \frac{E_0}{E_u} - \frac{\tan^{-1} \frac{E_u}{E_0}}{E_0} \right) \quad (5.47)\]

The concrete compressive force acting on the section is computed from the equilibrium equation for force as follows:

\[C_u = \int_A f \ast dA \quad (5.48)\]

\[C_u = \int_0^x f \ast bdy \quad (5.49)\]

Substituting equation (5.45) into equation (5.49) gives:

\[C_u = \int_0^x f \ast b \frac{x_u d}{E_u} \ast d \epsilon \quad (5.50)\]

Substituting for "f" from equation (5.28), and integrating:

\[C_u = bx_u f_0 \frac{E_0}{E_u} \log_e \left( 1 + \left( \frac{E_u}{E_0} \right)^2 \right) \quad (5.51)\]

\[C_u = bx_u d \ast f_{av} \quad (5.52)\]

where:

\[f_{av} = f_c \frac{E_u}{E_u} \log_e \left( 1 + \left( \frac{E_u}{E_0} \right)^2 \right) \quad (5.53)\]

The concrete cylinder compressive strength, \(f_c\), is assumed equal to \(f_0\), and the values of \(E_0\) and \(E_u\) are obtained from equations (5.19) and (5.20), respectively. Then the distance of the neural axis from compressive face at ultimate can be determined according to the following failure conditions:

(a) For a tension failure, assuming the tension steel has yielded:

\[x_u d = \frac{A_s}{b} \left( \frac{f_y}{f_{av}} \right) \quad (5.54)\]

or:
\[ x_u = \rho \left( \frac{f_y}{f_{av}} \right) \quad \text{where} \quad \rho = \frac{A_s}{bd} \]

(b) For a compression failure where the steel has not yielded:

\[ x_u d = \frac{A_s}{b} \left( \frac{f_s}{f_{av}} \right) \]  \hspace{1cm} (5.55)

or:

\[ x_u = \rho \left( \frac{f_s}{f_{av}} \right) \]  \hspace{1cm} (5.56)

where the steel stress is expressed as:

\[ f_s = E_s \varepsilon_s \]  \hspace{1cm} (5.57)

and

\[ \rho = \text{tensile reinforcement ratio of steel.} \]
\[ f_y = \text{yield stress of the main reinforcement, } N/mm^2. \]
\[ A_s = \text{area of tension steel reinforcement, } mm^2. \]

Therefore the ultimate moment capacity of the cross-section can be expressed:

\[ M_u = C_u (d - x_u d + yd) \]  \hspace{1cm} (5.58)

and for:

\[ gd = (x_u d - yd) : \]
\[ M_u = C_u (d - gd) \]  \hspace{1cm} (5.59)

so that:

\[ M_u = bd^2 \cdot f_{av} (1 - g)x_u \]  \hspace{1cm} (5.60)
The above expressions for the compressive force and ultimate moment capacity of the cross-section are based entirely on the classical method and essentially adopts the procedure suggested by Nirjar [1].

5.5.2. Stress-Block Computation Adopted by ACI Code for Flexural Members

Recent advances in the understanding of concrete technology and material properties have resulted in the adaptation of the equivalent shape given by Whitney [91] and simplified by Mattock, Kriz and Hognestad [93] without loss of accuracy. This has provided a simplified calculation for the evaluation of the compressive force and ultimate moment of a cross-section. Figure 5.14 shows the simplified representation of the ultimate compressive stress-block adopted by the new ACI [81] Code of Practice.

![Stress-Strain Distribution](image)

Figure 5.14. Stress-Strain Distribution.

For the doubly reinforced rectangular cross-section represented in Figure 5.14, the compressive force in the concrete and the ultimate moment of the section is determined as follows:

from strain compatibility:
\[ x_u d = \frac{\varepsilon_u}{\varepsilon_u + \varepsilon_Y} d \]  

(5.61)

from statics of the internal equilibrium forces shown in Figure 5.14(c):

\[ \sum F = 0: \]

\[ C_u = T - C_s \]  

(5.62)

\[ 0.85 f'_c \beta_1 b(x_u d) = A_s f_s - A_r f_s \]  

(5.63)

\[ x_u d = \frac{A_s f_s - A_r f_s}{0.85 f'_c \beta_1 b} \]  

(5.64)

where:

\[ 2gd = \beta_1 (x_u d) \]  

(5.65)

The moment equation can be expressed as:

\[ M_u = 0.85 f'_c \beta_1 (x_u d)b(d - gd) + A_s f_s (d - ad) \]  

(5.66)

\[ = (A_s f_s - A_r f_s )(d - gd) + A_r f_s (d - ad) \]  

(5.67)

where:

\[ 2gd = \frac{A_s f_s - A_r f_s}{0.85 f'_c b} \]  

(5.68)

The above procedure using the equivalent stress-block, represents the general analysis equation where yielding of both the compression and tension reinforcement takes place (Figure 5.15(a)):

For strain compatibility:

\[ \frac{x_u d}{\varepsilon_u} = \frac{ad}{\varepsilon_u - \varepsilon_Y} \]  

or

(5.69)
\[ x_d = \frac{\varepsilon_u - \varepsilon_y}{\varepsilon_u - \varepsilon_y} \]

and from statics:

\[ 0.85\beta_1 f_c b(x_d) + A_s f_s = A_s f_y \tag{5.70} \]

\[ A_s f_y - A_s f_s = 0.85\beta_1 f_c (\frac{\varepsilon_u}{\varepsilon_u - \varepsilon_y}) ad \tag{5.71} \]

Assuming yielding of the compression reinforcement:

\[ f_y = f_y \]

\[ \frac{A_s - A_s}{bd} = \frac{0.85\beta_1 f_c ad}{f_y} \frac{600.0}{d} \frac{600.0 - f_y}{600.0 - f_y} \tag{5.72} \]

Assuming:

\[ \frac{A_s - A_s}{bd} \geq \frac{0.85\beta_1 f_c ad}{f_y} \frac{600.0}{d} \frac{600.0 - f_y}{600.0 - f_y} \tag{5.73} \]

then \( f_y = f_y \)

The strain condition at this stage is illustrated in Figure 5.15(a).

(a) Compression and Tensile Reinforcement Yielding

(b) Balanced Reinforcement Content

Figure 5.15. Strain Conditions.
The strain condition at this stage for balanced steel content is shown in Figure 5.15(b). From statics (Figure 5.14(c)):

\[ C_u + C_s = T \]  \hspace{1cm} (5.74)

\[ 0.85 \beta_1 f'(x_u d)_{bal} * b + A_s f_s = f_s A_s \] \hspace{1cm} (5.75)

Hence distance of the neutral axis from the compression face for the balanced condition is as follows:

\[ (x_u d)_{bal} = (\frac{\varepsilon_u}{\varepsilon_u + \varepsilon_y})d \] \hspace{1cm} (5.76)

\[ 0.85 \beta_1 f' (\frac{\varepsilon_u}{\varepsilon_u + \varepsilon_y})bd + \rho b d f'_{sbal} = \rho b df_y \] \hspace{1cm} (5.77)

\[ \rho_b = \frac{0.85 \beta_1 f' (600.0)}{f_y (600.0 + f_y)} + \rho \cdot \frac{f'_{sbal}}{f_y} \] \hspace{1cm} (5.78)

If \( \rho \leq \rho_b \) then the tension steel yields.

The ACI code [81] requirement is as follows:

\[ \rho \leq \frac{3}{4} \rho_b + \rho \cdot \frac{f'_{sbal}}{f_y} \] \hspace{1cm} (5.79)

where:

\[ \rho_b = 0.85 \beta_1 f' (\frac{600.0}{600.0 + f_y}) \] \hspace{1cm} (5.80)

\[ f'_{sbal} = E_s [\varepsilon_u - \frac{ad}{d} (\varepsilon_u + \varepsilon_y)] \leq f_y \]

\[ f'_{sbal} = 600.0 - \frac{ad}{d} (600.0 + f_y) \leq f_y \] \hspace{1cm} (5.81)

\[ f'_{sbal} \] = yield stress of the main reinforcement for balanced condition, \( N/mm^2 \).
Analysis of a doubly reinforced beam involves the following two main cases:

**Case A:** \( \rho \leq \rho_b + \rho \cdot \frac{f_{shal}}{f_y} \)  

(5.82)

Therefore:

\[
\frac{A_s - A_y}{bd} \geq \frac{0.85 \beta_1 f_c' \alpha d}{f_y} \left( \frac{\varepsilon_u}{d - \varepsilon_y} \right)
\]

(5.83)

or

\[
\frac{A_s - A_y}{bd} \geq \frac{0.85 \beta_1 f_c' \alpha d}{f_y} \left( \frac{600.0}{600.0 - f_y} \right)
\]

(5.84)

i.e. both the tension steel and the compression steel have reached their yield stress and the ultimate moment of the cross-section is found from:

\[
M_u = (A_s - A_y) f_y (d - gd) + A_y f_y (d - ad)
\]

(5.85)

**Case B:** \( \rho \leq \rho_b + \rho \cdot \frac{f_{shal}}{f_y} \)  

(5.86)

\[
\frac{A_s - A_y}{bd} < \frac{0.85 \beta_1 f_c' \alpha d}{f_y} \left( \frac{\varepsilon_u}{d - \varepsilon_y} \right)
\]

(5.87)

then

\[ f_s = f_y \]

and

\[ f_s' < f_y \]

\[
f_s' = \varepsilon_s E_s \frac{x_s d - ad}{x_u d}
\]

(5.88)

\[
A_y f_y = 0.85 \beta_1 f_c' b(x_u d) + A_s \varepsilon_s E_s \frac{x_u d - ad}{x_u d}
\]

(5.89)

Use equation (5.89) and solve for \( x_u d \), then use equation (5.88) and solve for \( f_s' \).

The ultimate moment of the cross-section for **Case B** is then found from:
\[ M_u = 0.85 f_c b d (2gd)(1-a) + A_f f_y d(1-a) \]  \hspace{1cm} (5.90)

or

\[ M_u = d (A_x f_y - A_x f_y)(1-a) + A_f f_y d(1-a) \]  \hspace{1cm} (5.91)

For a single reinforced beam the equation for ultimate moment simplifies to:

\[ M_u = A_x f_y (d - ad) \]  \hspace{1cm} (5.92)

The value of the stress-block depth factor, \( \beta_1 \), is determined from the following conditions, where \( f'_c \) is expressed in \( N/mm^2 \):

\[
\beta_1 = \begin{cases} 
0.85 & \text{for } 0 < f'_c \leq 27.6 \\
0.85 - 0.05 \left( \frac{f'_c - 27.6}{6.875} \right) & \text{for } 27.6 < f'_c \leq 55.2 \\
0.65 & \text{for } f'_c > 55.2
\end{cases}
\]  \hspace{1cm} (5.93)

5.5.3. Stress-Block Computation at Preparatory Stage for Flexural Members

The preparatory stage investigates the application of the stress-block adopted by Nirjar [1] to determine the ultimate moment of a cross-section by analysing the 34 specimens tested. Taking into account all 34 specimen provides data on 6 concrete strengths (20, 25, 30, 35, 40 and 45 \( N/mm^2 \)), 3 different forms of reinforcement (2, 3 and 4 bars), 4 bar diameters (6, 8, 10, 12), i.e. a total of 2448 possible combinations of parameters. To overcome this problem, a FORTRAN program was carried out to provide sufficient training and testing data to train the neural network to determine the ultimate moment. To demonstrate the neural networks, capability of predicting results, the program was applied to 34 specimens, comprising 3 types of concrete strength (20, 30, 45 \( N/mm^2 \)), 2 format reinforcement types (2 bars and 4 bars) and 2 bar sizes (6 and 12). The number of possible combinations for analysis can therefore reduced from 2440 to 408. The number of possible combination for applying the
The neural network selected to predict the ultimate moment consisted of 7 input PE's ($b, d, f_c, f_\sigma, \rho, x_u, g$), two hidden layers comprising of 75 and 50 hidden PE's respectively, and one output PE representing the ultimate moments. The input training set selected consist of 243. The test set consisted of 29 patterns which were not used for training. This has an impact on the results and are essential requirements for this type of approach. Extended-Delta-Bar-Delta is used with sigmoidal transfer function such that the learning rate for the two hidden layers are 0.30 and 0.25 respectively, whilst the output layer has a 0.15 learning rate. The momentum for the network is kept at 0.4 with an epoch size of 16 and 0.2% for the RMS as the criteria for convergence. During training the network reached convergence after 68763 cycles with a confusion matrix of 0.9997 for both the trained and test networks. The classification rates were 0.5 and 1.0 for the trained and test networks respectively. The fully trained and tested neural networks is shown in Figure 5.16.
Figure 5.16. Trained and Tested Networks for Ultimate Moment at Preparatory Stage.
The final formulae predicted by the neural network for the preparatory stage can be expressed in the following form:

\[ y_{134}^{\text{out,p}} = x_{134}^{\text{out,p}} \times (8.583) + (-0.027) \]  \hspace{1cm} (5.94)

\[ x_{134}^{\text{out,p}} = \frac{1.0}{1.0 + e^{-x_{134}}} \]  \hspace{1cm} (5.95)

\( x_{134}^{\text{out,p}} \) is the moment predicted within the neural network.

\( y_{134}^{\text{out,p}} \) is the scaled moment by network output, \( kN-m \).

The results for the three strengths of concrete (20, 30, 45 N/mm\(^2\)) are shown in Table 5.9, 5.11 and 5.13 respectively. The results for the other three concrete strengths (25, 35 and 40 N/mm\(^2\)) are shown in Table 5.10, 5.12 and 5.14 respectively, but were never trained in the network. The results of the percentages of error are also tabulated to include the minimum and maximum values. These clearly indicate that neural network application represents a method of evaluating the ultimate moments for cross-section which is comparable with established methods using experimental and analytical procedures. The experimental investigations however require extensive data recording over an extended period of time.
Table 5.9. Comparison of Ultimate Moments Produced by Nirjar [1] and by Neural Networks for $f_c' = 20N / mm^2$.

<table>
<thead>
<tr>
<th>$f_c'$ (N/mm^2)</th>
<th>$f_{av}$ (N/mm^2)</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>$x_{av} \cdot d$</th>
<th>$gd$</th>
<th>$M_u$, kN-m Nirjar [1]</th>
<th>$M_u$, kN-m Neural Networks</th>
<th>%Error $= \frac{Abs(T_{Network} - \text{Desired})}{\text{Desired}} \times 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>16.08</td>
<td>0.0072</td>
<td>14.26</td>
<td>6.61</td>
<td>1.69</td>
<td>1.70</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0108</td>
<td>21.39</td>
<td>9.91</td>
<td>2.44</td>
<td>2.32</td>
<td>4.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0144</td>
<td>28.51</td>
<td>13.21</td>
<td>3.15</td>
<td>3.15</td>
<td>0.0Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0128</td>
<td>25.35</td>
<td>11.74</td>
<td>2.84</td>
<td>2.77</td>
<td>2.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0192</td>
<td>38.02</td>
<td>17.62</td>
<td>3.99</td>
<td>4.19</td>
<td>5.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0200</td>
<td>39.60</td>
<td>18.35</td>
<td>4.12</td>
<td>4.34</td>
<td>5.34Max</td>
</tr>
</tbody>
</table>
Table 5.10. Comparison of Ultimate Moment Produced by Nirjar [1] and by Neural Networks for $f_c^t = 25N/mm^2$. (Not seen by network)

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$f_{av}$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>$x_{ud}$</th>
<th>$gd$</th>
<th>$M_u$, kN-m</th>
<th>$M_u$, kN-m</th>
<th>%Error = Abs((Desired - Network) / Desired) * 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/mm²</td>
<td>N/mm²</td>
<td>mm</td>
<td>mm</td>
<td></td>
<td>Nirjar [1]</td>
<td>Neural Networks</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>19.94</td>
<td>0.0072</td>
<td>11.50</td>
<td>4.6</td>
<td>1.72</td>
<td>1.71</td>
<td>0.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0108</td>
<td>17.25</td>
<td>6.9</td>
<td>2.52</td>
<td>2.36</td>
<td>6.35</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0144</td>
<td>22.99</td>
<td>9.2</td>
<td>3.28</td>
<td>3.26</td>
<td>0.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0128</td>
<td>20.44</td>
<td>8.18</td>
<td>2.95</td>
<td>2.84</td>
<td>3.73</td>
</tr>
<tr>
<td>0.0192</td>
<td>30.66</td>
<td>12.26</td>
<td></td>
<td></td>
<td>4.24</td>
<td>4.53</td>
<td>6.84</td>
</tr>
<tr>
<td>0.0256</td>
<td>40.88</td>
<td>16.35</td>
<td></td>
<td></td>
<td>5.4</td>
<td>5.67</td>
<td>5.00</td>
</tr>
<tr>
<td>0.0200</td>
<td>31.94</td>
<td>12.77</td>
<td></td>
<td></td>
<td>4.39</td>
<td>4.71</td>
<td>7.29 Max</td>
</tr>
<tr>
<td>0.0030</td>
<td>47.90</td>
<td>19.16</td>
<td></td>
<td></td>
<td>6.13</td>
<td>6.13</td>
<td>0.0 Min</td>
</tr>
<tr>
<td>0.0288</td>
<td>45.99</td>
<td>18.39</td>
<td></td>
<td></td>
<td>5.94</td>
<td>6.02</td>
<td>1.35</td>
</tr>
</tbody>
</table>
Table 5.11. Comparison of Ultimate Moment Produced by Nirjar [1] and by Neural Networks for $f_c' = 30N/mm^2$.

<table>
<thead>
<tr>
<th>$f_c'$</th>
<th>$f_{av}$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>$x_u d$</th>
<th>$gd$</th>
<th>$M_u$, kN-m Nirjar [1]</th>
<th>$M_u$, kN-m Neural Networks</th>
<th>%Error = $\frac{Abs(1 - \frac{Network}{Desired})}{100}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N/mm^2$</td>
<td>$N/mm^2$</td>
<td>$mm$</td>
<td>$mm$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>23.63</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0072</td>
<td>9.70</td>
<td>3.32</td>
<td>1.74</td>
<td>1.74</td>
<td>0.0Min</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0108</td>
<td>14.55</td>
<td>4.99</td>
<td>2.57</td>
<td>2.39</td>
<td>7.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0144</td>
<td>19.40</td>
<td>6.65</td>
<td>3.37</td>
<td>3.35</td>
<td>0.59</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0128</td>
<td>17.24</td>
<td>5.91</td>
<td>3.02</td>
<td>2.89</td>
<td>4.30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0192</td>
<td>25.87</td>
<td>8.87</td>
<td>4.39</td>
<td>4.75</td>
<td>8.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0256</td>
<td>34.49</td>
<td>11.82</td>
<td>5.68</td>
<td>6.01</td>
<td>5.86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0200</td>
<td>26.94</td>
<td>9.23</td>
<td>4.56</td>
<td>4.96</td>
<td>8.77Max</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0300</td>
<td>40.42</td>
<td>13.85</td>
<td>6.51</td>
<td>6.48</td>
<td>0.46</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.0288</td>
<td>38.8</td>
<td>13.30</td>
<td>6.29</td>
<td>6.38</td>
<td>1.43</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 5.12. Comparison of Ultimate Moments Produced by Nirjar [1] and by Neural Networks for $f'_c = 35 N / mm^2$. (Not seen by network)

<table>
<thead>
<tr>
<th>$f'_c$</th>
<th>$f_{av}$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>$x_u d$</th>
<th>$gd$</th>
<th>$M_u$, kN-m</th>
<th>$M_u$, kN-m</th>
<th>% Error = $\frac{Abs{\text{Nirjar} - \text{Network}}}{\text{Desired}}$ * 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>27.17</td>
<td>0.0072</td>
<td>8.44</td>
<td>2.45</td>
<td>1.76</td>
<td>1.76</td>
<td>0.0%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0108</td>
<td>12.66</td>
<td>3.67</td>
<td>2.61</td>
<td>2.43</td>
<td>6.90%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0144</td>
<td>16.87</td>
<td>4.89</td>
<td>3.43</td>
<td>3.42</td>
<td>0.29%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0128</td>
<td>15.00</td>
<td>4.35</td>
<td>3.07</td>
<td>2.94</td>
<td>4.23%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0192</td>
<td>22.50</td>
<td>6.53</td>
<td>4.50</td>
<td>4.91</td>
<td>9.11%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0256</td>
<td>30.00</td>
<td>8.70</td>
<td>5.87</td>
<td>6.22</td>
<td>5.96%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0200</td>
<td>23.44</td>
<td>6.80</td>
<td>4.68</td>
<td>5.12</td>
<td>9.4% max</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0300</td>
<td>35.16</td>
<td>10.20</td>
<td>6.77</td>
<td>6.68</td>
<td>1.33%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0288</td>
<td>33.75</td>
<td>9.79</td>
<td>6.53</td>
<td>6.58</td>
<td>0.77%</td>
</tr>
</tbody>
</table>
Table 5.13. Comparison of Ultimate Moments Produced by Nirjar [1] and by Neural Networks for $f'_c = 40 \text{N/mm}^2$. (Not seen by network)

<table>
<thead>
<tr>
<th>$f'_c$</th>
<th>$f_{av}$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>$x_ad$</th>
<th>$gd$</th>
<th>$M_u$, kN-m</th>
<th>$M_u$, kN-m</th>
<th>%Error = $\frac{Abs(Neural - Desired)}{Desired}$ * 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/mm$^2$</td>
<td>N/mm$^2$</td>
<td>mm</td>
<td>mm</td>
<td></td>
<td>Nirjar [1]</td>
<td>Neural Networks</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>30.56</td>
<td>0.0072</td>
<td>7.50</td>
<td>1.81</td>
<td>1.77</td>
<td>1.80</td>
<td>1.69</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0108</td>
<td>11.25</td>
<td>2.71</td>
<td>2.63</td>
<td>2.47</td>
<td>6.08</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0144</td>
<td>15.00</td>
<td>3.62</td>
<td>3.48</td>
<td>3.47</td>
<td>0.29</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0128</td>
<td>13.34</td>
<td>3.21</td>
<td>3.10</td>
<td>2.99</td>
<td>3.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0192</td>
<td>20.00</td>
<td>4.82</td>
<td>4.58</td>
<td>5.00</td>
<td>9.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0256</td>
<td>26.67</td>
<td>6.43</td>
<td>6.01</td>
<td>6.35</td>
<td>5.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0200</td>
<td>20.84</td>
<td>5.02</td>
<td>4.76</td>
<td>5.23</td>
<td>9.87\text{Max}</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0300</td>
<td>31.26</td>
<td>7.53</td>
<td>6.96</td>
<td>6.80</td>
<td>2.30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0288</td>
<td>30.00</td>
<td>7.23</td>
<td>6.70</td>
<td>6.70</td>
<td>0.0\text{Min}</td>
</tr>
</tbody>
</table>
Table 5.14. Comparison of Ultimate Moment by Nirjar [1] and by Neural Networks for $f_c = 45N/mm^2$.

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$f_{av}$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>$x_u d$</th>
<th>$gd$</th>
<th>$M_u$, $kN-m$</th>
<th>$M_u$, $kN-m$</th>
<th>% Error = $\frac{Abs(1 - \frac{\text{Network}}{\text{Desired}})}{100}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N/mm^2$</td>
<td>$N/mm^2$</td>
<td>$mm$</td>
<td>$mm$</td>
<td></td>
<td>Nirjar [1]</td>
<td>Neural Networks</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>33.81</td>
<td>0.0072</td>
<td>6.78</td>
<td>1.32</td>
<td>1.78</td>
<td>1.84</td>
<td>3.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0108</td>
<td>10.17</td>
<td>1.98</td>
<td>2.65</td>
<td>2.52</td>
<td>4.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0144</td>
<td>13.56</td>
<td>2.64</td>
<td>3.51</td>
<td>3.52</td>
<td>0.28Min</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0128</td>
<td>12.05</td>
<td>2.35</td>
<td>3.13</td>
<td>3.04</td>
<td>2.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0192</td>
<td>18.08</td>
<td>3.53</td>
<td>4.64</td>
<td>5.06</td>
<td>9.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0256</td>
<td>24.10</td>
<td>4.70</td>
<td>6.11</td>
<td>6.42</td>
<td>5.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0200</td>
<td>18.83</td>
<td>3.67</td>
<td>4.83</td>
<td>5.29</td>
<td>9.52Max</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0300</td>
<td>28.25</td>
<td>5.51</td>
<td>7.11</td>
<td>6.87</td>
<td>3.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.0288</td>
<td>27.12</td>
<td>5.29</td>
<td>6.84</td>
<td>6.77</td>
<td>1.02</td>
</tr>
</tbody>
</table>
5.5.4. Stress-Block Computation at Renew Stage for Flexural Members

The renew stage was introduced to enhance the method explained previously and also to apply the ACI Code of Practice [81]. The procedure involved generating training and testing data files as in the preparatory stage used by Nirjar [1] and then updating the data by applying the provisional of the new code. The input training set at this stage consist of 486 and the test set has 58 patterns. Learning and training started by selecting an error tolerance of 0.2% in the RMS instrument and using the same topology network but varying the learning rate for the first hidden layer, second hidden layer and the output layer to 0.4, 0.35 and 0.25 respectively. The momentum was increased to 0.6 due to an initial assessment of the network which did not converge. The network ceased learning at 310508 cycles with confusion matrix of 0.9998 and classification rate of 1.0 for the trained network, while the training and tested network had 0.9998 and 1.0 for the confusion matrix and classification rate respectively. The fully trained and tested neural networks is shown in Figure 5.17.

A comparison of the enhancement results for the renew stage are shown in Table 5.15 for concrete strengths of 20, 25 and 30 $N/mm^2$ and Table 5.16 for concrete strengths of 35, 40 and 45 $N/mm^2$. 


Figure 5.17. Trained and Tested Networks for Ultimate Moment at Renew Stage.
Table 5.15. Comparison of Percent Errors Due to Moment Computations for Renew Stage. \( \% \text{Error} = \frac{\text{Abs}(1 - \frac{\text{Network}}{\text{Desired}})}{100} \)

| \( f_c \) \( N / \text{mm}^2 \) | \( f_{av} \) | \( \rho = \frac{A_t}{bd} \) | \( x_vd \) | \( gd \) | \( M_u \) | \( M_u \) | \( \% \text{Error} \) | \( f_{av} \) \( N / \text{mm}^2 \) | \( \rho = \frac{A_t}{bd} \) | \( x_vd \) | \( gd \) | \( M_u \) | \( M_u \) | \( \% \text{Error} \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 16.08 | 0.0072 | 14.26 | 6.61 | 1.69 | 1.67 | 1.18_{\text{Min}} | 17.00 | 0.0072 | 15.86 | 6.74 | 1.68 | 1.67 | 0.60_{\text{Min}} |
| 20 | 16.08 | 0.0108 | 21.39 | 9.91 | 2.44 | 2.30 | 5.74_{\text{Max}} | 17.00 | 0.0192 | 42.3 | 17.98 | 3.98 | 4.15 | 4.27_{\text{Max}} |
| 25 | 19.94 | 0.0144 | 22.99 | 9.20 | 3.28 | 3.28 | 0.0_{\text{Min}} | 21.25 | 0.0144 | 25.38 | 10.79 | 3.23 | 3.22 | 0.31_{\text{Min}} |
| 25 | 19.94 | 0.0200 | 31.94 | 12.77 | 4.39 | 4.67 | 6.38_{\text{Max}} | 21.25 | 0.0108 | 19.03 | 8.09 | 2.49 | 2.33 | 6.43_{\text{Max}} |
| 30 | 23.63 | 0.0072 | 9.70 | 3.32 | 1.74 | 1.74 | 0.0_{\text{Min}} | 25.5 | 0.0288 | 43.19 | 17.98 | 5.96 | 5.95 | 0.17_{\text{Min}} |
| 30 | 23.63 | 0.0200 | 26.94 | 9.23 | 4.56 | 4.95 | 8.55_{\text{Max}} | 25.5 | 0.0108 | 16.20 | 6.74 | 2.53 | 2.33 | 7.91_{\text{Max}} |
Table 5.16. Comparison of Percent Errors Due to Moment Computations for Renew Stage. $\%Error = Abs(1 - \frac{Network}{Desired}) \times 100$

<table>
<thead>
<tr>
<th>$f'_c$</th>
<th>$f_{av}$</th>
<th>$\rho = \frac{A_t}{bd}$</th>
<th>$x_d' \cdot d$</th>
<th>$gd$</th>
<th>$M_u$</th>
<th>$% Error$</th>
<th>$f_{av}$</th>
<th>$\rho = \frac{A_t}{bd}$</th>
<th>$x_d' \cdot d$</th>
<th>$gd$</th>
<th>$M_u$</th>
<th>$M_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>27.17</td>
<td>0.0072</td>
<td>8.44</td>
<td>2.45</td>
<td>1.76</td>
<td>1.76</td>
<td>0.00Min</td>
<td>29.75</td>
<td>0.0288</td>
<td>38.71</td>
<td>15.41</td>
<td>6.14</td>
</tr>
<tr>
<td>35</td>
<td>27.17</td>
<td>0.0020</td>
<td>23.44</td>
<td>6.80</td>
<td>4.68</td>
<td>5.13</td>
<td>9.62Max</td>
<td>29.75</td>
<td>0.0200</td>
<td>26.89</td>
<td>10.70</td>
<td>4.49</td>
</tr>
<tr>
<td>40</td>
<td>30.56</td>
<td>0.0288</td>
<td>30.00</td>
<td>7.23</td>
<td>6.70</td>
<td>6.68</td>
<td>0.30Min</td>
<td>34.00</td>
<td>0.0288</td>
<td>35.50</td>
<td>13.48</td>
<td>6.27</td>
</tr>
<tr>
<td>40</td>
<td>30.56</td>
<td>0.0200</td>
<td>20.84</td>
<td>5.02</td>
<td>4.76</td>
<td>5.24</td>
<td>10.08M</td>
<td>34.00</td>
<td>0.0200</td>
<td>24.65</td>
<td>9.36</td>
<td>4.55</td>
</tr>
<tr>
<td>45</td>
<td>33.81</td>
<td>0.0144</td>
<td>13.56</td>
<td>2.64</td>
<td>3.51</td>
<td>3.54</td>
<td>0.85Min</td>
<td>38.25</td>
<td>0.0072</td>
<td>8.28</td>
<td>3.00</td>
<td>1.75</td>
</tr>
<tr>
<td>45</td>
<td>33.81</td>
<td>0.0200</td>
<td>18.83</td>
<td>3.67</td>
<td>4.83</td>
<td>5.30</td>
<td>9.73Max</td>
<td>38.25</td>
<td>0.0200</td>
<td>23.01</td>
<td>8.32</td>
<td>4.99</td>
</tr>
</tbody>
</table>
The final general formulae for ultimate moment at renew stage are determined from the following equations:

\[ y_{134}^{\text{out, renew}} = x_{134}^{\text{out, renew}} \cdot (8.60) + (-0.040) \]  (5.96)

\[ x_{134}^{\text{out, renew}} = \frac{1.0}{1.0 + e^{-\Sigma x_{134}}} \]  (5.97)

\( y_{134}^{\text{out, renew}} \) is the moment predicted within the neural network.

\( x_{134}^{\text{out, renew}} \) scaled moments by network output, \( kN-m \).

The use of neural network models to predict the ultimate moment of the resistance of the cross-section is based on a new and fundamental concept which introduces an explicit form of the stress-strain relationship obtained by neural networks through generation of the necessary parameters such as the modulus of elasticity, the strain at maximum stress and the strain at ultimate stress. This results in a significant reduction in the number of specimens required for testing and hence reduces the cost.

5.6. CONFINED PROPERTIES OF FLEXURAL MEMBERS

The presence of shear reinforcement in the flexural member provides confinement to the member which results in increased strength, ductility and prevents buckling of the main reinforcement. Confinement of the concrete can also be achieved by reducing the spacing of transverse the reinforcement resulting in an increase in the transverse steel ratio. This confinement is concentrated in the core surrounded by the concrete cover and behaviour of the concrete is completely different despite being subjected to the same type of loading. This behaviour produces an additional complication in the analysis procedure. Experimental evidence has suggested however that the ascending region of the stress-strain curve can be adopted for both confined and unbound concrete [11].
Most investigators have ignored the effect of this transverse reinforcement and only a few investigators have considered the effect of transverse reinforcement in the behaviour of a flexural member [11]. The effect of spiral reinforcement in columns was investigated by Richart and Brown [94]. This study demonstrated that circular spirals confine the concrete more effectively than rectangular transverse reinforcement due to the shape of the spiral which applies a uniform circular pressure. Various investigators, including King [95], Chan [11] and Pfister [96] have investigated the influence of confinement on structural concrete members subjected to different type of loading. They observed that the characteristics of the concrete stress-strain relationship varies according to the value of Poisson's ratio which is affected by the transverse reinforcement around the core. They also observed that the value of the Modulus of Elasticity for the concrete not affected by the transverse reinforcement. The expressions for deriving the stress-block related to bound concrete, however, indicate that the confining stress is influenced by the spacing and amount of transverse reinforcement, and can be expressed as:

\[ f'_c = 0.9 f'_c (1 + 0.05q^*) \]  

(5.98)

The average stress for bound concrete is given by:

\[ f'_{av} = 0.72 f'_c (1 + 0.144 \sqrt{q^*}) \]  

(5.99)

\[ q^* = 1.4 \left( \frac{A_h}{A_c} - 0.45 \right) \frac{A_v (s_0 - s_v)}{A_s + 0.0028 B(s_v)^2} \]  

(5.100)

Figure 5.18 shows these details for a bound concrete cross-section.
where:

\( f'_c \) = cylinder compressive strength of concrete, \( N/mm^2 \).

\( f'_e \) = compressive strength of bound concrete analogous to \( f'_c \), \( N/mm^2 \).

\( f_{av} \) = average concrete compressive stress for bound concrete, \( N/mm^2 \).

\( q^* \) = parameter for the effectiveness of the lateral reinforcement.

\( A_b \) = area of bound concrete under compression (\( A_b = b'x' \)), \( mm^2 \).

\( A_c \) = area of concrete under compression (\( A_c = bx \)), \( mm^2 \).

\( A_{av} \) = area of one leg of a link, \( mm^2 \).
\[ B = \text{the greater area of } (0.7 \times') \text{ or } (0.7 b'), \text{ mm.} \]

\[ s_v = \text{spacing of the lateral reinforcement, mm.} \]

\[ s_o = \text{longitudinal spacing of the links at which the lateral reinforcement was not effective in confining the concrete, mm.} \]

Chan [11] produced an expression for the ultimate properties of bound concrete by investigating the failures of compression members with different amounts of confinement. The expression used by Chan [11] was extended by Burns and Siess [97] who introduced a binding ratio, known as \( p'' \), which takes into consideration the property of the longitudinal reinforcement which Chan [11] did not use. They defined the parameter \( p'' \) as:

\[ p'' = \frac{V_s}{V_b} + 0.1 \frac{D}{s_v} \] (5.101)

\( p'' \) = binding ratio parameter to allow for the effect of confinement.

\[ v_s = \text{volume of stirrups, mm}^3. \]

\[ v_b = \text{volume of bound concrete, mm}^3. \]

\[ D = \text{diameter of the compression steel, mm.} \]

Based on the above equation, the average compressive stress for bound concrete can be represented as:

\[ f'_{av} = f_{av}(1 + 10p'') \] (5.102)

With this knowledge the strength of the confined section can be determined from the expression used for the unbound concrete, viz:
\[ C_u = f'_{av} b^* x_u d \] (5.103)

The distance from the neutral axis to the compression face is determined from:

\[ x_u = \rho \frac{f_y}{f_{av}} \] (5.104)

The ultimate strength capacity of the section for the confined section is found from:

\[ M_u = f'_{av} x_u (1 - g) b d^2 \] (5.105)

The ratio of the distance from the neutral axis to the compression face was computed by Soliman and Yu [98] as follows:

\[ \frac{g}{x_u} = \frac{0.84 + 0.5q^*}{2 + q} \] (5.106)

Assuming values for the maximum confined concrete strength, \( f'_c \), and the ratio \( g/x_u \), the stress-block parameters can be determined for the bound concrete section from equations (5.98) to (5.106). The methodology presented for the confined section is therefore to determine the ultimate strength capacity of a cross-section by establishing a procedure which is based on behaviour of an unconfined section.

5.7. CONCRETE BIAXIAL STRESS-STRAIN RELATIONSHIP

The behaviour of a Reinforced Concrete Joint involves the interaction of several key phenomena such as shear, anchorage, bond, fatigue and confinement of concrete. All these parameters create problems in establishing the stress-strain relationship for concrete. Despite intensive research and many theoretical studies on the performance of beam-column joints under load a general theory on the failure criteria has not been produced. The main research in this area of study has been devoted to the behaviour and analysis of the beam-column joint. A knowledge of the biaxial stress-strain relationship is essential in the evaluation of the stress-block parameters. Previous
studies have introduced a number of parameters that take into account the biaxial effect. The following expression has been adopted for the biaxial stress-strain relationship:

\[
f = \frac{\varepsilon}{(1-\mu\alpha)(A + B\varepsilon''^n)}
\]  

(5.107)

where:

\[
f = \text{concrete stress in the direction considered, } N/mm^2.
\]

\[
\varepsilon = \text{concrete strain in the direction of concrete stress, } f, N/mm^2.
\]

\[
\mu = \text{Poisson's ratio.}
\]

\[
\alpha = \text{ratio of the principle stress in the orthogonal direction to the principle stress in the direction considered.}
\]

To obtain values for coefficients \( A, B \) and \( n' \), certain criteria must be satisfied concerning the conditions at zero concrete stress and maximum strain, \( \varepsilon_0 \). These coefficients are determined in accordance with the following:

\[
A = \frac{1}{E_c}
\]  

(5.108)

\[
B = \frac{R-(1-\mu\alpha)}{\varepsilon_0''(1-\mu\alpha)E_c}
\]  

(5.109)

\[
R = \frac{E_c}{E_0}
\]  

(5.110)

\[
n' = \frac{R}{R-(1-\mu\alpha)}
\]  

(5.111)

Substituting the above coefficients in equation (5.107), equation (5.107) becomes:
The equation for the biaxial stress-strain relationship can be reduced to:

\[
f = \frac{E \varepsilon}{(1 - \mu \alpha) + [R - (1 - \mu \alpha)](\varepsilon \varepsilon_0)}
\]

Equations (5.112) and (5.113) were proposed by Nirjar [1] to describe the biaxial stress-strain relationship. The values for Poisson's ratio assumed to be 0.2, but is influenced by many factors. Equation (5.113) can be further simplified by using properties obtained from the uniaxial stress-strain curve such as concrete strain at maximum stress, the maximum concrete stress and finally the Modulus of Elasticity of the concrete.

The concrete biaxial stress-strain relationship can therefore be established and utilising the knowledge obtained previously from the preparatory stage and renew stage, the final stress-strain relationship can be obtained using similar procedure to that used in section 5.4.4.1.
6.1. INTRODUCTION

The aim of this chapter is to review the relevant theoretical and experimental results analysed by Nirjar [1] with respect to the moment-curvature relationships and the updated information provided by Park and Ruitong [99]. Neural network implementations are then carried out to assess any benefits provided by this technique. The author's study investigates in detail the unconfined section and provides a procedure for analysing the behaviour of confined sections within the framework of neural networks. Taking advantage of computational techniques, it is proposed to establish a computational procedure which can solve design formulae and provide realistic information on the behaviour of the structural components involved. The effects of several variables were studied, and assumptions and limitations imposed to examine the problem in detail. All computation manipulations were carried out by programming the relevant equations in FORTRAN language to collect the training and test data necessary for the prediction process. The network architecture, topologies and neurodynamics are then presented to provide a clear picture of the network's structure with the instrumentation of the confusion matrices, classification rates, network weights as well as the layer's learning rates. The relevant equations were obtained from the process of Flashcode
which is an integrated part of the NeuralWorks® Professional software [17] and converts a trained and tested network to a 'C' code subroutine.

The Reinforced Concrete structure is by far the most commonly used form of construction used worldwide even though concrete is relatively weak in tension resulting in a type of brittle failure. When the concrete cracks the interaction between the cracked concrete and reinforcing steel can cause highly non-linear behaviour which produces complication in the ultimate analysis. The acceptance and widespread use of the ultimate strength design method has introduced the need to consider additional stresses in the compression zone of the section as a result of non-linear analysis of the stress-strain relationship which is emphasised with the use of slimmer sections. The tendency to design smaller sections produces larger deflection in the members and greater attention must now be paid to the serviceability states to control excessive deflections on cracking. The ability of the structural member to undergo deformation without excessive loss of flexural capacity in the member is indication of the ductility of the member. The flexural strength and the ductility of unconfined reinforced concrete beams mainly depends on the tensile reinforcement ratio and the compression reinforcement ratio.

The curvature and moment characteristics have been investigated at the pre-cracking stage of the load-deflection curve. This involves a linear relationship defining elastic behaviour. The post-cracking stage ends with the pre-cracking region and the initiation of the visual cracking. The moment-curvature curve is constructed at all stages to describe the uncracked, cracked, yield and finally ultimate behaviour of the reinforced concrete cross-section under applied load. The behaviour of a joint is entirely related to the members that compose it, and these members play a major role in predicting the behaviour of the joint. At every stage, the flexural members exhibit a specific behaviour stage which provides a relationship which is unique to that stage. For the flexural member under investigation, it is necessary to
assume a prismatic area of concrete with constant areas of reinforcement and characteristics for deformations and curvatures which are uniform along the span length.

The variables and parameters that are required to satisfy certain equations can be predicted by implementing the neural networks technique. The experimental data provided by Nirjar [1] were evaluated and tested by integrating the experimental data into neural networks as a reliability test to validate the neural network predictive procedure. This new approach attempts to characterise the mode of failure of the cross-section at ultimate load by specifying values of curvature and moment at specific stages using an entirely new approach employing the implementation of a predictive procedure.

6.2. RADIUS OF CURVATURE

The radius of curvature of a reinforced concrete member varies along the member, particularly between cracks where the concrete may carry some tension. This variation in the radius of curvature has an effect on the location of the neutral axis, the concrete strain in the compression zone and the strain in the tensile steel. Determination of the radius of curvature is essential to study the relationship between curvature and the moment. Figure 6.1 shows a simply supported beam together with curvature of an element and the strain distribution.
The basic assumptions hold true, namely that plane sections remain plane before and after bending, and the radius of curvature is determined from Figure 6.1 as follows:

For small $\Delta \phi : ab \equiv$ straight line:

$$\therefore \tan \Delta \phi \equiv \Delta \phi$$

(6.1)

For a small element of length '$dx'$ and from triangle $oab$:

$$\tan \Delta \phi = \frac{\Delta x}{R_c} \Rightarrow \Delta \phi = \frac{\Delta x}{R_c}$$

(6.2)

$$\frac{\Delta \phi}{\Delta x} = \frac{1}{R_c}$$

Let $\Delta x$ approach zero i.e. $\Delta x \to 0$:

$$\frac{d\phi}{dx} = \frac{1}{R_c}$$

(6.3)

From triangle $bde$:
\[
\frac{d\phi}{bd} = \frac{\varepsilon_s dx}{xd} \Rightarrow \frac{d\phi}{xd} = \frac{\varepsilon_s}{xd}
\]  
(6.4)

From equation (6.3): \( \frac{d\phi}{dx} = \frac{1}{R_c} \)

\[
\Rightarrow \frac{1}{R_c} = \frac{\varepsilon_s}{xd} \quad (6.5)
\]

Since \( \varepsilon_s = \frac{f_s}{E} \)

\[
\frac{1}{R_c} = \frac{f_s}{x d} = \frac{f_s}{E(xd)}
\]  
(6.6)

Also \( f_s = \frac{M(xd)}{I} \)

\[
\therefore \frac{1}{R_c} = \frac{M(xd)}{E(xd)} = \frac{M}{E I} \Rightarrow \frac{1}{R_c} = \frac{M}{E I}
\]  
(6.7)

Mathematically the radius of curvature can be obtained from relationship shown in Figure 6.2 as:

\[
R_c = \left[1 + \left(\frac{dy}{dx}\right)^2\right]^{1/2} \frac{d^2 y}{dx^2}
\]  
(6.8)

The slope \( (dy/dx) \) of the elastic curve for a beam is small. Neglecting the effect of \( (dy/dx)^2 \), equation (6.8) reduces to:

\[
R_c = \frac{1}{d^2 y / dx^2}
\]  
(6.9)

Or:

\[
\frac{1}{R_c} = \frac{d^2 y}{dx^2}
\]  
(6.10)

So that the rate of change of slope is given by:

\[
\frac{d}{dx} \left(\frac{dy}{dx}\right) = \frac{d}{dx} (\phi) = \frac{d\phi}{dx}
\]  
(6.11)
from equation (6.7):

\[
\frac{1}{R_c} = \frac{M}{EI} \Rightarrow \frac{d^2y}{dx^2} = \frac{M}{EI}
\] (6.12)

Therefore, equations (6.7) and (6.12) provide the computation of the radius of curvature for a beam element.

Figure 6.2. Mathematical Representation of the Radius of Curvature.

where:

\( \phi \) = curvature (rotation per unit length of member).

\( R_c \) = radius of curvature.

\( M \) = applied moment at the cross-section.

\( f_g \) = general stress at a point.

\( \varepsilon_g \) = general strain at a point.

\( xd \) = distance of the neutral axis from the compression face.
6.3. RELATIONSHIP BETWEEN MOMENT AND CURVATURE

Numerous approaches have been developed to study the relationship between moment and curvature. A uniaxial load, moment and curvature procedure was developed by Pfang, Siess, and Sozen [100]. They described a procedure to evaluate the relationship between uniaxial load, moment and curvature based on using a tool for the analytical procedure. A similar procedure was adopted by Kroenke, Gutzwiller and Lee [101] who also considered the effect of unsymmetrical reinforcement placement and the inclusion of strain hardening. An incremental-type method for the determination of the load-deformation relationship was developed by El-Metwally and Chen [102]. This method had been experimentally verified and could be applied to a non-linear analysis of frame structures.

A relationship between load-deformation and moment-curvature can be determined by applying the principle of equilibrium of the internal forces and compatibility of the strains. This approach will provide an acceptable estimate of the curvature and moments. An evaluation of the moment-curvature relationship is essential for the analysis of the deformation behaviour of a beam-column joint since the load-deformation and moment-rotation characteristics are evaluated on this basis. It is also more desirable to determine moment, rather than load, at each stage in the analysis.

There are usually three stages of behaviour which the member undergoes during application of the load. Figure 6.3 shows typical idealised load-deflection and moment-curvature relationships for the loading given in Figure 6.1.
Figure 6.3. Idealised Load-Deflection and Moment-Curvature Relationships.

The loading stages are characterised by:

(a) An elastic stage between \( O \) and \( a \): the stress in the concrete member is proportional to the strain up to point 'a' and the member is un-cracked. This stage is usually referred to as the 'un-cracked stage'. The tensile reinforcement at this stage is basically inactive.

(b) The yield stage between \( a \) and \( b \): as the load increases the tensile steel starts to yield and cracks appear in the tension zone. This stage is known as the 'yield stage'.

(c) The ultimate stage between \( b \) and \( c \): the member behaves inelastically and is extended until the member fails. This stage is known as the 'ultimate stage'.

As shown in Figure 6.3(b), the moment-curvature diagram follows the same sequence as the load-deflection response shown in Figure 6.3(a). Cracking of the concrete occurs at point 'a' and the tensile steel starts to yield at point 'b'. Beyond point 'b' the concrete carries the load in compression zone and reaches its the ultimate moment at point 'c' where failure of the concrete occurs. This is characteristic for an under-reinforced concrete member.
The analysis adopted in this study is based on a direct sectional analysis which applies the principles of strain compatibility and equilibrium of the internal forces as adopted by Mattock [10] and Furlong [103]. The assumption and limitations involved in carrying out this analysis are as follows:

(a) Plane sections normal to the neutral axis remain planes before and after bending.

(b) The ultimate strain, \( \varepsilon_u \), can be determined either from equation (5.20) in chapter 5 or from the value of 0.003 assumed by the ACI Code 318-89 [81].

(c) The concrete stress block is as described in section 5.5 of chapter 5.

(d) No slippage between the concrete and the steel occurs so that the strains in the steel and adjoining concrete are equal.

(e) There is an elastic distribution of stress at initial yielding of the tensile reinforcement so that the depth of the neutral axis can be determined by elastic theory and the maximum concrete compressive stress, located at the compression face, reduces linearly to zero at the neutral axis as shown in Figure 6.4(c).

(f) The compression steel yields at ultimate, \( \varepsilon_s = \varepsilon_y \).

(g) Concrete strength in the tension zone is neglected.
6.3.1. Cracking Moment and Curvature

Cracking is an important serviceability limit which must be considered in the analysis and design of reinforced concrete members. Cracking is not allowed in water retaining structures while it is permissible in other structures under certain limitations. As shown in Figure 6.3(b) the relationship between moment and curvature at this stage is linear up to point 'a'. Beyond point 'a' cracks start to appear in the tension zone. Prior to cracking the curvature, $\phi_{un}$, of the un-cracked section can be determined from:

$$\phi_{un} = \frac{M}{E_c I}$$  \hspace{1cm} (6.13)

where:

$$\phi_{un} = \text{un-cracked curvature, mm}^{-1}.$$  

$$E_c = \text{Modulus of Elasticity of concrete, N/mm}^2.$$  

$$I = \text{moment of inertia of the gross cross section equal to } b \frac{H^3}{12},$$  

where its centroidal axis is in the plane of bending, mm$^4$.  

$$H = \text{total depth of the cross-section, mm}.$$  

Figure 6.4. Elastic Stress Distribution Condition at Yield of the Tensile reinforcement.
\[ b = \text{width of the cross-section, } mm. \]

The Modulus of Elasticity of concrete can be determined from section 5.4.1 (as proposed by Nirjar [1]) or from section 5.4.1.1 of chapter 5 as predicted by the neural network. The moment of inertia of the gross section is approximated by neglecting the effect of the steel and considering only the gross area of the concrete. This provides a good approximation for determining the flexural rigidity \( (E_c I) \). Prior to cracking the flexural rigidity is the major factor in determining the deflection of the member.

Due to the low tensile strength of the concrete the cracking moment is obtained when the stress in the extreme fibres of the concrete in the tension zone reach the modulus of rupture of concrete. Hence the cracking moment is determined from:

\[ M_{cr} = f_t S \]  \hspace{1cm} (6.14)

where:

\[ M_{cr} = \text{cracking moment at the cross-section, } N\text{-}mm. \]

\[ f_t = \text{tensile strength of concrete expressed as the modulus of rupture, } N/mm^2. \text{ (As predicted by neural network in section 5.3.1.)} \]

\[ S = \text{section modulus equal to } b\frac{H^2}{6}, \text{ mm}^3. \]

The modulus of rupture by the neural network is given in section 5.3.1 by equations (5.6) and (5.7) or from predicted Figure 5.2 of chapter 5. Table 6.1 provides a comparison of the values of the cracking moment computed using the formulae proposed by Nirjar [1] and by using neural networks.
Table 6.1. Comparisons of Cracking Moments.

<table>
<thead>
<tr>
<th>Specimens</th>
<th>$f_c \text{ (N/mm}^2\text{)}$</th>
<th>Nirjar [1] $M_{cr} \times 10^6 \text{ N-m}$</th>
<th>Neural Networks $M_{cr} \times 10^6 \text{ N-m}$</th>
<th>Errors $%Error = (1 - \frac{\text{Network}}{\text{Desired}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N/\text{mm}^2$</td>
<td>$N/\text{mm}^2$</td>
<td>$N/\text{mm}^2$</td>
<td>$N/\text{mm}^2$</td>
</tr>
<tr>
<td>$NT_{31}$</td>
<td>45</td>
<td>851.63</td>
<td>750.00</td>
<td>820.31</td>
</tr>
<tr>
<td>$NT_{32}$</td>
<td>35</td>
<td>751.06</td>
<td>750.00</td>
<td>742.19</td>
</tr>
<tr>
<td>$NS_{30}$</td>
<td>30</td>
<td>695.35</td>
<td>480.00</td>
<td>703.13</td>
</tr>
<tr>
<td>$NT_{33}$</td>
<td>25</td>
<td>634.77</td>
<td>600.00</td>
<td>625.00</td>
</tr>
<tr>
<td>$NT_{34}$</td>
<td>20</td>
<td>567.75</td>
<td>570.00</td>
<td>546.88</td>
</tr>
</tbody>
</table>

The computational analysis for the load-deflection curve is obtained by applying the equilibrium conditions shown in Figure 6.5 and the adopted modulus of rupture.

Table 6.1 shows that a large error occurs for specimen $NS_{30}$ in the experiment results.
The load, $P_{cr}$, at cracking can be determined from:

$$P_{cr} = \frac{f_b H^2}{6L}$$

(6.15)

The deflection, $\Delta_{cr}$, at cracking is determined by:

$$\Delta_{cr} = \phi_{cr} \frac{L \times 2L}{2 \times 3}$$

(6.16)

The cracking curvature, $\phi_{cr}$, is determined as:

$$\phi_{cr} = \frac{M}{E_c I} = \left(\frac{P_{cr} \times L}{E_c I}\right) \frac{L}{2} = \left(\frac{f_b H^2}{6L}\right) \frac{L^2}{2E_c I}$$

(6.17)
So that the deflection, $\Delta_{cr}$, at the point of loading is given as:

$$\Delta_{cr} = \frac{P_c I^3}{3E_c J}$$  \hspace{1cm} (6.18)

### 6.3.2. Yield Moment and Curvature

As the load increases, the cracks forming in the tension zone propagate upward toward the neutral axis. The concrete in the tension zone is inactive and the tensile steel resists the entire tension. As illustrated in Figure 6.3(b) of the moment-curvature relationship, point 'b' is the point on the curve which defines yielding of the tension reinforcement and an increase in the deflection while the applied load remains nearly constant. As long as the cross-section is under-reinforced the tensile reinforcement reaches yield before the concrete crushes. The yield curvature, $\phi_y$, generally defined as the curvature at which the tensile reinforcement reaches its yield point stress, is determined from Figure 6.4(b) by strain compatibility as follows:

$$\phi_y = \frac{\varepsilon_y}{d - xd} = \frac{f_y}{E_s(1-x)d}$$  \hspace{1cm} (6.19)

For a doubly reinforced beam with a tensile reinforcement ratio of

$$\rho = \frac{A_s}{bd}$$

and compressive reinforcement ratio of

$$\rho' = \frac{A_c}{bd}$$  \hspace{1cm} (6.20)

the following expression can be derived:

$$x = \sqrt{(\rho + \rho')^2 n^2 + 2n(\rho + \rho') - n(\rho + \rho')}$$  \hspace{1cm} (6.22)

where:

- $n$ = modular ratio equals to $\frac{E_s}{E_c}$.
- $E_s$ = Modulus of Elasticity of steel, $N/mm^2$. 
\( d \) = effective depth of the cross-section, \( \text{mm} \).

\( ad \) = distance from the compression face of member to the centroid of compression steel, \( \text{mm} \).

\( A_s \) = area of tension steel reinforcement, \( \text{mm}^2 \).

\( A_c \) = area of compression steel reinforcement, \( \text{mm}^2 \).

\( \rho \) = tensile reinforcement ratio of steel.

\( \rho' \) = reinforcement ratio of compressive steel.

The final expression for the curvature at yield, \( \phi_y \), is then established by combining equation (6.19) and (6.22) to obtain:

\[
\phi_y = \frac{f_y}{E_s [n(\rho + \rho') - \sqrt{(\rho + \rho')^2 n^2 + 2n(\rho + \rho'a) + 1}]d}
\] (6.23)

The strain in the compression steel is determined from Figure 6.4(b) as:

\[
\varepsilon_c = (xd - ad) \phi_y = d(x - a) \phi_y
\] (6.24)

where the compression steel does not yield i.e. \( \varepsilon_c < \varepsilon_y \), the compressive force due to compression steel is \( C_c \) where:

\[
C_c = \varepsilon_c E_s A_c
\] (6.25)

where the compression steel yields, i.e. \( \varepsilon_c \geq \varepsilon_y \), the compressive force due to compression steel is \( C_c \) where:

\[
C_c = A_c f_y
\] (6.26)

Applying the internal equilibrium:

\[
T = C_c + C_c
\] (6.27)

so that:
The internal resisting moment of the critical cross-section is therefore obtained by taking the moments of the internal forces of the compressive concrete and the compressive reinforcement about the tensile reinforcement:

\[ M = C_c (d - \frac{xd}{3}) + C_s (d - ad) \]  \hspace{1cm} (6.29)

Substituting for \( C_c \) and \( C_s \) into equation (6.29) the yield moment, \( M_y \), is given as:

\[ M_y = (A_c f_y - A_s f_s)(d - \frac{xd}{3}) + A_s f_s (d - ad) \]  \hspace{1cm} (6.30)

The stress in the reinforcement compression, \( f_s \), can be obtained from Figure 6.4 and from the compression strain obtained earlier:

\[ f_s = E_s \varepsilon_s = E_s \varepsilon_y \frac{(x - a)}{(1 - x)} \]  \hspace{1cm} (6.31)

When the stress in the compression reinforcement reaches the yield stress

\[ (f_s = f_y) \]

and, \( C_s = f_y A_s \) \hspace{1cm} (6.32)

Previous computations provided expressions which relate the curvature and moment at the yield stage for a doubly reinforced beam.

For a singly reinforced beam, the expression for ratio 'x' can be simplified to:

\[ x = \sqrt{(\rho^2 n^2 + 2n\rho)} - n\rho \]  \hspace{1cm} (6.33)

The distance of the neutral axis from the compressive face becomes:

\[ xd = (\sqrt{(\rho^2 n^2 + 2n\rho)} - n\rho) d \]  \hspace{1cm} (6.34)

The curvature at yield is simplified to:
\[ \phi_y = \frac{f_y}{E_s[n\rho - \sqrt{\rho^2 n^2 + 2n\rho} + 1]} \tag{6.35} \]

and the moment for a singly reinforced beam becomes:

\[ M_y = A_s f_y (1 - \frac{x}{3}) d \tag{6.36} \]

Analysis at the yield stage to compute the deflection is achieved by applying the moment-area theorem to the beam. The deflection is first computed for the beam without considering the column stub, \( h_c \), as shown in Figure 6.6(b). The deflection is then computed taking into account the column stub, \( h_c \), as shown in Figure 6.6(c).
From the moment diagram given in Figure 6.6(d), the deflection at the load point is found from:

\[ \Delta_y = \frac{1}{2} \phi_y (L) \left( \frac{2}{3} L \right) \]  
(6.37)

i.e. \( \Delta_y = \frac{1}{3} \phi_y L^2 \)  
(6.38)

From the moment diagram given in Figure 6.6(d), considering the column stub:
\[
\Delta_y = \frac{1}{2} \phi_y (L) \left( \frac{2}{3} L \right) + \phi_y \frac{h_c}{2} \left( \frac{1}{2} \right) \frac{4L + h_c}{4}
\]  
(6.39)

i.e. \[
\Delta_y = \frac{1}{24} \phi_y (8L^2 + 6Lh_c + h_c^2)
\]  
(6.40)

Thus the yield curvature, the yield moment and the deflection at yield for a doubly reinforced beam can be computed from equations (6.23), (6.30) and (6.40) respectively.

6.3.3. Preparatory Stage for Yield Moment and Curvature

For the preparatory stage, the procedure adopted in this study to determine the moment curvature relationship is to select a network of (6, 40, 30, 2). This represents the number of input PE's, the hidden PE's in the first layer, the hidden PE's in the second hidden layer and output layer respectively. The RMS was set to 0.02% and provided satisfactory results. The EDBD learning rule adopted included a heuristic adjustment to the momentum term set to 0.4. The learning rates were set to 0.3 for the first hidden layer, 0.25 for the second hidden layer and 0.15 for the output layer. Input and desired patterns were presented to the network randomly in the training file and sequentially in the test file. The training and test files were generated with the FORTRAN program by selecting a possible combination of 34 specimens. Three concrete cylinder strengths, 20, 30 and 45 N/mm² were used, with 2 different bar arrangements (2 and 4) as well as two different bar diameters, 6 and 12 mm, to generate the data. The total number of combinations obtained were therefore 408, of which only 243 patterns were used for training and 29 patterns for the testing file. The remainder did not conform to the Code of Practice. The preliminary training and test pattern values obtained from a numerical analysis for the curvature, \( \phi_y \), were very small, as obtained from the FORTRAN program. It was observed that to enhance the network performance, the values of the curvature were magnified by \( 10^8 \) and the steel percentage 'p' by \( 10^4 \). This provided a better network performance. The results predicted by the neural networks for the yield curvatures and moments are in
close agreement with those obtained by Nirjar [1]. These results are tabulated in Table 6.2 together with the percentage errors.

The predicted formulae obtained by neural networks for the curvature can be presented in the form of:

\[ y_{78,y}^{\text{out},p} = x_{78,y}^{\text{out},p} * (478.41) + 2206.8 \]  
(6.41)

\[ x_{78,y}^{\text{out},p} = \tanh\left(\sum x_{78,y}^{p}\right) \]  
(6.42)

The moment at yield obtained can be expressed as:

\[ y_{79,y}^{\text{out},p} = x_{79,y}^{\text{out},p} * (2824.056) + 3877.195 \]  
(6.43)

\[ x_{79,y}^{\text{out},p} = \tanh\left(\sum x_{79,y}^{p}\right) \]  
(6.44)

in which \( y_{78,y}^{\text{out},p} \) and \( y_{79,y}^{\text{out},p} \) are the scaled curvature and moment at yield respectively for the preparatory stage. The full formulae are provided in a disk at the end of the thesis while the network architecture, topology and neurodynamics are shown in Figure 6.7 for the trained and test network. The network ceased learning at a 47479 cycles with excellent correlations of 1.0 for both the trained and test networks in the confusion matrices. The classification rate resulted in 1.0 which indicates that the network correctly classified 100% of the good and did not misclassify any other categories.

Thus, the curvature and moment at the yield stage can be computed from equations (6.23) and (6.30) respectively, as proposed by Nirjar [1], or alternatively by neural network for a reinforced beam as proposed by equations (6.41) and (6.43) respectively.
Table 6.2. Comparisons of Moments and Curvatures at Yield by Nirjar [1] and by Neural Networks for the Preparatory Stage.

<table>
<thead>
<tr>
<th>Specimens</th>
<th>$f'_e$</th>
<th>$\rho$</th>
<th>Nirjar [1]</th>
<th>Neural Networks</th>
<th>%Error = Abs($\frac{\text{Network}}{\text{Desired}}$) * 100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N/mm²</td>
<td>(%)</td>
<td>Computed</td>
<td>Computed</td>
<td>Experimental</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\phi_y \times 10^{-8}$</td>
<td>$M_y$</td>
<td>$M_y$</td>
</tr>
<tr>
<td>$NN_1$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2880</td>
</tr>
<tr>
<td>$NN_2$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2940</td>
</tr>
<tr>
<td>$NN_3$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2850</td>
</tr>
<tr>
<td>$NN_4$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2790</td>
</tr>
<tr>
<td>$NN_5$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2760</td>
</tr>
<tr>
<td>$NN_6$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2760</td>
</tr>
</tbody>
</table>
Table 6.2. (continued)

<table>
<thead>
<tr>
<th></th>
<th>(NM_7)</th>
<th>(NM_8)</th>
<th>(NM_9)</th>
<th>(NM_{10})</th>
<th>(NO_{11})</th>
<th>(NO_{12})</th>
<th>(NO_{13})</th>
<th>(NO_{14})</th>
<th>(NP_{15})</th>
<th>(NP_{16})</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
<td>(30.0)</td>
</tr>
<tr>
<td></td>
<td>(1.28)</td>
<td>(1.28)</td>
<td>(1.28)</td>
<td>(1.28)</td>
<td>(1.28)</td>
<td>(1.28)</td>
<td>(1.28)</td>
<td>(1.28)</td>
<td>(0.72)</td>
<td>(2.00)</td>
</tr>
<tr>
<td></td>
<td>(2120)</td>
<td>(2120)</td>
<td>(2120)</td>
<td>(2120)</td>
<td>(2120)</td>
<td>(2120)</td>
<td>(2120)</td>
<td>(2120)</td>
<td>(1900)</td>
<td>(2360)</td>
</tr>
<tr>
<td></td>
<td>(2829)</td>
<td>(2829)</td>
<td>(2829)</td>
<td>(2829)</td>
<td>(2829)</td>
<td>(2829)</td>
<td>(2829)</td>
<td>(2829)</td>
<td>(1637)</td>
<td>(4310)</td>
</tr>
<tr>
<td></td>
<td>(2880)</td>
<td>(2880)</td>
<td>(2820)</td>
<td>(2820)</td>
<td>(2820)</td>
<td>(2760)</td>
<td>(2760)</td>
<td>(2760)</td>
<td>(1650)</td>
<td>(4200)</td>
</tr>
<tr>
<td></td>
<td>2062.99</td>
<td>2062.99</td>
<td>2062.99</td>
<td>2062.99</td>
<td>2062.99</td>
<td>2062.99</td>
<td>2062.99</td>
<td>2062.99</td>
<td>1881.73</td>
<td>2445.08</td>
</tr>
<tr>
<td></td>
<td>2615.82</td>
<td>2615.82</td>
<td>2615.82</td>
<td>2615.82</td>
<td>2615.82</td>
<td>2615.82</td>
<td>2615.82</td>
<td>2615.82</td>
<td>1632.30</td>
<td>5067.66</td>
</tr>
<tr>
<td></td>
<td>2.69</td>
<td>2.69</td>
<td>2.69</td>
<td>2.69</td>
<td>2.69</td>
<td>2.69</td>
<td>2.69</td>
<td>2.69</td>
<td>0.96</td>
<td>3.60</td>
</tr>
<tr>
<td></td>
<td>7.54</td>
<td>7.54</td>
<td>7.54</td>
<td>7.54</td>
<td>7.54</td>
<td>7.54</td>
<td>7.54</td>
<td>7.54</td>
<td>0.29</td>
<td>17.58</td>
</tr>
<tr>
<td></td>
<td>9.17</td>
<td>8.22</td>
<td>7.24</td>
<td>7.24</td>
<td>5.22</td>
<td>5.22</td>
<td>5.22</td>
<td>10.11</td>
<td>1.07</td>
<td>20.66</td>
</tr>
</tbody>
</table>

210
Table 6.2. (continued)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>2540</th>
<th>4312</th>
<th>5260</th>
<th>2563.34</th>
<th>5870.79</th>
<th>0.92</th>
<th>36.15</th>
<th>11.61</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP17</td>
<td>30.0</td>
<td>2.55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NP18</td>
<td>30.0</td>
<td>2.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NQ19</td>
<td>30.0</td>
<td>0.72</td>
<td>1900</td>
<td>1637</td>
<td>1680</td>
<td>1881.73</td>
<td>1632.30</td>
<td>0.96</td>
<td>0.29</td>
<td>2.84</td>
</tr>
<tr>
<td>NQ20</td>
<td>30.0</td>
<td>2.00</td>
<td>2360</td>
<td>4310</td>
<td>4200</td>
<td>2445.08</td>
<td>5067.66</td>
<td>3.60</td>
<td>17.58</td>
<td>20.66</td>
</tr>
<tr>
<td>NQ21</td>
<td>30.0</td>
<td>2.55</td>
<td>2540</td>
<td>5412</td>
<td>5160</td>
<td>2563.34</td>
<td>5870.79</td>
<td>0.92</td>
<td>8.48</td>
<td>13.78</td>
</tr>
<tr>
<td>NQ22</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5690</td>
<td>2595.78</td>
<td>6105.05</td>
<td>2.78</td>
<td>2.80</td>
<td>7.29</td>
</tr>
<tr>
<td>NR23</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5880</td>
<td>2589.51</td>
<td>6057.88</td>
<td>3.01</td>
<td>3.55</td>
<td>3.03</td>
</tr>
<tr>
<td>NR24</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5820</td>
<td>2589.51</td>
<td>6057.88</td>
<td>3.01</td>
<td>3.55</td>
<td>4.09</td>
</tr>
<tr>
<td>NR25</td>
<td>30.0</td>
<td>2.99</td>
<td>267</td>
<td>6281</td>
<td>5760</td>
<td>2589.51</td>
<td>6057.88</td>
<td>3.01</td>
<td>3.55</td>
<td>5.17</td>
</tr>
<tr>
<td>NR26</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5760</td>
<td>2589.51</td>
<td>6057.88</td>
<td>3.01</td>
<td>3.55</td>
<td>5.17</td>
</tr>
</tbody>
</table>
Table 6.2. (continued)

<p>| | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NS27</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2700</td>
<td>2062.99</td>
<td>2615.82</td>
<td>2.69</td>
<td>7.54</td>
<td>3.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS28</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2730</td>
<td>2062.99</td>
<td>2615.82</td>
<td>2.69</td>
<td>7.54</td>
<td>4.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS29</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2700</td>
<td>2062.99</td>
<td>2615.82</td>
<td>2.69</td>
<td>7.54</td>
<td>3.12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS30</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2760</td>
<td>2062.99</td>
<td>2615.82</td>
<td>2.69</td>
<td>7.54</td>
<td>5.22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT31</td>
<td>40</td>
<td>1.28</td>
<td>204</td>
<td>2857</td>
<td>2910</td>
<td>1985.51</td>
<td>2641.02</td>
<td>2.67</td>
<td>7.56</td>
<td>9.24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT32</td>
<td>35</td>
<td>1.28</td>
<td>2070</td>
<td>2846</td>
<td>2850</td>
<td>2027.80</td>
<td>2623.67</td>
<td>2.04</td>
<td>7.81</td>
<td>7.94</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT33</td>
<td>25</td>
<td>1.28</td>
<td>2170</td>
<td>2812</td>
<td>2790</td>
<td>2107.50</td>
<td>2607.58</td>
<td>2.88</td>
<td>7.27</td>
<td>6.54</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT34</td>
<td>20</td>
<td>1.28</td>
<td>2250</td>
<td>2790</td>
<td>2700</td>
<td>2161.82</td>
<td>2623.14</td>
<td>3.92</td>
<td>5.98</td>
<td>2.85</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 6.7. Curvature and Moment at Yield for the Preparatory Stage.
6.3.4. Moment and Curvature Relationship at Ultimate

This is the third stage before failures occur. This region extends from point 'b' to point 'c' on the moment-curvature diagram of Figure 6.3(b), which is defined by yield of the tensile reinforcement, an increase in cracking and finally crushing of the concrete in the compression zone. The ultimate capacity of a member is therefore reached with further increase in load. The crack becomes visible and the curvature increases. The ultimate curvature, \( \phi_u \), is defined as the curvature at which the limiting value of concrete strain is attained in the extreme concrete fibre. The analysis for this stage must take into consideration the following assumptions:

(a) The compressive steel reinforcement has yielded, \( \varepsilon_c \geq \varepsilon_y \).

(b) The concrete compressive stress is inelastic and the stress distribution is as shown in Figure 6.8(c).

(c) The parameters obtained in chapter 5, namely, the ultimate concrete strain, \( \varepsilon_u \), the stress distribution in the compression zone, \( f \), the average concrete compressive stress, \( f_{av} \), and other relevant parameters predicted by neural network were adopted.
(d) The idealised **tri-linear** stress-strain relation shown in Figure 6.9, can be assumed to include the effect of strain hardening of the steel, to represent the moment-curvature relationship and to simplify the deflection analysis. The reinforcement properties at strain hardening and ultimate used by Nirjar [1] viz 0.024 and 0.124 respectively, and the yield and ultimate steel stresses used viz 304 and 454 N/mm$^2$ respectively can be used.

The stress-strain relationship beyond strain hardening is represented by the equation (6.45):

\[
fs = fy + \left( \frac{fu - fy}{\epsilon_u - \epsilon_h} \right)(\epsilon_s - \epsilon_y)
\]  

(6.45)

where:

\[
\begin{align*}
fs &= \text{ultimate steel stress, N/mm}^2. \\
fy &= \text{yield steel stress, N/mm}^2. \\
\epsilon_u &= \text{strain hardening}. \\
\epsilon_h &= \text{strain hardenig}. \\
\epsilon_s &= \text{strain at ultimate}. \\
\epsilon_y &= \text{strain at yield}. \\
0 &= \text{strain at zero stress}.
\end{align*}
\]

![Figure 6.9. An Idealised Tri-linear Steel Stress-Strain Relationship.](image)

To determine the distance of the neutral axis from the compression face at ultimate, the following equilibrium conditions are considered:
The ultimate compressive concrete force, $C_u$, is given by:

$$C_u = f_{av} b x_u d$$

(6.48)

so that equation (6.47) becomes:

$$A_s f_s = f_{av} b x_u d + A_s f_y$$

(6.49)

and distance of the neutral axis from the compression face at ultimate is found from:

$$x_u d = \frac{A_s f_s - A_s f_y}{f_{av} b}$$

(6.50)

or:

$$x_u = \frac{\rho f_s - \rho f_y}{f_{av}}$$

(6.51)

Thus, the ultimate moment of resistance at the critical cross-section is given as:

$$M_u = C_u (d - x_u d + y d) + A_s f_y (d - y d)$$

(6.52)

Substituting for $C_u$ and $x_u$ into the above equation:

$$M_u = b d^2 [f_{av} x_u (1 - g) + \rho f_y (1 - a)]$$

(6.53)

where the value of 'g' can be obtained from:

$$g = x_u - y$$
The value, $x_u d$, is dependent on the condition as shown in Figure 6.8 of the tensile steel. From Figure 6.8(b), considering the triangle element, $MNT$, the curvature at ultimate is:

$$
\phi_u = \frac{\varepsilon_s + \varepsilon_u}{d}
$$

(6.54)

Also from Figure 6.8(b) using strain compatibility:

$$
\frac{x_u d}{d} = \frac{\varepsilon_u}{\varepsilon_s + \varepsilon_u}
$$

(6.55)

Substituting the value of '$x_u$' obtained in equation (6.51) into equation (6.55):

$$
\frac{\rho f_x - \rho f_y}{f_n} = \frac{\varepsilon_u}{\varepsilon_s + \varepsilon_u}
$$

(6.56)

and the tensile stress in the steel is found from:

$$
f_s = \frac{1}{\rho} \left( \rho f_y + \frac{f_n \varepsilon_u}{\varepsilon_s + \varepsilon_u} \right)
$$

(6.57)

Equation (6.57) defines the tensile stress in the tension reinforcement which is essential to determine the moment and curvature at ultimate for the critical cross-section. To compute the steel strain at a particular condition, the tri-linear curve of Figure 6.9 can be applied since the strain in the steel at any stage can be evaluated according to three the main conditions:

**CONDITION 1**: Tensile steel stress below the yield stress, $f < f_y$, then $\varepsilon_s = \frac{f_s}{E_s}$

Equation (6.57) becomes:

$$
f_s = \frac{1}{\rho} \left( \rho f_y + \frac{f_n \varepsilon_u}{(f_s/E_s) + \varepsilon_u} \right)
$$

(6.58)

**CONDITION 2**: Tensile stress equals the yield stress, $f_s = f_y$, then $\varepsilon_s = \varepsilon_y$

Equation (6.57) becomes:
\[ f_s = \frac{1}{\rho} \left( \rho f_y + \frac{f_{av}^\varepsilon u}{\varepsilon_y + \varepsilon_u} \right) \] (6.59)

**CONDITION 3:** Tensile stress above the yield stress, \( f_s > f_y \),

then \( \varepsilon_s = \varepsilon_y + \frac{f_s - f_y}{E_s} \)

Equation (6.57) becomes:

\[ f_s = \frac{1}{\rho} \left( \rho f_y + \frac{f_{av}^\varepsilon u}{(\varepsilon_y + \frac{f_s - f_y}{E_s}) + \varepsilon_u} \right) \] (6.60)

where, \( E_s = \frac{f_u - f_y}{\varepsilon_u - \varepsilon_h} \)

If the member is singly reinforced, equation (6.50) simplifies to:

\[ x_d = A_s f_y \frac{b}{f_{av}} \] (6.61)

If the tension reinforcement has not yielded, equation (6.61) becomes:

\[ x_d = A_s f_s \frac{b}{f_{av}} \] (6.62)

The ultimate moment equation then simplified to:

\[ M_u = bd^2 [f_{av} x_u (1 - g)] \] (6.63)

To determine the stress in the tensile steel at ultimate stage the tensile reinforcement ratio, \( \rho \), must be evaluated for the particular strain condition. This can be accomplished by applying the strain compatibility conditions and equilibrium of the internal forces:

\[ x_u = \frac{\varepsilon_u}{\varepsilon_y + \varepsilon_u} = \frac{\rho f_s}{f_{av}} \] (6.64)

so that the tensile reinforcement ratio can be evaluated from:
\[
\rho = \frac{1}{f_s} \left( \frac{f_u \varepsilon_u}{\varepsilon_s + \varepsilon_u} \right)
\]  

(6.65)

For the balanced condition, the reinforcement ratio, \( \rho_b \), is given by:

\[
\rho_b = \frac{1}{f_y} \left( \frac{f_u \varepsilon_u - (6.66)}{\varepsilon_y + \varepsilon_u} \right)
\]  

(6.66)

At the start of strain hardening \( \varepsilon_u = \varepsilon_h \), and the hardening strain ratio, \( \rho_h \), is determined from:

\[
\rho_h = \frac{1}{f_y} \left( \frac{f_u \varepsilon_u - (6.67)}{\varepsilon_h + \varepsilon_u} \right)
\]  

(6.67)

Note that three possible conditions can apply:

1. When \( \rho > \rho_b \) then \( f_s < f_y \), and the tensile stress in the steel is below the yield stress.
2. When \( \rho_b > \rho > \rho_h \) and \( f_s = f_y \), the tensile stress is equal to the yield stress.
3. When \( \rho_h > \rho \) then \( f_s > f_y \), and the tensile stress is at ultimate.

The ultimate curvature is found from the stress distribution:

\[
\phi_u = \frac{\varepsilon_s + \varepsilon_u}{d}
\]  

(6.68)

The value of the tensile strain, \( \varepsilon_u \), is found by taking into account the three following conditions:

1. When \( f_s < f_y \) and \( \varepsilon_s < \varepsilon_y \):

\[
\varepsilon_s = \frac{f_s}{E_s}
\]  

(6.69)

The ultimate curvature is computed from:

\[
\phi_u = \frac{f_s + \varepsilon_u E_s}{E_s d}
\]  

(6.70)
2. When \( f_s = f_y \) and \( \varepsilon_s = \varepsilon_y \):

The ultimate curvature is computed from:

\[
\phi_u = \frac{\varepsilon_u}{x_s d} \tag{6.71}
\]

3. When \( f_s > f_y \) and \( \varepsilon_s > \varepsilon_y \):

\[
\varepsilon_s = \varepsilon_y + \frac{f_s - f_y}{E_s} \tag{6.72}
\]

The ultimate curvature is computed from:

\[
\phi_u = \frac{\varepsilon_y [(f_s - f_y) + E_s]}{E_s d} \tag{6.73}
\]

For a beam with no compression reinforcement, the tensile steel stress, \( f_s \), can be obtained from:

\[
f_s = \frac{1}{\rho} \left( -\frac{f_{ov} \varepsilon_u}{\varepsilon_s + \varepsilon_u} \right) \tag{6.74}
\]

The ultimate moment at the critical section for a beam with no compression reinforcement becomes:

\[
M_u = bd^2 [f_{ov} x_u (1 - g)] \tag{6.75}
\]

The variables, \( \varepsilon_0 \), \( \varepsilon_u \) and \( f_{ov} \), which are used in the derivation of the above equations are obtained from chapter 5.

The relationships developed in the previous sections were based on the classical method which express the moment-curvature and load-deformation relationships for unconfined concrete through the distribution of the moment and curvature at different stages. The above evaluations were adopted by Nirjar [1].
6.3.5. Preparatory Stage for Ultimate Moment and Curvature

The preparatory stage for the moment-curvature relationship at the ultimate stage was carried using the same network architecture, topology and neurodynamics as that implemented at the yield stage in order to be consistent and compare the results. The training patterns contained 243 patterns while the test set contained 29. The 9th element of each training pattern was selected to represent the test data and was passed sequentially to the network to increase confidence in the network performance. The tested network was then checked to observe its reliability by passing the experimental data obtained by Nirjar [1]. The training and test data were generated using three different concrete strengths, 20, 30 and 45 $N/\text{mm}^2$, two different groups of bar reinforcement (2 and 4) and two different bars sizes (6 and 12).

Although a FORTRAN program was generated to calculate the curvature and moments for possible combinations of 408, a total of 272 only were used to conform with the limitations on minimum values of 'p', limitations on maximum allowable values of 'p', as well as the effect of strain hardening. Better convergence was observed and less cycles were obtained when the training and test input data were scaled to larger values especially for the curvature and values of steel ratio ,p, as well as the ultimate moments. The network selected for the moment-curvature relationship is shown in Figure 6.10 for both the trained and tested networks. The trained network converges at 40340 cycles with good convergence criteria of 0.02%. A test of 29 patterns was passed to the network sequentially to increase the confidence in the network performance since the networks had not been exposed to this before. As the network became functional, the reliability of the performance was checked by presenting the data results obtained by Nirjar [1] sequentially. Table 6.3 provides a summary of the curvature and moments obtained by neural networks and
Figure 6.10. Curvature and Moment at Ultimate for the Preparatory Stage.
Table 6.3. Comparisons of Moments and Curvatures at Ultimate by Nirjar [1] and by Neural Networks for the Preparatory Stage.

<table>
<thead>
<tr>
<th>Specimens</th>
<th>$f_c \ (N/\text{mm}^2)$</th>
<th>$\rho$ (%)</th>
<th>Nirjar [1]</th>
<th>Neural Networks</th>
<th>%Error = $\frac{\text{Abs}(1 - \text{Network}/\text{Desired})}{100}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\phi_u \times 10^{-8}$</td>
<td>$M_u$ W.R.T Experimental</td>
<td>$\phi_u \times 10^{-4}$</td>
<td>$M_u$ W.R.T (N)</td>
<td>(N) W.R.T (C)</td>
</tr>
<tr>
<td>$NN_1$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470 3002 3180</td>
<td>18213.31 2862.00</td>
<td>1.39 4.66 10.00</td>
</tr>
<tr>
<td>$NN_2$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470 3002 3240</td>
<td>18213.31 2862.00</td>
<td>1.39 4.66 1.67</td>
</tr>
<tr>
<td>$NN_3$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470 3002 3300</td>
<td>18213.31 2862.00</td>
<td>1.39 4.66 13.27</td>
</tr>
<tr>
<td>$NN_4$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470 3002 3000</td>
<td>18213.31 2862.00</td>
<td>1.39 4.66 4.60</td>
</tr>
<tr>
<td>$NN_5$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470 3002 2970</td>
<td>18213.31 2862.00</td>
<td>1.39 4.66 3.64</td>
</tr>
<tr>
<td>$NN_6$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470 3002 2940</td>
<td>18213.31 2862.00</td>
<td>1.39 4.66 2.65</td>
</tr>
</tbody>
</table>
Table 6.3. (continued)

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$NM_7$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3060</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>6.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_8$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3090</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>7.38</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_9$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3210</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>10.84</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_{10}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3270</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>12.48</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{11}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2790</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>2.58</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{12}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2820</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>1.49</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{13}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2910</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>1.65</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{14}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3060</td>
<td>18213.31</td>
<td>2862.00</td>
<td>1.39</td>
<td>4.66</td>
<td>6.47</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NP_{15}$</td>
<td>30.0</td>
<td>0.72</td>
<td>20100</td>
<td>1980</td>
<td>2040</td>
<td>20096.23</td>
<td>1876.98</td>
<td>0.02</td>
<td>12.04</td>
<td>7.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NP_{16}$</td>
<td>30.0</td>
<td>2.00</td>
<td>11860</td>
<td>4490</td>
<td>4500</td>
<td>9959.64</td>
<td>5240.63</td>
<td>16.02</td>
<td>12.31</td>
<td>16.46</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 6.3. (continued)

| NP_{17} | 30.0 | 2.55 | 9290   | 5532   | 5275   | 8288.42 | 6053.58 | 10.78 | 8.73  | 14.76 |
| NP_{18} | 30.0 | 2.99 | 7940   | 6301   | 5775   | 8308.97 | 6353.22 | 4.65  | 0.96  | 10.01 |
| NO_{19} | 30.0 | 0.72 | 20100  | 1980   | 2010   | 20096.23 | 1876.98 | 0.02  | 12.04 | 6.62  |
| NO_{20} | 30.0 | 2.00 | 11860  | 4490   | 4400   | 9959.64 | 5240.63 | 16.72 | 12.31 | 19.11 |
| NO_{21} | 30.0 | 2.55 | 9290   | 5563   | 5170   | 8288.42 | 6053.58 | 8.82  | 8.12  | 16.98 |
| NO_{22} | 30.0 | 2.99 | 7940   | 6301   | 5700   | 8308.97 | 6353.22 | 0.83  | 0.96  | 11.46 |
| NR_{23} | 30.0 | 2.99 | 7940   | 6301   | 6300   | 8259.65 | 6286.62 | 0.23  | 0.23  | 0.21  |
| NR_{24} | 30.0 | 2.99 | 7940   | 6301   | 6210   | 8259.65 | 6286.62 | 0.23  | 0.23  | 1.23  |
| NR_{25} | 30.0 | 2.99 | 7940   | 6301   | 5775   | 8259.65 | 6286.62 | 0.23  | 0.23  | 8.86  |
| NR_{26} | 30.0 | 2.99 | 7940   | 6301   | 5775   | 8259.65 | 6286.62 | 0.23  | 0.23  | 8.86  |
Table 6.3. (continued)

<table>
<thead>
<tr>
<th>$NT_{31}$</th>
<th>40</th>
<th>1.28</th>
<th>24000</th>
<th>3046</th>
<th>3090</th>
<th>20070.64</th>
<th>2992.96</th>
<th>1.74</th>
<th>2.09</th>
<th>3.49</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NT_{32}$</td>
<td>35</td>
<td>1.28</td>
<td>21300</td>
<td>3010</td>
<td>3000</td>
<td>18681.47</td>
<td>2947.63</td>
<td>2.07</td>
<td>2.03</td>
<td>1.75</td>
</tr>
<tr>
<td>$NT_{33}$</td>
<td>25</td>
<td>1.28</td>
<td>15630</td>
<td>2949</td>
<td>2880</td>
<td>19745.5</td>
<td>2687.89</td>
<td>8.85</td>
<td>4.68</td>
<td>6.67</td>
</tr>
<tr>
<td>$NT_{34}$</td>
<td>20</td>
<td>1.28</td>
<td>12590</td>
<td>2878</td>
<td>2730</td>
<td>16257.09</td>
<td>2696.49</td>
<td>6.31</td>
<td>4.72</td>
<td>1.23</td>
</tr>
</tbody>
</table>
these are compared with the results obtained by Nirjar [1]. The errors are listed to give a comparison between the values obtained from both methods.

The trained and tested network produced curvature formulae at ultimate stage in the form:

$$y_{78,u}^{out,p} = x_{78,u}^{out,p} \ast (10385.01) + 16567.54$$

$$x_{78,u}^{out,p} = \tanh \left( \sum x_{78,u}^{p} \right)$$

(6.76)

(6.77)

While the formulae for ultimate moment was in the form:

$$y_{79,u}^{out,p} = x_{79,u}^{out,p} \ast (3219.13) + 4261.97$$

$$x_{79,u}^{out,p} = \tanh \left( \sum x_{79,u}^{p} \right)$$

where $y_{78,u}^{out,p}$ and $y_{79,u}^{out,p}$ are the scaled curvature and moment at ultimate respectively. The full equations produced by the Flashcode and provided by the NeuralWorks® [17] is available on a disk at the end of the thesis.

6.3.6. Renew Stage for Yield Moment and Curvature

Previous work to determine the curvature and moments at both yield and ultimate was based on a procedure adopted by Nirjar [1]. This shows the capability of learning from a set of examples obtained from experimental work which can be used to extend the experimental study by implementing the predictive procedure provided by the neural network. Also with new experimental findings and Code of Practice revisions, the work can be enhanced and updated by reviewing specific experimental work and changes in the code of practice. Park and Ruitong [99] proposed new methods of defining the curvature and moments at yield and ultimate and their work is vital for this study. The first yield curvature was defined as:

$$\phi_y = \frac{e_y}{d - xd} = \frac{f_y}{E_s (1 - x) d}$$

(6.80)
Equation (6.80) is similar to equation (6.19) where the neutral axis depth factor, \( x \), is related to the concrete strain and is calculated by:

\[
x = \frac{\varepsilon_c}{f_y} \left( \frac{1}{E_s} + \varepsilon_c \right)
\]  

(6.81)

Alternatively, the neutral axis depth factor, \( x \), can be determined from the condition of the compression reinforcement, taking into consideration the fact that the sum of the horizontal forces equal zero. At yielding of the compression reinforcement, \( f_s = f_y \) and

since \( C_c + C_s = T \)

\[ \alpha f_c b x d + f_s A_s = f_y A_s \]  

(6.83)

Since \( \rho = \frac{A_s}{b d} \) and \( \rho' = \frac{A_c}{b d} \)

the neutral axis depth factor, \( x \), is determined by:

\[
x = \frac{f_y (\rho - \rho')}{\alpha f_c'}
\]  

(6.84)

If the compression reinforcement has not yielded, then the compression strain, \( \varepsilon_s \), is determined from strain compatibility as:

\[
\varepsilon_s = \left( \frac{xd - ad}{d - xd} \right) \frac{f_y}{E_s}
\]  

(6.85)

so that the sum of the horizontal forces for equation (6.83) is given as:

\[ \alpha f_c b x d + \left( \frac{xd - ad}{d - xd} \right) f_s A_s = f_y A_s \]  

(6.86)

Solving for \( x \) from the above quadratic equation gives:

\[
x = \left[ 0.5 + \frac{f_y (\rho' + \rho)}{2 \alpha f_c} \right] - \sqrt{\left[ 0.5 + \frac{f_y (\rho' + \rho)}{2 \alpha f_c} \right]^2 - \frac{f_y (\rho' a + \rho)}{\alpha f_c}}
\]  

(6.87)
The stress-strain relationship, as defined by Kent and Park [104] is shown in Figure 6.11. The parabolic region is defined for $\varepsilon_{c} < 0.002$:

$$f = f_{c} \left[ \frac{2\varepsilon_{c}}{0.002} - \left( \frac{\varepsilon_{c}}{0.002} \right)^2 \right]$$  \hspace{1cm} (6.88)

The linear region AB is defined in Figure 6.11 for $0.002 \leq \varepsilon_{c} \leq \varepsilon_{20c}$ as:

$$f = f_{c} \left[ 1.0 - z(\varepsilon_{c} - 0.002) \right]$$  \hspace{1cm} (6.89)

![Figure 6.11. Stress-Strain Relationship for Unconfined Concrete in Compression.](image)

The mean stress factor, $\alpha$, is determined from:

$$\alpha = \frac{\int_{0}^{\varepsilon_{cm}} f_{c} d\varepsilon_{c}}{f_{c} \varepsilon_{cm}}$$  \hspace{1cm} (6.90)

For the parabolic stress-strain relationship below a strain of $\varepsilon_{cm} = 0.002$, the mean stress factor, $\alpha$, is determined as:

$$\alpha = \frac{\varepsilon_{cm}}{0.002} \left( 1.0 - \frac{\varepsilon_{cm}}{0.006} \right)$$  \hspace{1cm} (6.91)
For the linear stress-strain relationship shown in Figure 6.11 when
$0.002 \leq \varepsilon_{cm} \leq \varepsilon_{20c}$, the mean stress factor, $\alpha$, is obtained from:

$$\alpha = \frac{0.004}{3\varepsilon_{cm}} + \left(1.0 - \frac{0.002}{\varepsilon_{cm}}\right)[1.0 - \frac{z}{2}(\varepsilon_{cm} - 0.002)]$$

(6.92)

The value of $z$, depends on the concrete cylinder strength, $f'_c$, for a range of
$3000 \leq f'_c \leq 5000$ (psi) and is determined from:

$$z = \frac{0.5}{\left(\frac{3 + 0.002f'_c}{f'_c - 1000}\right) - 0.002}$$

(6.93)

Expressing $f'_c$ in the SI system, the range is $20.70 \leq f'_c \leq 34.50$ $N/mm^2$ and:

$$z = \frac{0.5}{\left(\frac{0.0207 + 0.002f'_c}{f'_c - 6.90}\right) - 0.002}$$

(6.94)

where:

- $\varepsilon_c$ = compressive strain in the concrete.
- $\varepsilon_{20c}$ = concrete compressive strain when the concrete stress has reduced to $0.2 f'_c$.
- $\varepsilon_{cm}$ = concrete compressive strain in the extreme fibres.

The above computation was provided by Park and Ruitong [99] to determine the curvature and moment at the yield stage. These computations represent advanced research findings for the curvature and moment at the yield stage. Implementation of the renew stage benefits from this evaluation by integrating these equations into a form of numerical computation which were integrated with Nirjar's [1] procedure to enhance and update his results. The same network architecture, topology and neurodynamics were selected as in the preparatory stage in order to make comparisons and to obtain reliable results. However the training set pattern contains
732 patterns while the test pattern contains 90 patterns. The total number of possible combinations obtained was 816, generated numerically from 34 specimens. This covered three different concrete strengths 20, 30 and 45 $N/\text{mm}^2$, two different bar arrangements (2 and 4) and two different bar diameters, 6 and 12 $mm$, for both Nirjar [1] and Park and Ruitong [99]. The network ceased learning at the higher cycle of 77554 but with better percentages of error than the results obtained in the preparatory stage. This was due to a wider range of training and test data conforming to the same domain. Table 6.4 illustrates the renew stage results as compared to those of Nirjar [1].
Table 6.4. Comparison of Moments and Curvatures at Yield for the Renew Stage in Terms of Neural Networks.

<table>
<thead>
<tr>
<th>Specimens</th>
<th>$f'_c$ (N/mm$^2$)</th>
<th>$\rho$ (%)</th>
<th>Nirjar [1]</th>
<th>Neural Networks</th>
<th>%Error $= \frac{{\text{Abs}(1 - \frac{{\text{Network}}}{{\text{Desired}}})}}{100}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\phi_y \times 10^{-8}$</td>
<td>$M_y$ (N)</td>
<td>$M_y$ (N)</td>
</tr>
<tr>
<td>$NN_1$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120 (C)</td>
<td>2829 (C)</td>
<td>2880 (E)</td>
</tr>
<tr>
<td>$NN_2$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120 (C)</td>
<td>2829 (C)</td>
<td>2940 (E)</td>
</tr>
<tr>
<td>$NN_3$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120 (C)</td>
<td>2829 (C)</td>
<td>2850 (E)</td>
</tr>
<tr>
<td>$NN_4$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120 (C)</td>
<td>2829 (C)</td>
<td>2790 (E)</td>
</tr>
<tr>
<td>$NN_5$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120 (C)</td>
<td>2829 (C)</td>
<td>2760 (E)</td>
</tr>
<tr>
<td>$NN_6$</td>
<td>30.0</td>
<td>1.28</td>
<td>2120 (C)</td>
<td>2829 (C)</td>
<td>2760 (E)</td>
</tr>
</tbody>
</table>
Table 6.4. (continued)

<table>
<thead>
<tr>
<th>(NM_1)</th>
<th>30.0</th>
<th>1.28</th>
<th>2120</th>
<th>2829</th>
<th>2880</th>
<th>2096.45</th>
<th>2716.29</th>
<th>1.11</th>
<th>3.98</th>
<th>5.68</th>
</tr>
</thead>
<tbody>
<tr>
<td>(NM_8)</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2880</td>
<td>2096.45</td>
<td>2716.29</td>
<td>1.11</td>
<td>3.98</td>
<td>4.69</td>
</tr>
<tr>
<td>(NM_9)</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2820</td>
<td>2096.45</td>
<td>2716.29</td>
<td>1.11</td>
<td>3.98</td>
<td>3.68</td>
</tr>
<tr>
<td>(NM_{10})</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2820</td>
<td>2096.45</td>
<td>2716.29</td>
<td>1.11</td>
<td>3.98</td>
<td>3.68</td>
</tr>
<tr>
<td>(NO_{11})</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2760</td>
<td>2096.45</td>
<td>2716.29</td>
<td>1.11</td>
<td>3.98</td>
<td>1.58</td>
</tr>
<tr>
<td>(NO_{12})</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2760</td>
<td>2096.45</td>
<td>2716.29</td>
<td>1.11</td>
<td>3.98</td>
<td>1.58</td>
</tr>
<tr>
<td>(NO_{13})</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2760</td>
<td>2096.45</td>
<td>2716.29</td>
<td>1.11</td>
<td>3.98</td>
<td>1.58</td>
</tr>
<tr>
<td>(NO_{14})</td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2910</td>
<td>2096.45</td>
<td>2716.29</td>
<td>1.11</td>
<td>3.98</td>
<td>6.66</td>
</tr>
<tr>
<td>(NP_{15})</td>
<td>30.0</td>
<td>0.72</td>
<td>1900</td>
<td>1637</td>
<td>1650</td>
<td>1885.31</td>
<td>1674.73</td>
<td>0.77</td>
<td>2.30</td>
<td>1.50</td>
</tr>
<tr>
<td>(NP_{16})</td>
<td>30.0</td>
<td>2.00</td>
<td>2360</td>
<td>4310</td>
<td>4200</td>
<td>2340.98</td>
<td>4551.57</td>
<td>0.81</td>
<td>5.60</td>
<td>8.37</td>
</tr>
</tbody>
</table>

233
Table 6.4. (continued)

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NP_17</td>
<td>30.0</td>
<td>2.55</td>
<td>2540</td>
<td>4312</td>
<td>5260</td>
<td>2428.77</td>
<td>5645.86</td>
<td>4.38</td>
</tr>
<tr>
<td>NP_18</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5760</td>
<td>2463.29</td>
<td>6162.63</td>
<td>7.74</td>
</tr>
<tr>
<td>NQ_19</td>
<td>30.0</td>
<td>0.72</td>
<td>1900</td>
<td>1637</td>
<td>1680</td>
<td>1885.31</td>
<td>1674.73</td>
<td>0.77</td>
</tr>
<tr>
<td>NQ_20</td>
<td>30.0</td>
<td>2.00</td>
<td>2360</td>
<td>4310</td>
<td>4200</td>
<td>2340.98</td>
<td>4551.57</td>
<td>0.81</td>
</tr>
<tr>
<td>NQ_21</td>
<td>30.0</td>
<td>2.55</td>
<td>2540</td>
<td>5412</td>
<td>5160</td>
<td>2428.77</td>
<td>5645.86</td>
<td>4.48</td>
</tr>
<tr>
<td>NQ_22</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5690</td>
<td>2463.29</td>
<td>6162.63</td>
<td>7.74</td>
</tr>
<tr>
<td>NR_23</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5880</td>
<td>2455.83</td>
<td>6046.70</td>
<td>8.02</td>
</tr>
<tr>
<td>NR_24</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5820</td>
<td>2455.83</td>
<td>6046.70</td>
<td>8.02</td>
</tr>
<tr>
<td>NR_25</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5760</td>
<td>2455.83</td>
<td>6046.70</td>
<td>8.02</td>
</tr>
<tr>
<td>NR_26</td>
<td>30.0</td>
<td>2.99</td>
<td>2670</td>
<td>6281</td>
<td>5760</td>
<td>2455.83</td>
<td>6046.70</td>
<td>8.02</td>
</tr>
</tbody>
</table>
Table 6.4. (continued)

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NS&lt;sub&gt;27&lt;/sub&gt;</strong></td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2700</td>
<td>2096.45</td>
<td>2716.29</td>
</tr>
<tr>
<td><strong>NS&lt;sub&gt;28&lt;/sub&gt;</strong></td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2730</td>
<td>2096.45</td>
<td>2716.29</td>
</tr>
<tr>
<td><strong>NS&lt;sub&gt;29&lt;/sub&gt;</strong></td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2700</td>
<td>2096.45</td>
<td>2716.29</td>
</tr>
<tr>
<td><strong>NS&lt;sub&gt;30&lt;/sub&gt;</strong></td>
<td>30.0</td>
<td>1.28</td>
<td>2120</td>
<td>2829</td>
<td>2760</td>
<td>2096.45</td>
<td>2716.29</td>
</tr>
<tr>
<td><strong>NT&lt;sub&gt;31&lt;/sub&gt;</strong></td>
<td>40</td>
<td>1.28</td>
<td>2040</td>
<td>2857</td>
<td>2910</td>
<td>1993.90</td>
<td>2753.95</td>
</tr>
<tr>
<td><strong>NT&lt;sub&gt;32&lt;/sub&gt;</strong></td>
<td>35</td>
<td>1.28</td>
<td>2070</td>
<td>2846</td>
<td>2850</td>
<td>2064.33</td>
<td>2732.58</td>
</tr>
<tr>
<td><strong>NT&lt;sub&gt;33&lt;/sub&gt;</strong></td>
<td>25</td>
<td>1.28</td>
<td>2170</td>
<td>2812</td>
<td>2790</td>
<td>2125.76</td>
<td>2706.65</td>
</tr>
<tr>
<td><strong>NT&lt;sub&gt;34&lt;/sub&gt;</strong></td>
<td>20</td>
<td>1.28</td>
<td>2250</td>
<td>2790</td>
<td>2700</td>
<td>2154.44</td>
<td>2720.65</td>
</tr>
</tbody>
</table>
The specifications, trained and tested networks are shown in Figure 6.12 where the *confusion matrices* obtained for the curvature and moments are one with a *classification rate* of one. The test network also provided an excellent correlation of one with a *classification rate* of 1.0. It was observed that the time required to train the network is directly proportional to the number of training input data presented to the network as well as the size of the network. This was observed by comparing both implementations in the preparatory and renew stages at yield.

The formulae obtained for the yield curvature at the renew stage are:

\[ y^{\text{out},r}_{78,y} = x^{\text{out},r}_{78,y} \cdot (608.856) + 1950.375 \]  \hspace{1cm} (6.95)

\[ x^{\text{out},r}_{78,y} = \tanh(\sum x^r_{78,y}) \]  \hspace{1cm} (6.96)

while the yield moment is represented as:

\[ y^{\text{out},r}_{79,y} = x^{\text{out},r}_{79,y} \cdot (3033.731) + 4073.395 \]  \hspace{1cm} (6.97)

\[ x^{\text{out},r}_{79,y} = \tanh(\sum x^r_{79,y}) \]  \hspace{1cm} (6.98)

\[ y^{\text{out},r}_{78,y} \] and \[ y^{\text{out},r}_{79,y} \] represent the curvatures and moment presented by neural the networks in the final form. The remaining equations are included in a disk at the end of the thesis.
Figure 6.12. Curvature and Moment at Yield for the Renew Stage.
6.3.7. Renew Stage for Ultimate Moment and Curvature

Park and Ruitong [99] provided a simple expression to calculate the curvature and moments at ultimate stage, utilising the evaluation of the mean factor, $\alpha$, obtained at the yield stage. The proposed curvature at ultimate stage when the extreme compression fibre reaches the limiting value of $\varepsilon_u$ is written as:

$$\phi_u = \frac{\varepsilon_u}{x_ud}$$

(6.99)

The strain in the compression reinforcement is computed from:

$$\varepsilon_s = \left(\frac{x_ud - ad}{x_ud}\right)\varepsilon_u$$

(6.100)

By considering the internal equilibrium of the compressive forces in the steel and concrete and the tensile force in the steel and assuming that the compression reinforcement is in the elastic range:

$$\alpha f_c'bx_u + \left(\frac{x_ud - ad}{x_ud}\right)\varepsilon_u E_s A_s = f_y A_s$$

(6.101)

The above expression is then evaluated for $\varepsilon_s$ over the range $\left(-\frac{f_y}{E_s} < \varepsilon_s < \frac{f_y}{E_s}\right)$ and reduces to:

$$x_u = \frac{(f_y - \varepsilon_u E_s \rho') - \sqrt{(f_y - \varepsilon_u E_s \rho')^2 + 4\alpha f_c' \frac{ad}{d} \varepsilon_u E_s \rho'}}{2\alpha f_c'}$$

(6.102)

where yielding of the compression reinforcement occurs, the internal forces are related as:

$$\alpha f_c'bx_u + f_y A_s = f_y A_s$$

(6.103)

And the expression for $x_u$ is determined from:

$$x_u = \frac{f_y (\rho - \rho')}{\alpha f_c'}$$

(6.104)
where theoretically $\varepsilon_y$ is less than $\frac{-f_y}{E_t}$, i.e. the compression reinforcement yields in tension:

equation (6.103) becomes:

$$\alpha f_c' b x_d = f_y A_s + f_y A_f$$  \hspace{1cm} (6.105)

and the expression for $x_u$ is given by:

$$x_u = \frac{f_y (\rho + \rho')}{\alpha f_c'}$$  \hspace{1cm} (6.106)

The above expressions by Park and Ruitong [99] were programmed by FORTRAN to assess their effect on the values of moment and curvature. The work by Nirjar [1] was then merged with these evaluations in the form of training data containing 486 patterns and test data of 59. Similar network architecture, topology and neurodynamics were selected to be consistent with previous work as are shown in Figure 6.13. The network ceases learning at a cycle of 618861 with an approximation time of 6 hours largely due to the amount and type of training data. The trained and test data provide excellent correlations with the confusion matrices of 1.0 for both the curvature and the moment. The reliability of the network is then checked by passing the data obtained by Nirjar [1]. This provides a comparison of the curvatures and moments obtained by Nirjar [1] and these obtained using the new approach of Park and Ruitong [99]. Although higher error values were obtained in some cases, the results are satisfactory as shown in Table 6.5.
Figure 6.13. Curvature and Moment at Ultimate for the Renew Stage.
Table 6.5. Comparison of Moments and Curvatures at Ultimate by Nirjar [1] and by Neural Networks for the Renew Stage.

| Specimens | $f'_c$ | $N / mm^2$ | $\rho$ (%) | Nirjar [1] | Neural Networks | %Error = $Abs(1 - \frac{\text{Network}}{\text{Desired}}) * 100$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Computed (C)</td>
<td>Computed (E)</td>
<td>Experimental (E)</td>
</tr>
<tr>
<td>$NN_1$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3180</td>
<td>22094.20 2780.98</td>
</tr>
<tr>
<td>$NN_2$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3240</td>
<td>22094.20 2780.98</td>
</tr>
<tr>
<td>$NN_3$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3300</td>
<td>22094.20 2780.98</td>
</tr>
<tr>
<td>$NN_4$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3000</td>
<td>22094.20 2780.98</td>
</tr>
<tr>
<td>$NN_5$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2970</td>
<td>22094.20 2780.98</td>
</tr>
<tr>
<td>$NN_6$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2940</td>
<td>22094.20 2780.98</td>
</tr>
</tbody>
</table>
Table 6.5. (continued)

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_7$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3060</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NM_8$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3090</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NM_9$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3210</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NM_{10}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3270</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NO_{11}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2790</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NO_{12}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2820</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NO_{13}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>2910</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NO_{14}$</td>
<td>30.0</td>
<td>1.28</td>
<td>18470</td>
<td>3002</td>
<td>3060</td>
<td>22094.20</td>
<td>2780.98</td>
<td>19.62</td>
<td>7.36</td>
</tr>
<tr>
<td>$NP_{15}$</td>
<td>30.0</td>
<td>0.72</td>
<td>20100</td>
<td>1980</td>
<td>2040</td>
<td>20099.58</td>
<td>2780.98</td>
<td>0.0</td>
<td>5.00</td>
</tr>
<tr>
<td>$NP_{16}$</td>
<td>30.0</td>
<td>2.00</td>
<td>11860</td>
<td>4490</td>
<td>4500</td>
<td>10338.24</td>
<td>1881.03</td>
<td>12.83</td>
<td>16.29</td>
</tr>
</tbody>
</table>
Table 6.5. (continued)

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N_{P17}$</td>
<td>30.0</td>
<td>2.55</td>
<td>9290</td>
<td>5532</td>
<td>5275</td>
<td>8804.98</td>
<td>5221.43</td>
<td>5.22</td>
<td>10.08</td>
<td>15.44</td>
<td></td>
</tr>
<tr>
<td>$N_{P18}$</td>
<td>30.0</td>
<td>2.99</td>
<td>7940</td>
<td>6301</td>
<td>5775</td>
<td>8002.11</td>
<td>6089.44</td>
<td>0.78</td>
<td>0.50</td>
<td>9.65</td>
<td></td>
</tr>
<tr>
<td>$N_{Q19}$</td>
<td>30.0</td>
<td>0.72</td>
<td>20100</td>
<td>1980</td>
<td>2010</td>
<td>20099.58</td>
<td>6332.57</td>
<td>0.00</td>
<td>5.00</td>
<td>6.42</td>
<td></td>
</tr>
<tr>
<td>$N_{Q20}$</td>
<td>30.0</td>
<td>2.00</td>
<td>11860</td>
<td>4490</td>
<td>4400</td>
<td>10338.24</td>
<td>1881.03</td>
<td>12.83</td>
<td>16.29</td>
<td>18.67</td>
<td></td>
</tr>
<tr>
<td>$N_{Q21}$</td>
<td>30.0</td>
<td>2.55</td>
<td>929</td>
<td>5563</td>
<td>5170</td>
<td>8804.98</td>
<td>5221.43</td>
<td>5.22</td>
<td>9.46</td>
<td>17.67</td>
<td></td>
</tr>
<tr>
<td>$N_{Q22}$</td>
<td>30.0</td>
<td>2.99</td>
<td>7940</td>
<td>6301</td>
<td>5700</td>
<td>8002.11</td>
<td>6089.44</td>
<td>0.78</td>
<td>0.50</td>
<td>11.10</td>
<td></td>
</tr>
<tr>
<td>$N_{R23}$</td>
<td>30.0</td>
<td>2.99</td>
<td>7940</td>
<td>6301</td>
<td>6300</td>
<td>8209.21</td>
<td>6284.95</td>
<td>3.39</td>
<td>0.25</td>
<td>0.24</td>
<td></td>
</tr>
<tr>
<td>$N_{R24}$</td>
<td>30.0</td>
<td>2.99</td>
<td>7940</td>
<td>6301</td>
<td>6210</td>
<td>8209.21</td>
<td>6284.95</td>
<td>3.39</td>
<td>0.25</td>
<td>1.21</td>
<td></td>
</tr>
<tr>
<td>$N_{R25}$</td>
<td>30.0</td>
<td>2.99</td>
<td>7940</td>
<td>6301</td>
<td>5775</td>
<td>8209.21</td>
<td>6284.95</td>
<td>3.39</td>
<td>0.25</td>
<td>8.83</td>
<td></td>
</tr>
<tr>
<td>$N_{R26}$</td>
<td>30.0</td>
<td>2.99</td>
<td>7940</td>
<td>6301</td>
<td>5775</td>
<td>8209.21</td>
<td>6284.95</td>
<td>3.39</td>
<td>0.25</td>
<td>8.83</td>
<td></td>
</tr>
</tbody>
</table>

243
Table 6.5. (continued)

<table>
<thead>
<tr>
<th>$NT_{31}$</th>
<th>40</th>
<th>1.28</th>
<th>24000</th>
<th>3046</th>
<th>3090</th>
<th>27392.02</th>
<th>2859.76</th>
<th>14.13</th>
<th>6.11</th>
<th>7.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NT_{32}$</td>
<td>35</td>
<td>1.28</td>
<td>21300</td>
<td>3010</td>
<td>3000</td>
<td>25952.84</td>
<td>2867.57</td>
<td>21.84</td>
<td>4.73</td>
<td>4.41</td>
</tr>
<tr>
<td>$NT_{33}$</td>
<td>25</td>
<td>1.28</td>
<td>15630</td>
<td>2949</td>
<td>2880</td>
<td>16956.43</td>
<td>2338.64</td>
<td>8.49</td>
<td>20.70</td>
<td>18.80</td>
</tr>
<tr>
<td>$NT_{34}$</td>
<td>20</td>
<td>1.28</td>
<td>12590</td>
<td>2878</td>
<td>2730</td>
<td>15894.50</td>
<td>2614.12</td>
<td>26.25</td>
<td>9.17</td>
<td>4.24</td>
</tr>
</tbody>
</table>
The final formulae for curvature and moment at ultimate stage were presented by neural networks and results of the data provided by Park and Ruitong [99] and the evaluations by Nirjar [1] are in the form:

\[ y_{78,u}^{out,r} = x_{78,u}^{out,r} \times (20813.693) + 23738.135 \]  \hspace{1cm} (6.107)

\[ x_{78,u}^{out,r} = \tanh(\sum x_{78,u}^{r}) \]  \hspace{1cm} (6.108)

The above expression for the curvature and the moment is in the form:

\[ y_{79,u}^{out,r} = x_{79,u}^{out,r} \times (3219.13) + 4261.97 \]  \hspace{1cm} (6.109)

\[ x_{79,u}^{out,r} = \tanh(\sum x_{79,u}^{r}) \]  \hspace{1cm} (6.110)

where \( y_{78,u}^{out,r} \) and \( y_{79,u}^{out,r} \) are the curvature and moment respectively.

The analyses of curvature at the preparatory and renew stages indicate similar trends to those obtained from equations (6.78) and (6.109) respectively. This similarity is indicated in Table 6.5 which shows small error values when the trained and tested network is passed to the network for its reliability results.

The above computations for the yield and ultimate moment relationships at the preparatory and renew stages were carried out numerically on the 34 specimens investigated by Nirjar [1]. The results demonstrate that neural network methodology offers potential and valuable alternatives to obtaining curvatures and moments from experimental data. The objective demonstrated that is rather than carrying out intensive experimental tests, neural networks offer an alternative procedure by depending on fewer experimental specimen, fewer concrete strengths, specific bar arrangements and maximum and minimum bar sizes. By constructing a model that encounters in its functionality possible experimental testing combinations with numerical analysis potentiality, neural network paradigms provide an advance in the prediction of parameters that usually require repetition of the experimental work.
6.4. MOMENT-CURVATURE RELATIONSHIP FOR THE CONFINED CROSS-SECTION

The confinement of a section is effective at higher loads due to the contribution of the lateral steel stress. Confinement of the concrete can be achieved by reducing the spacing of the links and can also be improved to some extent by increasing the yield strength of the links. The ultimate flexural strength of a confined cross-section has been discussed in detail in chapter five and the neural network procedures adopted there for the unconfined state could be extended to carry out a similar procedure for this case. Several investigators have proposed an ultimate value for strain in a confined section. The ultimate strain adopted by Chan [11] is in the form of:

\[ \varepsilon_u^c = \varepsilon_u + \frac{(p^u)^{1/3}}{24.44} \]  \hspace{1cm} (6.111)

where:

\[ \varepsilon_u^c \] = ultimate concrete strain for the confined section.
\[ \varepsilon_u \] = ultimate concrete strain for the unconfined section.
\[ p^u \] = binding ratio parameter to allow for the effect of confinement as described in chapter 5.

Corely [105] proposed an expression to determine the ultimate strain for the confined concrete by defining the size-effect and the strength of the lateral reinforcement in the following equation:

\[ \varepsilon_u^c = 0.02 \frac{b}{a_z} + \left( \frac{p^u f_y}{140} \right)^2 + 0.003 \]  \hspace{1cm} (6.112)

where:

\[ a_z \] = distance between points of zero and maximum moments, \text{mm}. 
\[ f_{by} = \text{yield stress of the lateral reinforcement}, \text{ N/ m}^2. \]

Note that the ultimate concrete strain, \( \varepsilon_u^c \), depends on the volume of the links and the distance of the neutral axis from the compression face at ultimate, \( x_u d \).

Soliman and Yu [98] also reported on experimental results for the stress-strain relationship of a confined section. They represented a parameter, \( q^* \), which took into account the effectiveness of the confinement. They then adopted the following expression:

\[ \varepsilon_u^c = 0.003(1 + 0.8)q^* \]  \hspace{1cm} (6.113)

The above equations (6.112) and (6.113), are based on an ultimate strain of 0.003. The expression can however be modified to take into account variations in the ultimate strain as follows:

\[ \varepsilon_u^c = \varepsilon_u(1 + 0.8)q^* \]  \hspace{1cm} (6.115)

Utilising the knowledge of ultimate strain for the unconfined section and the information obtained in chapter 5 for the unbound concrete, provided by Nirjar's [1] analysis or the prediction of neural networks, the ultimate moment capacity of a doubly reinforced beam section can be modified and rewritten as:

\[ M_u = b d^2 [f_{\mu}(1 - g) + \rho f_y(1 - a)] \]  \hspace{1cm} (6.116)

For a singly reinforced beam the above expression reduces to:

\[ M_u = b d^2 f_{\mu}(1 - g) \]  \hspace{1cm} (6.117)

where:
\[ f'_{av} = \text{average confined compressive stress for bound concrete,} \]
\[ N/mm^2. \]

The other variables are as defined earlier. The ultimate curvature for the confined cross-section can be obtained as below:

\[ \phi_u^c = \frac{\varepsilon_u^c}{x_u d} \]  \hspace{1cm} (6.118)

The distance from the neutral axis to the compression face is then given as:

\[ x_u d = \frac{\rho f'_y}{f'_{av}} \]  \hspace{1cm} (6.119)
CHAPTER 7

SHEAR BEHAVIOUR AND STRENGTH CHARACTERISTICS

7.1. INTRODUCTION

Shear is an important phenomenon in concrete structures, mainly due to the weakness of concrete in tension. Shear is a significant parameter has been concentrated in the behaviour and design of deep beams, corbels and brackets. Extensive research during the past decades on understanding the fundamental behaviour of reinforced concrete members in shear. However, this high level of research has not led to the development of a rational theory applicable to flexural behaviour or axial load. The shear behaviour of a reinforced concrete beam is distinctly different from its behaviour in flexure. A shear failure is sudden without sufficient warning resulting from diagonal tensile cracks and occasionally compression cracks at inclined planes that are considerably wider than corresponding flexural cracks. A critical failure mechanism that must be considered is the shear-failure mechanism that develops diagonal tension failure due to combination of shear force and bending moments and occasionally the action of axial, torsion or a combination of these forces. These failures cause a reduction in the capacity of the flexural member which also results in a reduction in the ductility of the member.

There are three principal types of failure associated with shear namely: flexural, diagonal tension and shear compression. These types of failure produce cracks that are categorised as flexural, flexural-shear and web shear. The mode of failure of a beam depends on the shear span/depth ratio. The shear span for a uniformly
distributed load is the clear span. The shear span for a concentrated load is the
distance from the application of the load to the face of the support. The design
provisions for reinforced concrete members introduce an adequate margin of safety to
prevent sudden and brittle diagonal shear failure which might occur due to forces
acting on the structure during its lifetime. These provisions are based on the concept
that the shear failure of reinforced concrete occurs when the shear capacity of the
critical section is exceeded.

The numerous complex factors which influence the shear strength and
formulation of the inclined cracks hinder the establishment of an acceptable
mechanism for predicting the inclined cracking. The report by ACI-ASCE Task
Committee 326 [106] on shear and diagonal tension is devoted to structural members
with no web reinforcement. The revised report by ASCE-ACI Committee 426 [107]
extended the study and provided comprehensive knowledge on the shear mechanisms
in various concrete structural elements. These reports resulted from an intensive
knowledge of the principle of shear mechanism transfer, loading conditions and
information on shear strength quantitative evaluation.

This chapter reviews data on fundamental diagonal tension cracking and shear
failure phenomena provided by several investigators over the past few years and
evaluates the computation of diagonal tensile cracks which lead in most cases to
failure. The study is also extended to review the new code of practice for the
determination of shear strength of joints and provides evaluations and equations
which will encourage future work in this area. The beam-colum joint design Type I,
for gravity and normal wind loading, recommended by the ACI-ASCE 352 [2,3]
committees is explained in detail to provide an understanding of joint behaviour
under static and lateral loading. Type II covers joints in the inelastic range produced
by earthquakes or reversal of loading. High wind-loads represent another category
that is recommended by the committee but is beyond the scope of this study.
The study investigates the general feasibility of shear strength by using new published data. Having established an appropriate shear computational procedure, the work is extended to neural network implementation to predict several variables that are essential for this study. The manipulation of equations was carried out by FORTRAN programming and an individual equation was evaluated by Mathcad® 4 [16] to provide fast results. The backpropagation was implemented in its several forms to carry out the study and trained and tested networks were obtained for the predicting process. The results were compared with the practical work of Nirjar [1].

7.2. SHEAR STRESS CONCEPT

The adaptation and understanding of classical concepts of shear stress in a homogeneous, linearly elastic material and isotropic conditions are appropriate to gaining knowledge on the problem of shear in reinforced concrete members. The basic internal normal and shear stresses shown in Figure 7.1(a) are obtained respectively from [108]:

\[ f_T = \frac{M y}{I} \] 
\[ \nu = \frac{V A y_h}{I b} \] 

Considerations of equilibrium for infinitesimal elements within a beam provide an insight into the horizontal and vertical stress intensities which are eventually equal. By considering a beam with two infinitesimal elements of '1' and '2' as shown in Figure 7.1(b), and from the principles of classical mechanics applying Mohr's circle, the principal stresses for element '1' below the neutral axis for tension are found from:

\[ f_T^{\text{max}} = \frac{f_T}{2} + \sqrt{\left(\frac{f_T}{2}\right)^2 + \nu^2} \] 

For compression, the principal stresses are given by:
\[ f_{c}^{\text{max}} = \frac{f_T}{2} - \sqrt{\left(\frac{f_T}{2}\right)^2 + v^2} \quad (7.4) \]

And the maximum angle, \( \theta_{\text{max}} \), measured in radians for the occurrence of these stresses is obtained from:

\[ \tan 2\theta_{\text{max}} = \frac{v}{f_T/2} \quad (7.5) \]

where:

\[ f_T = \text{flexural stress due to bending, } N / mm^2. \]
\[ v = \text{shear stress, } N / mm^2. \]
\[ I = \text{moment of inertia of the gross cross-section, } mm^4. \]
\[ A_{cc} = \text{cross-area of the section at the plane that passes through the centroid of element '1', } mm^2. \]
\[ y = \text{distance from the element to the neutral axis, } mm. \]
\[ y_b = \text{distance from the centroid of } A_{cc} \text{ to the neutral axis, } mm. \]
\[ f_{l}^{\text{max}} = \text{maximum principal tensile stress, } N / mm^2. \]
\[ f_{c}^{\text{max}} = \text{maximum principal compressive stress, } N / mm^2. \]
\[ M = \text{applied moment at the cross-section, } N mm \]
\[ \theta_{\text{max}} = \text{the maximum angle for occurrence of maximum and minimum principal stresses measured in radians.} \]
Figure 7.1. Behaviour of Homogenous Rectangular Beam.
7.3. FUNDAMENTAL SHEAR TRANSFER FAILURE

Several investigators have attempted to refine the concept of concrete failure theories to understand the shear failure mechanism. The investigations either involve the description of mechanical truth behaviour or the development of dependable design recommendations for the joint. Reinforced concrete beams can be analysed by classical theories before the formulation of cracks to provide a guide in understanding the behaviour of reinforced concrete members.

7.3.1. Concrete Failure Criteria

The shear failure mechanism theories have been refined by several investigators to understand the complex state of stress including diagonal cracking, shear compressive failure, splitting and web crushing. The behaviour of concrete under three dimensional stress is fundamental to the development of a universal failure criterion. However, a simplified biaxial stress analysis which predicts the concrete failure criterion is adequate if the third stress direction is zero. The principal tensile stress or the principal tensile strain theories of failure for the un-cracked concrete cross-section were simplified as a cracking criterion to predicate the concrete failure. This approach was adopted by Cowan [109] and was useful in predicting tensile failure when applied to certain simple states of stress. Krahl, Khachaturian and Siess [110] presented an analysis method for a beam under combined shear and bending moment that provided reasonable results when applied to tensile cracking. Other investigators, including Hannant and Frederick [111], have emphasised the minor role of the intermediate principal stress in the contribution of strength and mechanism failures by indicating that only a limited number of experimental and analytical findings are related to compression.

The most acceptable concrete failure criteria were provided by Iyengar, Chandrashekasraka and Krishnaswamy [112] by testing concrete tubes until failure
under biaxial compression. Their test results provide a dimensionless form relationship that relates the octahedral shearing stress to the octahedral normal stress at failure by correlating the shear stress and octahedral normal stresses with the principal stresses. The behaviour of concrete under biaxial stresses in tension and compression has been reviewed by Kupfer Hilsdorf and Rüsch [113] to develop a universal failure criterion for concrete. They observed that the concrete strength in tension under biaxial compression was only 16.5 to 27 % larger than in uniaxial compression. Principal distortion energy was adopted by Ojha [114] in an attempt to provide criteria for concrete subjected to direct and shear stresses.

7.3.1.1. Mohr's Circle Failure Criteria

The behaviour of concrete at high stresses, the non-homogeneous nature of concrete and the fracture discontinuity influenced by micro-cracking have all contributed to the failure to develop an adequate general theory for the strength of concrete under combined stresses. However, modifications of several strength theories, such as Mohr, octahedral-stress theories and maximum-tension stress have been adopted with varying degrees of success. Mohr's rupture theory is based on the assumption that the materials act as a homogeneous body such that there is a relationship between the normal stress and shearing stress in any plane, and so that resistance to failure is controlled along that plane. The theory is widely used in many areas to predict the strength of a concrete section subjected to combined stresses. Mohr's theory assumes that the failure envelope occurred by splitting or sliding along a defined plane of rupture within the material so that the normal and shear stress can be related to each other and represented on a Mohr's stress circle representing a combination of the stresses that caused failure of the material. This relationship is shown in Figure 7.1(c) and Figure 7.2(a) and given by:

$$\tau = F(\sigma)$$

(7.6)
where:

\[ \tau = \text{general shear stress.} \]

\[ \sigma = \text{general normal stress.} \]

Intensive research was also carried out by several investigators [115,116] in an attempt to utilise Mohr's rupture theory to define a specific failure plane. The general findings recognised that the failure envelope was a combination of the normal and shearing stresses, as illustrated in Figure 7.2.
Shear stress

Normal stress (Tension)

Normal stress (Compression)

(a) Utilisation of Mohr's Envelope.

(b) Combinations of Shear and Normal Stress.

Figure 7.2. Failure Envelopes.

7.3.1.2. The ACI 318 Code Requirements between 1963 and 1989

The ACI building code requirements of 1963 [117] to predict shear cracking is specified in terms of stresses while the ACI codes of 1971 [73], 1977 [118], 1983 [119] and 1989 [81] specified the shear cracking in terms of forces.
The appearance of the first inclined cracks determined the shear strength of a beam with no web reinforcement. A knowledge of the principal stress as the cracks develop is essential to derive the basic equation. The controlling principal stresses in concrete are based on the shear stress, $v$, due to an external shear load, $V$, and the horizontal flexural stress, $f_T$, due to an external bending moment, $M$. The ACI Code of 1983 [119] provides an empirical relationship based on extensive experimental failures of a large number of beams with no web reinforcement. The derivation of the basic equation for two-dimensional principal stress at a point is given by:

$$f_{r}^\text{max} = f_t = \frac{f_T}{2} + \sqrt{\left(\frac{f_T}{2}\right)^2 + v^2} \quad \text{(7.7)}$$

It was assumed that ultimate strength is reached when the principle tensile stress, $f_{r}^\text{max}$, reaches the tensile strength, $f_t$, of concrete expressed as the modulus of rupture. Also it was shown in the previous chapters that the tensile strength of concrete, $f_t$, is a function of the compressive strength, $\sqrt{f_c}$. Rewriting the above equation (7.7) as a function of $\sqrt{f_c}$:

$$\sqrt{f_c} = K_1\left[\frac{f_T}{2} + \sqrt{\left(\frac{f_T}{2}\right)^2 + v^2}\right] \quad \text{(7.8)}$$

where $K_1$ is a constant.

Assuming that the flexural tensile stress, $f_T$, varies as the ratio of $E_c/E_s$ times the tensile reinforcement stress, $f_s$, or the moment of resistance of the cross-section $M_u$:

$$f_T \propto \frac{E_c}{E_s} f_s \propto \frac{E_c M_u}{E_s A_s d} \quad \text{(7.9)}$$

Therefore the expression for $f_T$ can be written in terms of the reinforcement ratio, $\rho$, in the tension side and since $E_c/E_s$ is a constant:
\[ f_T = K_2 \frac{M_u}{\rho bd^2} \]  

(7.10)

Also the expression for general shear stress at a specific section due to an external load is:

\[ v = K_3 \frac{V_u}{bd} \]  

(7.11)

Evaluation of the experimental model constants \( K_1, K_2 \) and \( K_3 \) are obtained by substituting \( f_T \) and \( v \) into equation (7.8). Rearranging terms yields the shear cracking stress at which diagonal flexure-shear cracking develops which can be represented in the form of a regression expression for (inch-pound units):

\[ v_c = \frac{V_c}{bd} = 1.9 \sqrt{f_c} + 2500 \rho \frac{V_u d}{M_u} \leq 3.5 \sqrt{f_c} \]  

(7.12)

Or for the SI system:

\[ v_c = \frac{V_c}{bd} = 0.16 \sqrt{f_c} + 17.2 \rho \frac{V_u d}{M_u} \leq 0.29 \sqrt{f_c} \]  

(7.13)

An empirical and conservative expression for equation (7.13) was proposed by ACI-ASCE Committee 326 [106] as (SI system):

\[ v_c = 0.16 \sqrt{f_c} \]  

(7.14)

MacGregor and Gergely [120] suggested a revision to ACI code 1971 [73] when dealing with shear in beams by replacing equation (7.13) by:

\[ v_c = \sqrt{f_c} \leq (0.067 + 10 \rho) \sqrt{f_c} \leq 0.19 \sqrt{f_c} \]  

(7.15)

It was observed that when the reinforcement ratio, \( \rho \), is less than 0.009, equation (7.15) gives smaller values than equation (7.14) and when \( \rho \) is larger than 0.012, equation (7.15) gives larger values than equation (7.14).
For a plain concrete section with no web reinforcement, the shear force, $V_u$, is equal to the shear strength, $V_c$, resisted by the concrete and by transforming equations (7.12) and (7.13) in terms of forces, these expressions are reduced respectively to:

\[
V_c = 1.9bd\sqrt{f_c} + 2500\rho \frac{V_s d}{M_u} bd \leq 3.5bd\sqrt{f_c}
\]  
(7.16)

\[
V_c = 0.16bd\sqrt{f_c} + 17.2\rho \frac{V_s d}{M_u} bd \leq 0.29bd\sqrt{f_c}
\]  
(7.17)

It should be noted that the ratio $V_s d / M_u$ must not exceed 1.0 in equations (7.16) and (7.17).

where:

$V_u$ = ultimate shear force at the cross-section, Newtons.

$M_u$ = ultimate moment at the cross-section, N-mm.

$V_c$ = shear force of concrete, Newtons.

$\rho$ = tensile reinforcement ratio of steel equal to ($A_s / bd$).

7.3.1.3. Shear Strength by Existing Data

Shear strength prediction by analysis of existing data was investigated by Zsutty [121] who combined the techniques of dimensional study and statistical regression analysis to provide an empirical method that is based on a separation of beam behaviour into arch action that has the characteristics of short beams and beam action that is characterised by slender beams. The study suggested that inclined cracking was as a result of significant variables caused by diagonal tension failures. The shear failure test data were obtained from several sources that investigated reinforced prismatic concrete beams with no web reinforcement subjected to one or two concentrated loads. The shear cracking stress, $V_{cr}$, which is equal to the concrete
shear stress, \( v_c \), for shear span/depth ratio greater than 2.5 was based on the following regression analysis:

\[
v_c = 59(f_c \rho \frac{d}{a_s})^{\frac{1}{2}} \quad (lb/in^2)
\]  

(7.18)

Or,

\[
v_c = 2.14(f_c \rho \frac{d}{a_s})^{\frac{1}{2}} \quad \text{(SI units)}
\]  

(7.19)

where:

- \( v_c \) = concrete shear stress, \( N/mm^2 \).
- \( a_s \) = shear span from the first concentrated load to the beam reaction, mm.

### 7.3.1.4. Shear Strength by the Regan Method

Intensive analytical and experimental investigations to determine the criteria of failure for a concrete section were carried out by Regan [122,123] at Imperial College. The study investigated in detail the failure criteria for a section subjected to biaxial loading of shear and relatively high compression. A relationship was established that relates the normal stress to the shear strength as:

\[
\tau = C_r + \mu_f \sigma
\]  

(7.20)

where:

- \( \tau \) = general shear stress.
- \( C_r \) = cohesive constant \( \approx 0.44(f_{cu})^{0.67} \).
- \( \mu_f \) = coefficient of friction, between 0.8 and 1.0.
- \( \sigma \) = general normal stress.
Assuming the cylinder compressive of concrete strength is given by:

\[ f'_c = 0.8 f_{cu} \]  \hspace{1cm} (7.21)

Equation (7.20) can be approximated to:

\[ \tau = 0.465 \sqrt{(f'_c)^2} \]  \hspace{1cm} (7.22)

To obtain an expression for the shear strength carried by the concrete, several analyses for shear resistance, including aggregate interlock action were carried out by Regan [122,123] that involved the resistance of a section versus a shear failure and a shear compression failure. The expression derived by Regan [122,123] for the shear strength based on a specific partial factor of safety is:

\[ V_c = k_r (f'_{cu} \frac{100 A_x}{bd})^{1/3} \]  \hspace{1cm} (7.23)

where:

\[ f_{cu} \] = characteristic concrete strength.

\[ k_r \] = factor of safety, between 0.20 to 0.25.

The shear resistance derived for a short structural member is expressed as:

\[ \frac{V_u}{V_c} = \frac{2d}{a_z} \]  \hspace{1cm} (7.24)

A conservative expression was adopted by CP110 [72] for the beam-column joint as found by Taylor [124] which is used in design:

\[ \frac{V_u}{V_c} = 3 + \frac{2d_{ce}}{a_b} \]  \hspace{1cm} (7.25)

where:

\[ a_b \] = lever arm of the beam, mm.
\( a_z \) = distance between points of zero and maximum moments, mm.

\( d_{ce} \) = column effective depth, mm.

\( v_u \) = ultimate shear stress, N/mm\(^2\).

7.3.1.5. **Batchelor and Kwun Approach**

Batchelor and Kwun [125] investigated shear in reinforced concrete beams with no web reinforcement to evaluate the shear stress. The test specimens consist of 276 beams where the shear span/depth ratio was greater than 2.0. A statistical analysis was adopted by combining their test results with 266 beams investigated by other researchers. A conservative design formula was proposed in the form of:

\[
v_c = (0.05 + 9.13p)\lambda \sqrt{f_c}
\]

(7.26)

Where the range of \( v_c \) is between 0.083\( \lambda \sqrt{f_c} \leq v_c \leq 0.187\lambda \sqrt{f_c} \) and for normal weight concrete the value of \( \lambda = 1 \). Equation (7.26) was derived for reinforced concrete with no web reinforcements, where \( a/d > 2.0 \) and is expressed \( f_c \) in N/mm\(^2\).

7.3.1.6. **Approximation Analytical Approach**

Kim and White [126] proposed a hypothesis for the computation of the shear cracking mechanism in reinforced beams subjected to concentrated loads with no web reinforcement. The study is based on observations of local stress concentration associated with the type of bond between the concrete and the flexural reinforcement as well as the development of arch action at the shear span close to the end span of the beam. The following analytical formula was obtained for the inclined stress cracking position, \( a_c \):

\[
a_c = k_3 \left[ \frac{\rho d^2}{(1 - \sqrt{\rho})^2} \right]^\frac{1}{3} a
\]

(7.27)
the inclined shear cracking load, \( V_c \), was given as:

\[
V_c = k_4 \left[ \sqrt{\rho (1-\sqrt{\rho})^2 (d/a)} \right]^{\frac{1}{3}} \sqrt{f_c} bd
\]  

(7.28)

where \( k_3 \) and \( k_4 \) are constants evaluated for test data, and expressed as:

\[
k_3 = (k_1/k_2)^{2/3} \quad \text{and} \quad k_4 = (k_1 k_2^2)^{1/3}
\]  

(7.29)

According to Kim and White [126], the constant \( k_3 \) was difficult to determine due to the fact that a limited number of tests recording the value \( a_c \) were available, however data obtained from a statistical analysis confirmed that a value of 3.3 is reliable, and the final inclined shear cracking position of equation (7.27) reduced to:

\[
a_c = 3.3 \left[ \frac{\rho (d/a)^2}{(1-\sqrt{\rho})^2} \right]^{1/3} a
\]  

(7.30)

The constant \( k_4 \) was based on voluminous data available from experimental testing of a wide range of beam properties which reduced equation (7.28) to:

\[
V_c = 0.752 \left[ \sqrt{\rho (1-\sqrt{\rho})^2 (d/a)} \right]^{\frac{1}{3}} \sqrt{f_c} bd
\]  

(7.31)

It was further observed by Kim and White [126] that the inclined shear cracking load, \( V_c \), obtained from equation (7.31) was similar to that given by the Zsutty equation [121] obtained earlier, which was based on statistical analysis of a test data.

7.3.1.7. Fracture Mechanics Approach

A more recent approach proposed by So and Karihaloo [127] estimated the maximum force in the steel by studying the ultimate bond stress of the reinforcing bar and the longitudinal splitting strength failure of the concrete. The method was based on an analysis of the splitting failure of concrete which was insensitive to the shape of the tension softening diagram of concrete, and was caused by bond stress. This approach gave consistent results. According to So and Karihaloo [29] the shear
capacity of the beams was improved due to the contributions of dowel action and aggregate interlock.

7.3.2. Reinforced Concrete Shear Transfer Mechanism Failure

The interdependence of many factors affects the shear transfer mechanism of concrete and makes the isolation of one single effect difficult to achieve. Shear transfer mechanism failures in reinforced concrete members are caused by a combination of stresses acting on an inclined crack plane due to the mechanism illustrated in Figure 7.3(a):

7.3.2.1. Shear Resistance of Un-cracked Concrete Section

The shear resistance of an un-cracked concrete section is the simplest form of shear transfer. The combination of compressive and tensile stresses produces a principal stress that causes cracking or crushing of the concrete as explained previously.

7.3.2.2. Interface Shear Stress Transfer

This phenomenon is known as an aggregate interlock shear stress, shear friction, tangential shear transfer or surface roughness. The interface shear stress transfer across a plane occurred due to slippage of a plane. Several types of interface shear transfer are available depending on the condition of the existing cracks or the interface. In monolithic concrete, diagonal cracks occur across the interface which causes a truss action along the plane resulting in failure. This type of phenomena has been studied by several investigators who have attempted to prevent this type of failure by introducing lateral reinforcement. The failure usually occurs along a rough and irregular crack plane through which shear transfer is transmitted by the existing lateral reinforcement such that slip occurs along the cracked areas of the concrete causing separation on either side of the crack. The separation imposes stresses on the steel in tension that create compressive stresses in the concrete across the crack. It
was observed that the shear capacity in this case was proportional to the average restraining stress, $\rho_{v} f_{y}$, which is the lateral steel ratio, $\rho_{v}$, multiplied by the yield stress of the lateral reinforcement, $f_{y}$, as shown in Figure 7.3(b). This concept is analogous to the simple friction case which is known as the "shear-friction hypothesis" where the shear stress and force at failure can be determined respectively from:

$$v_{f} = \rho_{v} f_{y} \tan \phi_{f} \quad (7.32)$$

and,

$$V_{f} = A_{v} f_{y} \tan \phi_{f} \quad (7.33)$$

where:

- $v_{f}$ = average shear stress at the shear interface of the cracked area, $N/mm^{2}$.
- $V_{f}$ = shear force along the shear interface, Newtons.
- $f_{y}$ = yield stress of the lateral reinforcement, $N/mm^{2}$.
- $\phi_{f}$ = the equivalent coefficient of friction determined by tests.(analogous to the cohesionless of soils).
- $A_{v}$ = total area of lateral reinforcement, $mm^{2}$.
- $\rho_{v}$ = lateral reinforcement ratio equals ($A_{v}/bd$).

At higher loads, the lateral reinforcement prevents failure due to short diagonal cracks across the shear plane, as shown Figure 7.3(c). Failure only occurs if the steel reinforcement and the compression between the diagonal cracks form a truss action to resist further loading. This type of failure mechanism is initiated by the action of combined normal and shear forces that crush the compression diagonals.
7.3.2.3. **Shear Transfer in Reinforced Concrete by Lateral Reinforcement**

The function of shear lateral reinforcement is to carry a portion of the externally applied shear force, restrict the progress of inclined tension diagonal cracks and provide confinement of the concrete in the compression zone. Shear lateral reinforcement also provides dowel capacity to assist sustaining the flexural load by holding the main reinforcement in place, increases the effective capacity of the structural member and slows the progress of interface shear transfer.

Although the truss analogy has provided a simple explanation for the behaviour of beams in shear action for some years, it did not include significant components of the shear force transmission. A concept is now required to design shear reinforcement to prevent shear failure once the shear capacity is exceeded in order to attain the beam's flexural capacity. The shear force strength provided by stirrups can be derived from:

\[ V_s = \frac{A_s f_y d}{s_v} \]  

(7.34)

where:

- \( V_s \) = shear force provided by stirrups, *Newton*.
- \( s_v \) = spacing of the lateral reinforcement, *mm*.

7.3.2.4. **Arch Action Stress**

This phenomenon is observed in deep beams and slabs where part of the load is carried by the arch action provided by a substantial horizontal reaction at the support. This imposes tremendous demands on the anchorages and as result accounts for most types of the arch failure.

7.3.2.5. **Dowel Action Stress**
The dowelling force in the bar resists the crack that crosses the reinforcing bars. Generally, the dowel shear force is not important in a beam relative to other shear mechanisms and the dowel test is characterised by several variables which make the testing process tedious.
Principal tensile stresses

(a) shear resistance forces after formation of inclined crack

(b) Interface Shear Transfer Stress at Existing Crack Surface

(c) Interface Shear Transfer form Across a Plane in Un-cracked Zone

Figure 7.3. Reinforced Concrete Shear Transfer Mechanism Failure.
7.4. SHEAR CRACKING STRESS AT THE PREPARATORY STAGE

The analysis at this stage involves the evaluation of shear cracking stress by either the ACI code of 1963 [117], Zsutty [121], or Regan [122] to observe the contribution of each equation. The study was carried out by generating training and test files for the above three methods. Having generated their results, a network was set up contains 7 PE's in the input layer (b, d, f', ρ, r_b, d_b, A_b). The output layer contains the shear cracking stress, v_c. Two hidden layers were selected with 20 and 10 PE's in the first and second layers respectively. The learning rate for the first hidden layer was 0.4 and the second was 0.35 while the output layer was 0.25. A high momentum value of 0.6 was selected to avoid any local minimum entrapment and the epoch size was set to 36. The selection of this network was based on trial and error due to the difficulty in reaching convergence as a result of the diverse nature of the equations used. By tuning the network and trying many learning algorithms, the final learning algorithm selected was EDBD with the option of fast learning. This option adds a portion of the estimated error to the activation value of the connection source PE prior to the connection learning to enhance conversion. A value of 5% was selected in the RMS instrument and the network ceased learning at 40723 cycle. The full network parameter results with the confusion matrices and classifications rates are shown in Table 7.1 which was obtained by passing each investigator's result to the trained and tested network to provide comparisons between the investigator's networks. The network architectures, topologies and neurodynamics obtained by neural networks for the trained and tested network are shown in Figure 7.4 with the prediction networks by the ACI code [117], Zsutty [121] and Regan [122]. The numerical values obtained by each investigator and the neural networks are shown in Table 7.2 with the percentage errors. The networks provide excellent correlations compared to the ACI code [117] while a higher percentage error was obtained by Zsutty [121] and Regan [122], although these results are satisfactory.
Table 7.1. Network Parameters for Shear Cracking Stress at the Preparatory Stage.

\[
\% Error = Abs(1.0 - \frac{\text{Network}}{\text{Desired}}) \times 100
\]

Network ceased *learning* and converge at 40723 learn

Convergence criteria \( RMS = 5 \% \)

\[\text{Epoch Size} = 36\]

<table>
<thead>
<tr>
<th>Network Parameters at Preparatory Stage</th>
<th>Confusion Matrices</th>
<th>Classification Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trained and Tested Network</td>
<td>0.8440</td>
<td>0.8455</td>
</tr>
<tr>
<td>ACI [117]</td>
<td>0.9230</td>
<td>0.50</td>
</tr>
<tr>
<td>Zsutty [121]</td>
<td>0.9388</td>
<td>0.50</td>
</tr>
<tr>
<td>Regan [122]</td>
<td>0.9354</td>
<td>0.50</td>
</tr>
</tbody>
</table>
Figure 7.4. Shear Cracking Stress by Neural Networks at the Preparatory Stage.
Table 7.2. Comparisons of Concrete Shear Cracking by Several Investigators and Neural Networks at Preparatory Stage.

<table>
<thead>
<tr>
<th>$f'_c$</th>
<th>$\rho$</th>
<th>ACI (Equation (7.13))</th>
<th>Zsutty (Equation (7.19))</th>
<th>Regan (Equation (7.23))</th>
<th>Neural Networks</th>
<th>% Error = $\text{Abs}(1.0 - \frac{\underline{\underline{\text{Neural}}} \text{ Networks}}{\underline{\underline{\text{Desired}}}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.0128</td>
<td>0.79</td>
<td>0.96</td>
<td>0.74</td>
<td>0.84</td>
<td>6.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.50</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13.51</td>
</tr>
<tr>
<td>25</td>
<td>0.0128</td>
<td>0.88</td>
<td>1.03</td>
<td>0.79</td>
<td>0.87</td>
<td>1.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10.13</td>
</tr>
<tr>
<td>30</td>
<td>0.0072</td>
<td>0.92</td>
<td>0.91</td>
<td>0.70</td>
<td>0.86</td>
<td>6.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22.86</td>
</tr>
<tr>
<td>30</td>
<td>0.0128</td>
<td>0.95</td>
<td>1.10</td>
<td>0.84</td>
<td>0.91</td>
<td>4.21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>17.27</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8.33</td>
</tr>
<tr>
<td>30</td>
<td>0.020</td>
<td>1.00</td>
<td>1.28</td>
<td>0.98</td>
<td>1.01</td>
<td>1.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>21.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.06</td>
</tr>
<tr>
<td>30</td>
<td>0.0256</td>
<td>1.03</td>
<td>1.38</td>
<td>1.06</td>
<td>1.02</td>
<td>0.97</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>26.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.77</td>
</tr>
<tr>
<td>30</td>
<td>0.0288</td>
<td>1.05</td>
<td>1.05</td>
<td>1.11</td>
<td>1.16</td>
<td>10.48</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>19.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.50</td>
</tr>
</tbody>
</table>
Table 7.2. (continued)

<table>
<thead>
<tr>
<th></th>
<th>30b</th>
<th>0.0300</th>
<th>1.06</th>
<th>1.46</th>
<th>1.12</th>
<th>1.12</th>
<th>1.12</th>
<th>5.66</th>
<th>23.29</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>35a</td>
<td>0.0128</td>
<td>1.02</td>
<td>1.16</td>
<td>0.89</td>
<td>0.98</td>
<td>3.92</td>
<td>15.52</td>
<td>10.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>0.0128</td>
<td>1.15</td>
<td>1.26</td>
<td>0.97</td>
<td>1.17</td>
<td>1.74</td>
<td>7.14</td>
<td>20.62</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b Partial data seen by neural networks.

a Data not seen by neural networks.
7.5. SHEAR CRACKING STRESS AT THE RENEW STAGE

The ACI code of 1989 [81] provided substantial modifications for the design and analysis of flexural shear strength. The work by Batchelor and Kwun [125] also provided an understanding of the shear phenomena and adequate shear cracking strength computations. Kim and White [126] laid down reliable findings for shear cracking and its location that have been adopted in modifications to previous work to understand and enhance the data that relies on new experimental findings, both theoretical and analytical. Similar architectural, topology and neurodynamics were selected for the renew stage to observe the differences between the preparatory and the renew stage, as shown in Figure 7.5. The network ceased learning at 297352, approximately three hours longer than for the preparatory stage. Table 7.3 provides network parameter predictions by the three investigators with the trained and tested network. A comparison at the renew stage for certain specimens is shown in Table 7.4 with the percentage error for all investigators. The percentage error obtained by Kim and White [126] is high compared to the other results. Generally the results obtained by ACI code [117] and Zsutty [121] are better than those obtained by Regan [122] and Kim and White [126].
Table 7.3. Network Parameters for Shear Cracking Stress at Renew Stage.

\[ \% Error = \text{Abs}(1.0 - \frac{\text{Network}}{\text{Desired}}) \times 100 \]

Network ceased learning and converge at 297352 learn

Conference criteria \( RMS = 5 \% \)

\[ \text{Epoch Size} = 36 \]

<table>
<thead>
<tr>
<th>Network Parameters at Renew Stage</th>
<th>Confusion Matrices</th>
<th>Classification Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trained Network</td>
<td>0.8812</td>
<td>0.7982</td>
</tr>
<tr>
<td>Trained and Tested Network</td>
<td>0.7975</td>
<td>0.7793</td>
</tr>
<tr>
<td>ACI [117]</td>
<td>0.9707</td>
<td>0.5000</td>
</tr>
<tr>
<td>Zsutty [121]</td>
<td>0.9303</td>
<td>0.5000</td>
</tr>
<tr>
<td>Regan [122]</td>
<td>0.9317</td>
<td>0.5000</td>
</tr>
<tr>
<td>Kim and White [126]</td>
<td>0.9469</td>
<td>0.5000</td>
</tr>
</tbody>
</table>
Figure 7.5. Shear Cracking Stress by Neural Networks at Renew Stage.
Table 7.4. Comparisons of Concrete Shear Cracking by Several Investigators and Neural Networks at Renew Stage.

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$\rho$</th>
<th>$\sigma_{cr}$ $N/mm^2$</th>
<th>%Error $= \text{Abs}(1.0 - \frac{\text{Neural}}{\text{Desired}})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0.0128</td>
<td>0.79</td>
<td>0.96</td>
</tr>
<tr>
<td>25</td>
<td>0.0128</td>
<td>0.88</td>
<td>1.03</td>
</tr>
<tr>
<td>30</td>
<td>0.0072</td>
<td>0.92</td>
<td>0.91</td>
</tr>
<tr>
<td>30</td>
<td>0.0128</td>
<td>0.95</td>
<td>1.10</td>
</tr>
<tr>
<td>30</td>
<td>0.020</td>
<td>1.00</td>
<td>1.28</td>
</tr>
</tbody>
</table>
Table 7.4. (continued)

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>0.0256</td>
<td>1.03</td>
<td>1.38</td>
<td>1.06</td>
<td>1.40</td>
<td>1.16</td>
<td>12.62</td>
<td>15.94</td>
<td>9.43</td>
<td>17.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.0288</td>
<td>1.05</td>
<td>1.05</td>
<td>1.11</td>
<td>1.41</td>
<td>1.19</td>
<td>13.33</td>
<td>17.36</td>
<td>7.21</td>
<td>15.60</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.0300</td>
<td>1.06</td>
<td>1.46</td>
<td>1.12</td>
<td>1.42</td>
<td>1.21</td>
<td>14.15</td>
<td>17.12</td>
<td>8.04</td>
<td>14.79</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>0.0128</td>
<td>1.02</td>
<td>1.16</td>
<td>0.89</td>
<td>1.39</td>
<td>1.10</td>
<td>7.84</td>
<td>5.17</td>
<td>23.60</td>
<td>20.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>0.0128</td>
<td>1.15</td>
<td>1.26</td>
<td>0.97</td>
<td>1.58</td>
<td>1.26</td>
<td>9.57</td>
<td>0.00</td>
<td>29.90</td>
<td>20.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b Partial data seen by neural networks.

aData not seen by neural networks.
7.6. THE STRENGTH AND BEHAVIOUR OF BEAM-COLUMN JOINTS

The use of high strength materials, larger reinforcing bars and smaller member cross-sections has changed the approach of joint design which is no longer limited to satisfying the anchorage requirements. The fundamental requirements at the joint are to carry all the existing forces at the end of members and transmit them through the joint to support the members.

7.6.1. Axial Force and Shear Strength of a Joint

The failure criteria of reinforced concrete members subjected to combinations of axial force and shear for an isotropic material as suggested by Seth [128] is in the form given below:

\[(\sigma_1 - \sigma_3) - C_1(\sigma_1 + \sigma_2 + \sigma_3) = C_2\]  \hspace{1cm} (7.35)

The following elliptical relationship was obtained for the formulation of the biaxial stress condition in which \(\nu\), \(f_c\), and \(f\) are expressed in kg/cm²:

\[
\frac{\nu}{f_c} = 0.5 \sqrt{[0.0484 + 0.342(\frac{f^2}{f_c^2}) - 0.3916(\frac{f}{f_c})^2]}
\]  \hspace{1cm} (7.36)

The ACI-ASCE Committee 326 [106] adopted an approximation for equation (7.36) by considering the principal tensile stresses at the tip of the flexural cracks in the presence of an axial compression force, \(P_a\), in the form of:

\[
\nu_e = 0.16\sqrt{f_c^2 + 17.24\rho\left[\frac{V_c d}{M - P_a(0.5H - 0.125d)}\right]} \leq 0.3 \sqrt{f_c^2(1 + 0.3\frac{P_a}{A_g})}
\]  \hspace{1cm} (7.37)

where:

\(A_g\) = gross area of cross-section, \(mm^2\).

\(C_1, C_2\) = material constants for the biaxial stress state.
Regan [122,123] proposed an expression for evaluating the shear force in an axially loaded member that resulted in a lower bound value in which \( v_c \) is expressed in \( N/\text{mm}^2 \):

\[
V_c = v_c bd (1 + 0.17 \frac{P_a d}{M}) \tag{7.38}
\]

The expression for evaluating the shear carried by the concrete was obtained by equating the maximum principal tensile stress, \( \sigma_1 \), to the concrete tensile stress, \( f_t \):

\[
\sigma_1 = \sqrt{\tau^2 + \frac{\sigma^2}{2} - \frac{\sigma}{2}} = f_t \tag{7.39}
\]

Hence the shear stress carried by the concrete is:

\[
\tau = f_t \sqrt{1 + \frac{\sigma}{f_t}} \tag{7.40}
\]

For the beam-column joint section, taking

\[
\tau = \frac{1.5V}{b_c d_c} \tag{7.41}
\]

the shear stress carried by the concrete, as adopted by Taylor [124], in defining the diagonal cracking shear stress for the beam-column joint reduces to:

\[
v_c = 0.67 f_t \sqrt{1 + \frac{P_a}{A_g f_t}} \tag{7.42}
\]

The experimental evaluation for the shear stress, \( v_c \), carried by the joint according to Nirjar [1] is:
\[ v_c = \sqrt{f_t^2 + f_t \frac{p_a}{A_g}} \]  

(7.43)

Based on two beams framing into the column.

By substituting the expression for \( f_t = 0.65 \sqrt{f_c} \) as proposed by Nirjar [1] in equation (7.43) the above expression becomes:

\[ v_c = 0.65 \sqrt{f_c (1 + 0.24 \frac{p_a}{A_g})} \text{ for } f_c = 40 \text{ N/mm}^2. \]  

(7.44)

\[ v_c = 0.65 \sqrt{f_c (1 + 0.34 \frac{p_a}{A_g})} \text{ for } f_c = 20 \text{ N/mm}^2. \]  

(7.45)

The average for the above expression, as proposed by Nirjar [1], is:

\[ v_c = 0.65 \sqrt{f_c (1 + 0.30 \frac{p_a}{A_g})} \]  

(7.46)

Rewriting the above expression, and allowing for two beams framing into a joint, so that \( \gamma = 1.5 \):

\[ v_c = 1.5 \times 0.67 f_t \sqrt{1 + \frac{p_a}{A_g f_t}} \]  

(7.47)

7.6.2. ACI-ASCE Committee 352 (1976)

ACI-ASCE Committee 352-1976 [2] suggested substantial recommendations for the design of the joints in a cast-in place reinforced concrete frame structure. These recommendations were based on the strength and ductility requirements that are related to the joint function. The recommendations apply only to normal weight concrete for two classes of joints and are based on specific loading conditions and anticipated deformations of the joint when subjected to lateral loading. Type I is restricted to static loading with no large strains in the elastic region so that significant ductility is not acceptable while Type II for earthquakes or equivalent blasts requires
substantial strength through large deformations or reversal of load and implies significant ductility.

*Type I* joints consider two types of behaviour which depend on the loading conditions which are characterised as gravity and lateral loading. Figure 7.6 shows a *Type I* joint which falls into these categories.
Joint in a Frame Structure.

Net Moment at Joint = $M_1 + M_2$
Joint Shear is Maximised

(a) Behaviour of Joint Under Vertical Loads

Net Moment at Joint = $M_1 + M_2$
Joint Shear is Maximised

(b) Behaviour of Joint Under Lateral Loads

Figure 7.6. Beam-Column Joints Under Loading For Type I.
The design of beam-column joints must consider factors such as axial load, shear, bending moment, reinforcement developments, torsion and secondary effects (shrinkage, creep, settlement and temperature). The axial forces transmit loads through the joint which require the provision of adequate lateral reinforcements around the column main reinforcement or additional lateral reinforcement for the joint itself. The performance of a joint depends strongly on the amount of confinement. The confinement of a joint strengthens the core concrete, improves the strain capacity and prevents the column reinforcement from buckling. The confinement can be accommodated either by lateral reinforcement within the joint or by beams framing into the joint which are perpendicular to the plane of the forces being considered. If forces are transmitted along the 'main beam', as shown in Figure 7.7(a), then it is important to check the confinement for the spandrel beam, i.e:

\[ b_2 \geq \frac{3}{4} b_1 \; ; \; h_2 \geq \frac{3}{4} h_i \]  

(7.48)

But if:

\[ b_2 < \frac{3}{4} b_1 \; ; \; h_2 < \frac{3}{4} h_i \]  

(7.49)

Then lateral reinforcements must be provided.

7.6.2.1. Axial or Compression Load Design Requirements

Lateral reinforcement is needed around the column bars where:

\[ P_u > 0.4 P_b \]  

(7.50)

The reinforcement ratio, \( \rho_s \), assuming rectangular hoops is calculated from:

\[ \rho_s = \frac{A_{ch}}{L_h s_h} = 0.3\left(\frac{A_{ch}}{A_{ch}} - 1\right) \frac{f_c'}{f_y} \]  

(7.51)

For spiral hoops, the volumetric ratio, \( \rho_s \), is computed from:
\[ \rho_s = 0.45 \left( \frac{A_g}{A_{ch}} - 1 \right) \frac{f'_c}{f_y} \]  

(7.52)

Lateral reinforcement is not needed around the column bars if:

\[ P_u \leq 0.4 P_b \]  

(7.53)

where:

\[ P_u \] = column ultimate strength, *Newtons*.

\[ P_b \] = axial load capacity at the simultaneously assumed ultimate strain of concrete and yielding of tension steel, *Newtons*.

\[ A_{sh} \] = total area of all hoops and extra cross-tie legs crossing the mid-depth of the section, within a spacing, \( s_h \), in the direction considered, *\( mm^2 \).*

\[ A_{ch} \] = area of rectangular core measured to the outside of the hoop or tie, *\( mm^2 \).*

\[ A_g \] = gross area of cross-section, *\( mm^2 \).*

\[ s_h \] = spacing of the tie reinforcement measured along the column bars, *\( mm \).*

\[ L_h \] = width, measured to outside of the tie reinforcement in the direction of the perpendicular to that of the shear force being considered, *\( mm \).*

### 7.6.2.2. Shear Design Requirements

The shear stress, \( \nu_c \), carried by the concrete cross-section as recommended by the committee shall not exceed the following value for \( P_u / A_g \) in *\( N / mm^2 \)*:
\[ v_c \leq 0.3\beta y \sqrt{f'_c(1 + 0.3\frac{P_a}{A_g})} \quad (7.54) \]

where:

\[ \beta = 1.4 \text{ for joint Type I.} \]

\[ \gamma = 1.4 \text{ for a confined joint and } 1.0 \text{ for an unconfined joint} \]

perpendicular to the direction of the shear forces considered.

The joint design for shear stress required the computation of:

\[ v_u = \frac{V_u}{A_{ch}} \quad (7.55) \]

And the maximum shear stress permitted by ACI-ASCE Committee 352 [2] was:

\[ v_u \leq 1.66\sqrt{f'_c} \quad (7.56) \]

If \( v_u > v_c \), then provide lateral shear reinforcement to carry the balance of the shear stress, equal to \( (v_u - v_c) \).

The area of the lateral reinforcement is determined from:

\[ A_v = \frac{V_s s_v}{f_y d} = \frac{(v_u - v_c)b_E s_v}{f_y} \quad (7.57) \]

Consequently, assuming the area of lateral reinforcement area is, \( A_v \), the spacing, \( s_v \), can be found from:

\[ \frac{A_v}{s_v} = \frac{V_s}{f_y d} = \frac{(v_u - v_c)b_E}{f_y} \quad (7.58) \]

where:

\[ b_E = \text{outside to outside width of column for the confined cross-section, as shown in Figure 7.7(c), mm.} \]
outside to outside of the ties, or column bars if no ties are used for the unconfined cross-section, as shown in Figure 7.7(c), mm.

The study conducted by Nirjar [1] on model specimens revealed a higher value for shear strength which was presented as:

\[
v_c \leq 0.588 \sqrt{f'_c(1 + 0.3\frac{P}{A_g})}
\]  

(7.59)

### 7.6.2.3. Design of the Reinforcement

The design of reinforcement should be checked for the critical section and should be taken at the face of the concrete core in the column, i.e. at the outside face of the column bars or ties. The main reinforcement should not be spliced within the joint and a standard hook, usually of 90°, should be provided by a standard hook if the joint has insufficient length. The anchorage in a joint is considered to develop the following stress for \( f'_c \) and \( f_y \), in N/mm²:

\[
f_h = 60(1 - 0.012d_b)\psi \sqrt{f'_c} \leq f_y
\]  

(7.60)

where:

- \( f_h \) = stress developed by a standard hook, N/mm².
- \( d_b \) = bar diameter, mm.
- \( \psi \) = 1.0; if any are of the following conditions (i) and (iii) is not satisfied.
- \( \psi \) = 1.4; if the followings conditions are satisfied.
- (i) bar is 36 mm in diameter or smaller.
(ii) side concrete cover normal to the plane of the hook bar is not less than 65 mm.

(iii) cover on the exterior beyond the bend is not less than 50 mm, and the exterior is contained within the confined core.

\[ \psi = 1.8; \text{ if in addition to the above conditions (i) and (iii), the following condition is also satisfied:} \]

(iv) that the joint is confined by closed ties at a maximum spacing of three times the diameter of the anchored bar.

Although ACI code of 1983 [119] does not make a distinction between the top and bottom bars, the computation of the reinforcement length required the following steps, as shown in Figure 7.7(b):

Anchorage length required: \[ L_{ip} = \frac{0.02 A_h f_y}{\sqrt{f_c}} \] (7.61)

Equivalent length provided by a hook: \[ L_{ip} = \frac{0.02 A_h f_h}{\sqrt{f_c}} \] (7.62)

Required straight length: \[ L_r = L_{ip} - L_{ip} = \frac{0.02 A_h (f_y - f_h)}{\sqrt{f_c}} \] (7.63)

Such that \( L_r \) is not less than \( 4d_b \) or 10 mm which is ever the greater.

Allowed required straight length: \[ L_s = \frac{L_r}{\psi} = \frac{0.02 A_h (f_y - f_h)}{\psi \sqrt{f_c}} \] (7.64)
Figure 7.7. Beam-Column Joint Requirements.
7.6.3. ACI-ASCE Committee 352 (1985)

The revision provided by the ACI-ASCE committee 352 [3] presented a substantial amount of updating information that was based on extensive laboratory testing and field studies. The recommendations are provided to determine the correct joint size and to design the longitudinal reinforcement at the intersection of the beam column joint for cast in place concrete. The recommendations are written mainly to satisfy the strength as well as the ductility requirements of the joint. In general, the recommendations provide a summary to focus on the joint behaviour and ensure realistic structural performance under specified loading conditions.

The committee's confinement recommendations, for beams framing into four sides of a joint, are relevant if the beam width is at least (3/4) of the column width of the intersected column face and does not leave more than 100 mm of the column face exposed on either side of the beam. Where there are only two beams framing into two opposite sides of a joint, an adequate confinement can be assumed in the direction of the beams if the beam widths are at least (3/4) the column width and no more than 100 mm of the column concrete is exposed on either side of the beams. For the other direction, lateral reinforcement must be provided to confine the joint. In the presence of a third beam perpendicular to the direction of the two beams, the requirements for lateral reinforcement remain the same.

7.6.3.1. Revised Axial or Compression Load Design Requirements

The reinforcement steel ratios recommended by the committee are confirmed by the ACI code [19] of 1983. Two layers of lateral reinforcement are recommended at the top and bottom levels of the beam longitudinal reinforcement of the deepest beam framing into the joint, such that the centre-to-centre spacing does not exceed 300 mm. The requirements for the rectangular hoop reinforcement ratio, \( \rho_r \), are within the following range:
For spiral reinforcement the requirement is the same as in equation (7.52) that was specified by the ACI-ASCE committee of 352-1976 [2].

7.6.3.2. Revised Shear Design Requirements

A beam-column joint which is subjected to moments as a result of external loads, will develop diagonal tension stresses from the normal and shearing forces and consequently patterns of diagonal cracking appear. The approach used by the committee was to limit the shear force on a horizontal plane through the joint to a value established by tests. The basis of the design was:

\[ V_u \leq \phi_s V_n \]  \hspace{1cm} (7.66)

where:

\[ V_u = \text{ultimate shear force at the cross-section, } N / mm^2. \]
\[ V_n = \text{nominal shear strength of a joint, } N / mm^2. \]
\[ \phi_s = \text{factor of safety equal to 0.85.} \]

The nominal shear strength of joint is given by:

\[ V_n = 0.083 \gamma \sqrt{f_c b_j h_c} \]  \hspace{1cm} (7.67)

where:

\[ b_j = \text{effective joint width and depends on the lateral width of the beams framing into the column as well as the lateral width of the column. In this study } b_j = 0.5(b + b_c), \text{ } mm. \]
\[ b_c = \text{width of the column, } mm. \]
\[ h_c = \text{depth of column in the direction of the load being considered, mm.} \]

\[ \gamma = \text{coefficient which depends on the confinement of the joint provided by the beam framing into it, and can be classified as:} \]

- interior joint = 24.
- exterior joint = 20.
- corner joint = 15.

7.6.3.3. **Revised Design for the Development Length of the Reinforcement**

The flexural reinforcement in the beam which is continuous through an interior joint is unlikely to pull-out due to the continuation of the reinforcement. Problems arise in the exterior joints due to the discontinuation of the beam which produces an anchorage problem. The critical section for the design of a joint is taken at the face of the column. The minimum development length, \( L_{dh} \), should not be less than \( 3d_b \) or 150 mm. For a development length that is terminated as a standard hook, the recommendation for SI system is:

\[
L_{dh} = \frac{f_y d_b}{4.2 \sqrt{f_c}} \tag{7.68}
\]

For a development length that terminates as a straight bar, the minimum requirement is \( (0.058d_b f_y) \) and the formula recommended is:

\[
L_d = \frac{f_y A_b}{52.7 \sqrt{f_c}} \quad \text{in the SI system} \tag{7.69}
\]

where:
294

\[ L_{dh} = \text{development length for a hook bar measured from the critical section to the outside end of the hook, } mm. \]

\[ L_d = \text{development length for a straight bar, } mm. \]

7.7. PREPARATORY STAGE FOR MODEL SPECIMEN EVALUATION

The generation of training examples throughout the numerical analysis is guided by the experimental results information from 34 specimens was used to construct a simulated model to deliver a correct diagnosis based on the selection of several possibilities. Three types of concrete strength, 20, 30 and 45 \( N/m^2 \) were selected with the same geometrical structural shape. The beam reinforcement diameters selected were 6 and 12 \( mm \) while the number of bars selected in the tension zone was 2 and 4. The column's main reinforcement diameters were 6 and 12 \( mm \), the number of bars was 4 and 5 and column loads of 10 and 60 % of the ultimate load, \( P_u \), were selected. The joint lateral reinforcement diameters selected were 3 and 4.5 \( mm \) at spacings of 40 and 80 \( mm \) respectively. The criteria of selecting these parameters were based on the information provided by Nirjar [1] and by considering the minimum and maximum reinforcement and spacing as criteria for implementation.

To simulate such work on a computer, a FORTRAN program was implemented to perform all the possible combinations. A possible combination of 13056 models was established. By introducing code limitations on the minimum, balance and maximum ratios of reinforcement the number of possibilities dropped to 8704 specimens which is obviously less than the actual number tested. The training file has 7739 patterns while the test file has 965 patterns. The program was run for the criteria to calculate the ultimate shear stress, \( \sigma_u \), shear stress carried by the concrete, \( \tau_c \), and the shear carried by the stirrups, \( \tau_s \). Table 7.5 shows the results provided by neural networks for the 34 specimens. The errors were compared to the results provided by Nirjar [1]. Values for shear strength and the shear carried by the concrete showed good correlation, however large errors were obtained for the amount of shear carried by the
Table 7.5. Preparatory Stage Comparisons between the Shear Strength of the Model Specimens by Nirjar [1] and Neural Networks.

<table>
<thead>
<tr>
<th>Specimens</th>
<th>$P_u , kN$</th>
<th>$P_a , kN$</th>
<th>Nirjar [1] $v_u , N/\text{mm}^2$</th>
<th>$v_c , N/\text{mm}^2$</th>
<th>$v_s , N/\text{mm}^2$</th>
<th>Neural Networks $v_u , N/\text{mm}^2$</th>
<th>$v_c , N/\text{mm}^2$</th>
<th>$v_s , N/\text{mm}^2$</th>
<th>%Error $= \text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \times 100$</th>
<th>For</th>
<th>For</th>
<th>For</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN₁</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>2.73</td>
<td>4.31</td>
<td>1.77</td>
<td>7.77</td>
<td>4.01</td>
<td>22.92</td>
<td></td>
</tr>
<tr>
<td>NN₂</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>5.47</td>
<td>1.44</td>
<td>2.72</td>
<td>4.90</td>
<td>1.78</td>
<td>8.11</td>
<td>10.42</td>
<td>23.61</td>
<td></td>
</tr>
<tr>
<td>NN₃</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>6.30</td>
<td>1.44</td>
<td>2.72</td>
<td>5.59</td>
<td>1.79</td>
<td>8.11</td>
<td>11.27</td>
<td>24.31</td>
<td></td>
</tr>
<tr>
<td>NN₄</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>7.03</td>
<td>1.44</td>
<td>2.72</td>
<td>6.36</td>
<td>1.80</td>
<td>8.11</td>
<td>9.53</td>
<td>25.00</td>
<td></td>
</tr>
<tr>
<td>NN₅</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>7.70</td>
<td>1.44</td>
<td>2.72</td>
<td>7.20</td>
<td>1.80</td>
<td>8.11</td>
<td>6.49</td>
<td>25.00</td>
<td></td>
</tr>
<tr>
<td>NN₆</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>8.31</td>
<td>1.44</td>
<td>2.73</td>
<td>8.07</td>
<td>1.80</td>
<td>7.77</td>
<td>2.89</td>
<td>25.00</td>
<td></td>
</tr>
<tr>
<td>NM₇</td>
<td>380.99</td>
<td>38.10</td>
<td>2.96</td>
<td>4.80</td>
<td>1.44</td>
<td>2.73</td>
<td>4.37</td>
<td>1.77</td>
<td>7.77</td>
<td>8.96</td>
<td>22.92</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2.96</td>
<td>4.74</td>
<td>1.44</td>
<td></td>
<td>2.73</td>
<td>4.36</td>
<td>1.77</td>
<td></td>
<td>7.77</td>
<td>8.02</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td>NM</td>
<td>8</td>
<td></td>
<td>364.37</td>
<td>34.25</td>
<td>2.96</td>
<td>4.74</td>
<td>1.44</td>
<td></td>
<td>2.73</td>
<td>4.36</td>
<td>1.77</td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>9</td>
<td></td>
<td>342.49</td>
<td>34.25</td>
<td>2.96</td>
<td>4.66</td>
<td>1.44</td>
<td></td>
<td>2.73</td>
<td>4.34</td>
<td>1.77</td>
<td></td>
</tr>
<tr>
<td>NM</td>
<td>10</td>
<td></td>
<td>311.00</td>
<td>31.10</td>
<td>2.96</td>
<td>4.55</td>
<td>1.44</td>
<td></td>
<td>2.73</td>
<td>4.32</td>
<td>1.77</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>11</td>
<td></td>
<td>380.99</td>
<td>190.50</td>
<td>2.96</td>
<td>8.58</td>
<td>1.44</td>
<td></td>
<td>2.73</td>
<td>8.35</td>
<td>1.80</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>12</td>
<td></td>
<td>364.37</td>
<td>182.18</td>
<td>2.96</td>
<td>8.42</td>
<td>1.44</td>
<td></td>
<td>2.73</td>
<td>8.13</td>
<td>1.80</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>13</td>
<td></td>
<td>342.49</td>
<td>171.25</td>
<td>2.96</td>
<td>8.20</td>
<td>1.44</td>
<td></td>
<td>2.72</td>
<td>7.83</td>
<td>1.80</td>
<td></td>
</tr>
<tr>
<td>NO</td>
<td>14</td>
<td></td>
<td>311.00</td>
<td>155.50</td>
<td>2.96</td>
<td>7.87</td>
<td>1.44</td>
<td></td>
<td>2.72</td>
<td>7.41</td>
<td>1.80</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>15</td>
<td></td>
<td>294.37</td>
<td>29.44</td>
<td>1.65</td>
<td>4.49</td>
<td>1.44</td>
<td></td>
<td>1.59</td>
<td>4.41</td>
<td>1.75</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>16</td>
<td></td>
<td>294.37</td>
<td>29.44</td>
<td>4.69</td>
<td>4.49</td>
<td>1.44</td>
<td></td>
<td>5.43</td>
<td>4.41</td>
<td>1.75</td>
<td></td>
</tr>
<tr>
<td>NP</td>
<td>17</td>
<td></td>
<td>294.37</td>
<td>29.44</td>
<td>6.07</td>
<td>4.49</td>
<td>1.44</td>
<td></td>
<td>6.59</td>
<td>4.48</td>
<td>1.74</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.5. (continued)
Table 7.5. (continued)

<table>
<thead>
<tr>
<th></th>
<th>294.37</th>
<th>29.44</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$NP_{18}$</td>
<td>294.37</td>
<td>29.44</td>
<td>7.17</td>
<td>4.49</td>
<td>1.44</td>
<td>6.94</td>
<td>4.48</td>
<td>1.73</td>
<td>3.21</td>
<td>0.22</td>
<td>20.14</td>
</tr>
<tr>
<td>$NQ_{19}$</td>
<td>294.37</td>
<td>176.62</td>
<td>1.65</td>
<td>8.31</td>
<td>1.44</td>
<td>1.60</td>
<td>8.07</td>
<td>1.77</td>
<td>3.03</td>
<td>2.89</td>
<td>22.92</td>
</tr>
<tr>
<td>$NQ_{20}$</td>
<td>294.37</td>
<td>176.62</td>
<td>4.69</td>
<td>8.31</td>
<td>1.44</td>
<td>5.51</td>
<td>8.24</td>
<td>1.80</td>
<td>17.48</td>
<td>0.84</td>
<td>25.00</td>
</tr>
<tr>
<td>$NQ_{21}$</td>
<td>294.37</td>
<td>176.62</td>
<td>6.07</td>
<td>8.31</td>
<td>1.44</td>
<td>6.65</td>
<td>8.19</td>
<td>1.78</td>
<td>9.56</td>
<td>1.44</td>
<td>23.61</td>
</tr>
<tr>
<td>$NQ_{22}$</td>
<td>294.37</td>
<td>176.62</td>
<td>7.17</td>
<td>8.31</td>
<td>1.44</td>
<td>6.98</td>
<td>8.06</td>
<td>1.78</td>
<td>2.65</td>
<td>3.01</td>
<td>23.61</td>
</tr>
<tr>
<td>$NR_{23}$</td>
<td>342.49</td>
<td>34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>1.08</td>
<td>6.85</td>
<td>4.59</td>
<td>1.08</td>
<td>0.29</td>
<td>1.50</td>
<td>0.00</td>
</tr>
<tr>
<td>$NR_{24}$</td>
<td>342.49</td>
<td>34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>0.86</td>
<td>6.82</td>
<td>4.65</td>
<td>0.85</td>
<td>0.73</td>
<td>0.21</td>
<td>1.16</td>
</tr>
<tr>
<td>$NR_{25}$</td>
<td>343.49</td>
<td>34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>0.43</td>
<td>6.79</td>
<td>4.67</td>
<td>0.44</td>
<td>1.16</td>
<td>0.21</td>
<td>2.33</td>
</tr>
<tr>
<td>$NR_{26}$</td>
<td>342.49</td>
<td>34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>0.43</td>
<td>6.79</td>
<td>4.67</td>
<td>0.44</td>
<td>1.16</td>
<td>0.21</td>
<td>2.33</td>
</tr>
<tr>
<td>$NS_{27}$</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>2.73</td>
<td>4.31</td>
<td>1.77</td>
<td>7.77</td>
<td>4.01</td>
<td>22.92</td>
</tr>
</tbody>
</table>
Table 7.5. (continued)

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NS 28</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>2.73</td>
<td>4.31</td>
<td>1.77</td>
<td>7.77</td>
<td>4.01</td>
<td>22.92</td>
</tr>
<tr>
<td>NS 29</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>2.73</td>
<td>4.31</td>
<td>1.77</td>
<td>7.77</td>
<td>4.01</td>
<td>22.92</td>
</tr>
<tr>
<td>NS 30</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>2.73</td>
<td>4.31</td>
<td>1.77</td>
<td>7.77</td>
<td>4.01</td>
<td>22.92</td>
</tr>
<tr>
<td>NT 31</td>
<td>420.07</td>
<td>210.03</td>
<td>2.92</td>
<td>10.96</td>
<td>1.44</td>
<td>2.70</td>
<td>11.06</td>
<td>1.79</td>
<td>7.53</td>
<td>0.91</td>
<td>24.31</td>
</tr>
<tr>
<td>NT 32</td>
<td>336.27</td>
<td>168.14</td>
<td>2.94</td>
<td>8.79</td>
<td>1.44</td>
<td>2.70</td>
<td>8.49</td>
<td>1.80</td>
<td>8.16</td>
<td>3.41</td>
<td>25.00</td>
</tr>
<tr>
<td>NT 33</td>
<td>252.47</td>
<td>126.24</td>
<td>2.99</td>
<td>6.60</td>
<td>1.44</td>
<td>2.77</td>
<td>6.04</td>
<td>1.79</td>
<td>7.36</td>
<td>8.48</td>
<td>24.31</td>
</tr>
<tr>
<td>NT 34</td>
<td>210.57</td>
<td>105.29</td>
<td>3.03</td>
<td>5.50</td>
<td>1.44</td>
<td>2.85</td>
<td>5.06</td>
<td>1.78</td>
<td>5.94</td>
<td>8.00</td>
<td>23.61</td>
</tr>
</tbody>
</table>
stirrups. These values will be adjusted in chapter 8 to demonstrate the capability of neural networks in reducing the errors.

The network selected to execute this problem is based on the training data provided by the FORTRAN program which resulted in a selection of network architecture of 50, 20 and 10 PE's in first, second and third hidden layers respectively. The input layer consists of 15 PE's \((b, d, f_c, \rho, x_u, P_a, P_u, M_u, b_e, d_e, V_{col}, A_v, f_{\mu}, s_v, b_c)\) which are combinations of variables that established the relationships to provide the shear strength characteristics for the model specimens. The output layer consists of three PE's, mainly the ultimate stress, \(v_u\), the shear stress of the concrete, \(v_c\), and shear carried by the stirrups, \(v_s\). The EDBD learning algorithm is selected with the sigmodal function and the criteria for convergence was set to 0.5 \% in the RMS instrument. The network ceased learning at 203870 cycle with confusion matrices of 0.9999, 0.9980 and 0.9998 for the trained network while the tested network has confusion matrices of 0.9998, 0.9980 and 0.9998 respectively. The classification rates for the trained and tested networks are 1.0 and 0.9393 respectively. Figure 7.8 provides the trained and test networks with the architecture and topology.

The predicted equations for the shear stress, shear concrete stress and the shear carried by the stirrup reinforcement are respectively:

\[
y_{97}^{out,p} = x_{97}^{out,p} \times (8.733) + (-0.117) \tag{7.70}
\]

\[
x_{97}^{out,p} = \frac{1.0}{1.0 + e^{-x_{97}}} \tag{7.71}
\]

\[
y_{98}^{out,p} = x_{98}^{out,p} \times (16.40) + (0.10) \tag{7.72}
\]

\[
x_{98}^{out,p} = \frac{1.0}{1.0 + e^{-x_{98}}} \tag{7.73}
\]

\[
y_{99}^{out,p} = x_{99}^{out,p} \times (2.883) + (-0.147) \tag{7.74}
\]
Figure 7.8. Shear Strength Parameters by Neural Networks for the Preparatory Stage.
\[ x_{99}^{out,p} = \frac{1.0}{1.0 + e^{-\Sigma x_{99}}} \]  \hspace{1cm} (7.75)

in which \( y_{97}^{out,p}, y_{98}^{out,p} \) and \( y_{99}^{out,p} \) are the final shears predicted by the networks in \( N/mm^2 \), while \( x_{97}^{out,p}, x_{98}^{out,p}, x_{99}^{out,p} \) shear are the predicted within the network.

### 7.8. PREPARATORY AND RENEW STAGES FOR SHEAR FORCES

ACI-ASCE Committee 352 of 1985 [3] expressed the joint strength in terms of the shear force, \( V_u \), instead of the shear stress contributed by the concrete. To provide adequate comparisons for the shear force strength at both the preparatory and renew stages, the joint was evaluated in terms of the shear force, \( V_i \). Implementation of the ACI-ASCE 352 recommendations [3] depends on the strength criteria of the members attached to the joint. In the case of the preparatory stage, the strength of member and the determination of forces with moments for the beams are based on the strength method discussed in chapter five and proposed by Nirjar [1]. Evaluation for the renew stage depends on the ACI code of 1989 [81] for flexural members. An ultimate and maximum strain of 0.003 and 0.002 respectively were selected for the analysis.

The preparatory and renew stages both investigate the evaluation of the shear force in a joint. Computation involved the determination of the following: concrete cylinder strength; 20, 30, and 45 \( N/mm^2 \); diameter of the main bars and the number of tension bars in the beam; the number and diameter of the bars in the column; the percentage of ultimate column load; the diameter and spacing of the lateral reinforcement in the joint for the 34 specimen. All these possible combinations are shown in Figure 7.9 for \( f'_c = 20 N/mm^2 \) only. A similar procedure was adopted for \( f'_c = 30 \) and \( f'_c = 45 N/mm^2 \). Also, the steel strength, \( f_y \), for bars of diameter 3 and 4.5 \( mm \) were 242 and 272 \( N/mm^2 \) respectively.
Figure 7.9. Possible Combinations for Joint Shear Determination at Preparatory and Renew Stages.
The network architecture selected for both the preparatory and renew stages consisted of 13 PE's in the input layer \((b, d, f_c, \rho, x_u, M_u, b_E, d_c, V_{col}, A_s, f_v, s_v, b_c)\). Three hidden layers were selected, with 50, 20 and 10 PE's in the first, second and third hidden layers with learning rates of 0.30, 0.25 and 0.20 respectively. The EDBD was implemented with a sigmoidal transfer function, and a momentum of 0.40 was retained with an epoch size of 16. The criteria for convergence at the RMS for both stages were maintained at 0.05% in order to make comparisons. The preparatory stage training consisted of 7739 patterns and the test consisted of 965 patterns. The network ceased learning at 730290 with a confusion matrix of 1.0 and classification rates of 1.0. The training for the renew stage was 15481 and the test file had 1930 patterns. The learning ceased at a higher cycle of 796911 with confusion matrix of 1.0 and classification rates of 1.0. The results obtained for both stages are shown in Table 7.6. These demonstrate the ability of neural networks to predict the shear force by studying the alternative numerical models of a joint and deliver a correct diagnosis based on a set of training and test patterns through forward and backward propagations gained throughout the weight adjustments. These patterns were generated by FORTRAN programming by taking into consideration the strength criteria limitations imposed by the code of practice. The full interconnected networks at the preparatory and renew stages are shown in Figures 7.10 and 7.11. The predicted shear force in Newtons at the preparatory stage is given by:

\[
y_{95}^{out,p} = x_{95}^{out,p} \times (60967.45) + (-269.340)
\]

(7.76)

\[
x_{95}^{out,p} = \frac{1.0}{1.0 + e^{-\Sigma x_{95}}}
\]

(7.77)

The full equations are provided on disk at the end of the thesis. The renew stage provided a relationship for the shear force in Newtons in the form:

\[
y_{95}^{out,r} = x_{95}^{out,r} \times (65276.40) + (-1131.129)
\]

(7.78)
Table 7.6. Shear Force Strength for the Model Specimens at Preparatory and Renew Stages by Neural Networks.

<table>
<thead>
<tr>
<th>Specimens</th>
<th>$f'_c$ N/mm$^2$</th>
<th>$\rho$ %</th>
<th>Nirjar [1] $V_u$ Newtons</th>
<th>Neural Networks $V_u$ Newtons W.R.T Preparatory Stage</th>
<th>Neural Networks $V_u$ Newtons W.R.T Renew Stage</th>
<th>% Error = $\frac{Abs(\text{Network} - \text{Desired})}{\text{Desired}} \times 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NN_1$</td>
<td>30</td>
<td>1.28</td>
<td>21613.96</td>
<td>20357.93</td>
<td>20604.37</td>
<td>5.81</td>
</tr>
<tr>
<td>$NP_{15}$</td>
<td>30</td>
<td>0.72</td>
<td>12026.15</td>
<td>11993.21</td>
<td>11869.82</td>
<td>0.27</td>
</tr>
<tr>
<td>$NP_{16}$</td>
<td>30</td>
<td>2.00</td>
<td>34242.18</td>
<td>39858.11</td>
<td>39195.47</td>
<td>16.40</td>
</tr>
<tr>
<td>$NP_{17}$</td>
<td>30</td>
<td>2.55</td>
<td>44298.28</td>
<td>48082.03</td>
<td>47805.23</td>
<td>8.54</td>
</tr>
<tr>
<td>$NP_{18}$</td>
<td>30</td>
<td>2.99</td>
<td>52343.23</td>
<td>50798.46</td>
<td>50906.56</td>
<td>2.95</td>
</tr>
</tbody>
</table>
Table 7.6. (continued)

<table>
<thead>
<tr>
<th>$NQ_{20}$</th>
<th>30</th>
<th>2.00</th>
<th>34242.18</th>
<th>39858.11</th>
<th>39195.47</th>
<th>16.40</th>
<th>14.47</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NQ_{21}$</td>
<td>30</td>
<td>2.55</td>
<td>44298.28</td>
<td>48082.03</td>
<td>47805.23</td>
<td>8.54</td>
<td>7.92</td>
</tr>
<tr>
<td>$NQ_{22}$</td>
<td>30</td>
<td>2.99</td>
<td>52343.23</td>
<td>50798.46</td>
<td>50906.56</td>
<td>2.95</td>
<td>2.74</td>
</tr>
<tr>
<td>$NR_{23}$</td>
<td>30</td>
<td>2.99</td>
<td>50136.61</td>
<td>50191.02</td>
<td>50195.71</td>
<td>0.11</td>
<td>0.12</td>
</tr>
<tr>
<td>$NT_{31}$</td>
<td>45</td>
<td>1.28</td>
<td>21291.59</td>
<td>20015.09</td>
<td>20328.75</td>
<td>6.00</td>
<td>4.52</td>
</tr>
<tr>
<td>$NT_{32}$</td>
<td>35</td>
<td>1.28</td>
<td>21472.77</td>
<td>20200.48</td>
<td>20473.97</td>
<td>5.93</td>
<td>4.65</td>
</tr>
<tr>
<td>$NT_{33}$</td>
<td>25</td>
<td>1.28</td>
<td>21819.08</td>
<td>20605.74</td>
<td>20782.57</td>
<td>5.56</td>
<td>4.75</td>
</tr>
<tr>
<td>$NT_{34}$</td>
<td>20</td>
<td>1.28</td>
<td>22142.21</td>
<td>21051.30</td>
<td>21031.45</td>
<td>4.93</td>
<td>5.02</td>
</tr>
</tbody>
</table>
Figure 7.10. Joint Shear Force Strength at Preparatory Stage.
Figure 7.11. Joint Shear Force Strength at Renew Stage.
Design philosophy must consider the criteria for serviceability factors since characteristics such as excessive deflection and detrimental cracking may cause failure. The beam-column joint is subjected to several loading conditions and it is essential to alleviate the diagonal tension by designing the joint to reach its yield point before the development of diagonal cracking. Although, this consideration is an approximate procedure, it provides an initial check for serviceability. To overcome this stage the total shear force in the joint region provided by the beam steel is represented as:

\[ V = A_s f_y \]  \hspace{1cm} (7.80)

Thus the serviceability stress criterion is:

\[ \frac{A_s f_y}{b_c d_c} \leq v_c \]  \hspace{1cm} (7.81)

or limiting the steel reinforcement steel ratio:

\[ \rho f_y \leq \frac{b_d d_c}{b d} v_c \]  \hspace{1cm} (7.82)

or in general:

\[ \rho f_y \leq 0.67 \sqrt{(f_c)^2 + f_c \frac{P_o}{A_g} \frac{b_c d_c}{b d}} \]  \hspace{1cm} (7.83)

The serviceability criterion expressed in terms of concrete strength, \( f_c \), is:

\[ \rho f_y \leq 0.44 \sqrt{f_c (1 + 0.3 \frac{P_o}{A_g}) \frac{b_c d_c}{b d}} \]  \hspace{1cm} (7.84)
CHAPTER 8

EVALUATION FOR THE FINAL PREDICTIVE STAGE

8.1. INTRODUCTION

A general analysis of the beam-column joint was investigated for several member parameters under flexural strength, moment-curvature and shear strength using neural networks. The study concentrated on two stages; the preparatory and renew stages. The requirements to establish a new proposal of carrying out neural network simulations to assist with the experimental work were based on a selection of fixed dimensions of the experimental model, the identification of materials properties, the amounts of steel reinforcement, the type of loading, loading conditions and strength requirements.

Table 8.1 gives details the parameters used in the experimental work where the shaded areas indicate the parameter values used as criteria for performing the experimental work simulation process. The experimental simulation test was carried out using the minimum, intermediate and maximum properties of the cylinder concrete strength. The smallest and largest bar diameters and the number of bars were used as criteria for the analysis so as not to exceed Code requirements. The maximum and minimum permissible loads on the column were used in the analysis, together with the lower and upper boundaries for parameter link spacing. Having established these requirements, the simulation process is carried out on the extreme values of the parameters by considering all possible combinations. For each concrete cylinder strength, an analysis was carried out by FORTRAN to examine all the
possible combinations based on these selected parameters. Having satisfied the
requirements and obtained the desired data from the analysis, neural network models
were set up for each individual task. The results of the experimental findings,
together with the analytical results established within the experimental work was then
merged with the numerical analysts to provide enhancement for the data required by
the neural network. The results were separated into two main categories, one for
training and the other for testing. The testing results were 9 to 10% of the data which
enhanced the performance of the network. The testing data was passed to the learned
network to observe the behaviour of the network on data that had not been seen
before. From the network performance instruments for the selected criteria, the
results should provide a knowledge of the network performance. For acceptable
network performance, the reliability of the trained and tested network is verified by
passing the data once and sequentially to predict selected requirements. The reliable
network can be then deployed as a part of the system development. The deployment
system can be used either as interactive by entering the data directly or by creating a
predictive file that can be passed to the learned and tested network to obtain results
directly or by converting the results to a C subroutine that can be linked to a main C
code.

The renew stage is then applied to observe the effect of for example new
experimental findings, any revision of the Code of Practice, improvements in
building materials and the introduction of recent publications which update the
preparatory stage. The new knowledge encompasses huge amounts of information,
theoretical understanding and improvements in the numerical analysis which will
eventually improve the analysis.

The previous two stages have laid down strategies that work and can be adopted
for predicting several variables which usually can be obtained from tests and
validated by analytical formulation. The final stage proposed in this chapter attempts
to carry out a procedure similar to that adopted before by relying entirely on the updated code of practice and any new information relevant to the study. This can be also viewed as a secondary objective by performing preliminary experimental work before any real experimental work starts to provide inside information of the materials, loading and geometrical relationship.

This chapter investigates the ultimate moment, the moment-curvature relationship and shear strength of the beam-column joint for the 34 specimens tested by Nirjar [1] were investigated by considering only the model dimensions, as well the material properties and the loading conditions.
Table 8.1. Variations of Parameters Used in the Computational Process.

<table>
<thead>
<tr>
<th>$f_c$ ($N/mm^2$)</th>
<th>$f_y$ ($N/mm^2$)</th>
<th>Column load ($N$)</th>
<th>Longitudinal Column Reinforcement</th>
<th>Lateral Reinforcement in Column and Joint</th>
<th>Tension Reinforcement in Beam</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Number</td>
<td>Dia. size ($mm$)</td>
<td>Spacing</td>
</tr>
<tr>
<td>20</td>
<td>242</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td>60</td>
</tr>
<tr>
<td>25</td>
<td>272</td>
<td>20</td>
<td>5</td>
<td>8</td>
<td>80</td>
</tr>
<tr>
<td>30</td>
<td>304</td>
<td>30</td>
<td>10</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td></td>
<td>40</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>45</td>
<td></td>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
8.2. EVALUATION FOR THE FINAL STAGE

The general procedure in this section is to evaluate the essential strength and behaviour requirements of specimens evaluated previously by relying entirely on the ACI Code of Practice and essential updated information from journals. Only the beam-column joint dimensions, the properties of the materials and the number and type of reinforcement in the beams, column and joints are used as input data.

8.2.1. Final Stage for Ultimate Moment Computation

The moment computation evaluated in section 5.5.3. at the preparatory stage, by adopting Nirjar's [1] procedure, was in complete agreement with both the analytical and experimental results. This process was then extended at the renew stage to include the ACI Code 318-89 [81] to enhance and update the data evaluated in section 5.5.4. Both sets of data obtained from the analytical and experimental results of Nirjar [1] were included of the renew stage together with the provisions of the ACI code 318-89 [81]. The methodology adopted in the previous stage demonstrated the ability of the neural network to map the procedure from one domain into other by considering the boundary as criteria. The analysis carried out for the final stage conformed to the ACI code 318-89 [81] and was based on extensive experimental results, analytical formulations and progress in research on reinforced concrete structures.

The architecture, topology and neurodynamics selected to perform this task was the same as that adopted in the preparatory stage of section 5.5.3. The number of input, hidden and output layers were kept constant. The training and test data were obtained entirely from the analysis recommended by the ACI code 318-89 [81], as specified in section 5.5.2. The training contained 243 patterns and the test 29 patterns. The Extended-Delta-Bar-Delta learning rule was used as in the preparatory stage and the network ceased learning at 184437 with \( \text{RMS} \) convergence criteria of
0.2%. The confusion matrices and the classification rates for both the learned and tested networks were 0.9999 and 1.0 respectively. The results obtained at the preparatory stage by adopting Nirjar's procedure [1] and the final stage by adopting the ACI Code are shown in Tables 8.2, 8.4 and 8.7 for concrete strengths of 20, 30 and 45 $N/mm^2$ for data seen by the network. Tables 8.3, 8.5 and 8.6 gives results for concrete strengths of 25, 35 and 40 $N/mm^2$ for data not seen by the network. Comparisons of the final results are also included in the above tables.

The equations to compute the ultimate moment were obtained for the final stage were by converting the knowledge into a subroutine code which is included on a disk at the end of the thesis in the form:

\[ y^{out,f}_{134} = x^{out,f}_{134} \times (8.6) + (-0.04) \]  
\[ x^{out,f}_{134} = \frac{1.0}{1.0 + e^{-\Sigma x_{134}}} \]  

\( x^{out,f}_{134} \) is the moment predicted within the neural network at the final stage.

\( y^{out,f}_{134} \) is the scaled moment by network output at the final stage, $kN-m$.

The networks obtained from the above procedure are shown in Figures 8.1(a) and (b) for trained and tested networks respectively.
Table 8.2. Comparison of the Ultimate Moment Produced by Preparatory and Final Stages. $f_c = 20 \text{ N/mm}^2$. %Error = $\text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \times 100$

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>Preparatory Stage</th>
<th>Final Stage</th>
<th>ACI Code 318-89 [81]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_c$, $N/mm^2$</td>
<td>$f_{av}$, $N/mm^2$</td>
<td>$x_{ad}$, mm</td>
<td>$gd$, mm</td>
</tr>
<tr>
<td>20</td>
<td>0.0072</td>
<td>14.26</td>
<td>6.61</td>
<td>1.69</td>
</tr>
<tr>
<td></td>
<td>0.0108</td>
<td>21.39</td>
<td>9.91</td>
<td>2.44</td>
</tr>
<tr>
<td></td>
<td>0.0144</td>
<td>28.51</td>
<td>13.51</td>
<td>3.15</td>
</tr>
<tr>
<td></td>
<td>0.0128</td>
<td>25.35</td>
<td>11.74</td>
<td>2.84</td>
</tr>
<tr>
<td></td>
<td>0.0192</td>
<td>38.02</td>
<td>17.62</td>
<td>3.99</td>
</tr>
<tr>
<td></td>
<td>0.0200</td>
<td>39.60</td>
<td>18.35</td>
<td>4.12</td>
</tr>
</tbody>
</table>

315
Table 8.3. Comparison of the Ultimate Moment Produced by Preparatory and Final Stages. $f_c = 25 \, N/mm^2$. \% Error = $Abs\left(1 - \frac{\text{Network}}{\text{Desired}}\right) \times 100$

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>$p = \frac{A_s}{bd}$</th>
<th>$f_{av}$</th>
<th>$x_u$</th>
<th>$d$</th>
<th>$g$</th>
<th>$M_u$</th>
<th>$M$</th>
<th>% Error</th>
<th>$f_{av}$</th>
<th>$x_u$</th>
<th>$d$</th>
<th>$g$</th>
<th>$M_u$</th>
<th>$M$</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0072</td>
<td>0.0108</td>
<td>0.0144</td>
<td>0.0128</td>
<td>25</td>
<td>0.0192</td>
<td>0.0256</td>
<td>0.0200</td>
<td>0.0288</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.50</td>
<td>17.25</td>
<td>22.99</td>
<td>20.44</td>
<td>19.94</td>
<td>30.66</td>
<td>40.88</td>
<td>31.94</td>
<td>45.99</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.6</td>
<td>6.9</td>
<td>9.2</td>
<td>8.18</td>
<td>12.26</td>
<td>16.35</td>
<td>12.77</td>
<td>12.39</td>
<td>18.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.72</td>
<td>2.52</td>
<td>3.28</td>
<td>2.95</td>
<td>4.24</td>
<td>5.40</td>
<td>5.40</td>
<td>4.39</td>
<td>5.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.71</td>
<td>2.36</td>
<td>3.28</td>
<td>2.84</td>
<td>4.53</td>
<td>5.67</td>
<td>4.71</td>
<td>4.71</td>
<td>6.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.58^{max}</td>
<td>6.35</td>
<td>0.61</td>
<td>3.73</td>
<td>6.84</td>
<td>5.00</td>
<td>7.29^{max}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.69</td>
<td>19.03</td>
<td>25.38</td>
<td>22.56</td>
<td>21.25</td>
<td>33.84</td>
<td>45.12</td>
<td>35.25</td>
<td>50.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.71</td>
<td>2.49</td>
<td>3.23</td>
<td>2.91</td>
<td>4.14</td>
<td>5.23</td>
<td>5.41</td>
<td>4.29</td>
<td>5.72</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.68</td>
<td>2.33</td>
<td>3.18</td>
<td>2.79</td>
<td>4.32</td>
<td>5.41</td>
<td>4.49</td>
<td>4.49</td>
<td>5.76</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.75</td>
<td>6.43^{max}</td>
<td>1.55</td>
<td>4.12</td>
<td>4.35</td>
<td>3.44</td>
<td>4.66</td>
<td>0.70^{max}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 8.4. Comparison of the Ultimate Moment Produced by Preparatory and Final Stages. $f_c = 30 \, N/mm^2$. 

$\% Error = \text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \times 100$

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>Preparatory Stage</th>
<th>Final Stage</th>
<th>Final Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$f_{av}$</td>
<td>$x_u$</td>
<td>$d$</td>
<td>$M_u$</td>
</tr>
<tr>
<td>0.0072</td>
<td>9.70</td>
<td>3.32</td>
<td>1.74</td>
<td>1.74</td>
</tr>
<tr>
<td>0.0108</td>
<td>14.55</td>
<td>4.99</td>
<td>2.57</td>
<td>2.39</td>
</tr>
<tr>
<td>0.0144</td>
<td>19.40</td>
<td>6.65</td>
<td>3.37</td>
<td>3.35</td>
</tr>
<tr>
<td>0.0128</td>
<td>17.24</td>
<td>5.91</td>
<td>3.02</td>
<td>2.89</td>
</tr>
<tr>
<td>0.0192</td>
<td>23.63</td>
<td>25.87</td>
<td>8.87</td>
<td>4.39</td>
</tr>
<tr>
<td>0.0256</td>
<td>34.49</td>
<td>11.82</td>
<td>5.68</td>
<td>6.01</td>
</tr>
<tr>
<td>0.0200</td>
<td>26.94</td>
<td>9.23</td>
<td>4.56</td>
<td>4.96</td>
</tr>
<tr>
<td>0.0288</td>
<td>38.80</td>
<td>13.30</td>
<td>6.29</td>
<td>6.38</td>
</tr>
</tbody>
</table>
Table 8.5. Comparison of the Ultimate Moment Produced by Preparatory and Final Stages. $f_c' = 35 \text{ N/mm}^2$. %Error = $\text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \times 100$

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$\rho = \frac{A_i}{bd}$</th>
<th>Preparatory Stage</th>
<th></th>
<th>Final Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N/mm^2$</td>
<td>$f_{av}$</td>
<td>$x_u$</td>
<td>$gd$</td>
<td>$M_u$</td>
</tr>
<tr>
<td>N/mm$^2$</td>
<td>N/mm</td>
<td>mm</td>
<td>mm</td>
<td>kN-m</td>
</tr>
<tr>
<td>0.0072</td>
<td>8.44</td>
<td>2.45</td>
<td>1.76</td>
<td>1.76</td>
</tr>
<tr>
<td>0.0108</td>
<td>12.66</td>
<td>3.67</td>
<td>2.61</td>
<td>2.43</td>
</tr>
<tr>
<td>0.0144</td>
<td>16.87</td>
<td>4.89</td>
<td>3.43</td>
<td>3.42</td>
</tr>
<tr>
<td>0.0128</td>
<td>15.00</td>
<td>4.35</td>
<td>3.07</td>
<td>2.94</td>
</tr>
<tr>
<td>35</td>
<td>0.0192</td>
<td>27.17</td>
<td>22.50</td>
<td>6.53</td>
</tr>
<tr>
<td>0.0256</td>
<td>30.00</td>
<td>8.70</td>
<td>5.87</td>
<td>6.22</td>
</tr>
<tr>
<td>0.0200</td>
<td>23.44</td>
<td>6.80</td>
<td>4.68</td>
<td>5.12</td>
</tr>
<tr>
<td>0.0288</td>
<td>33.5</td>
<td>9.79</td>
<td>6.53</td>
<td>6.58</td>
</tr>
</tbody>
</table>
Table 8.6. Comparison of the Ultimate Moment Produced by Preparatory and Final Stages.  $f'_c = 40 \text{ N/mm}^2$.  

\[
\text{% Error} = \frac{\text{Abs}(1 - \frac{\text{Network}}{\text{Desired}})}{100}
\]

<table>
<thead>
<tr>
<th>$f_c$</th>
<th>$\rho = \frac{A_s}{bd}$</th>
<th>Preparatory Stage</th>
<th></th>
<th>Final Stage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$f_{av}$</td>
<td>$x_u d$</td>
<td>$gd$</td>
<td>$M_u$</td>
</tr>
<tr>
<td>$\text{N/mm}^2$</td>
<td>$\text{N/mm}^2$</td>
<td>$\text{mm}$</td>
<td>$\text{mm}$</td>
<td>$\text{kN-m}$</td>
<td>$\text{kN-m}$</td>
</tr>
<tr>
<td>0.0072</td>
<td>7.50</td>
<td>1.81</td>
<td>1.77</td>
<td>1.80</td>
<td>1.69</td>
</tr>
<tr>
<td>0.0108</td>
<td>11.25</td>
<td>2.71</td>
<td>2.63</td>
<td>2.47</td>
<td>6.08</td>
</tr>
<tr>
<td>0.0144</td>
<td>11.00</td>
<td>3.62</td>
<td>3.48</td>
<td>3.47</td>
<td>0.29</td>
</tr>
<tr>
<td>0.0128</td>
<td>13.34</td>
<td>3.21</td>
<td>3.10</td>
<td>2.99</td>
<td>3.55</td>
</tr>
<tr>
<td></td>
<td>30.56</td>
<td>20.00</td>
<td>4.82</td>
<td>4.58</td>
<td>5.00</td>
</tr>
<tr>
<td>40</td>
<td>0.0192</td>
<td>26.67</td>
<td>6.43</td>
<td>6.01</td>
<td>6.35</td>
</tr>
<tr>
<td></td>
<td>0.0256</td>
<td>20.84</td>
<td>5.02</td>
<td>4.76</td>
<td>5.23</td>
</tr>
<tr>
<td></td>
<td>0.0200</td>
<td>30.00</td>
<td>2.72</td>
<td>6.70</td>
<td>6.70</td>
</tr>
<tr>
<td></td>
<td>0.0288</td>
<td>30.00</td>
<td>7.23</td>
<td>6.70</td>
<td>6.70</td>
</tr>
</tbody>
</table>
Table 8.7. Comparison of the Ultimate Moment Produced by Preparatory and Final Stages. $f'_c = 45 \text{ N/mm}^2$. %Error = $\text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \times 100$

<table>
<thead>
<tr>
<th>$f'_c$ (N/mm$^2$)</th>
<th>Preparatory Stage</th>
<th>Final Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Nirjar [1]</td>
<td>Neural Network</td>
</tr>
<tr>
<td></td>
<td>$f_{av}$</td>
<td>$x_d$</td>
</tr>
<tr>
<td></td>
<td>N/mm$^2$</td>
<td>mm</td>
</tr>
<tr>
<td>0.0072</td>
<td>6.78</td>
<td>1.32</td>
</tr>
<tr>
<td>0.0108</td>
<td>10.17</td>
<td>1.98</td>
</tr>
<tr>
<td>0.0144</td>
<td>13.56</td>
<td>2.64</td>
</tr>
<tr>
<td>0.0128</td>
<td>12.05</td>
<td>2.35</td>
</tr>
<tr>
<td></td>
<td>18.08</td>
<td>3.53</td>
</tr>
<tr>
<td>45</td>
<td>33.81</td>
<td></td>
</tr>
<tr>
<td>0.0192</td>
<td>24.10</td>
<td>4.70</td>
</tr>
<tr>
<td>0.0256</td>
<td>18.83</td>
<td>3.67</td>
</tr>
<tr>
<td>0.0200</td>
<td>27.12</td>
<td>5.29</td>
</tr>
<tr>
<td>0.0288</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

320
Figure 8.1. Trained and Tested Networks for the Ultimate Moment at Final Stage.
8.2.2. Moment-Curvature Relationship at the Final Stage

8.2.2.1. Moment-Curvature Relationship at Yield

The control strategies used to obtain the moment-curvature relationship for the final stage was based on the previous architecture, topology and neurodynamics which was established for the preparatory stage at yield in chapter 6. The training and test files at this stage are based entirely on the theoretical formulation by Park and Ruitong [99]. The numerical analysis is based on the ACI Code of Practice [81]. 243 trainings were presented to the network while the test file contained 29. The neural network process of weight adjustments continued for 57587 cycles when the network ceased learning. Sequentially the network is presented with the test data and not seen by the network to encourage adaptations. The computed and predicated values for the curvature and moments at yield stage are shown in Table 8.8. The percentage error obtained for the experimental results at the preparatory stage together with the comparisons obtained both numerically and with the help of the neural network are also shown. It should be noted that the specimens presented to the network have the same parameters as those considered by Nirjar [1] for his experimental investigation.

The network performance was measured by the instrumentations. These provided an excellent performance for both the training and test networks with confusion matrices of 1.0 respectively. Classification rates are also in complete agreement with a value of 1.0 for both the trained and the test networks. The full networks shown in Figure 8.2(a) and 8.2(b) are for the trained and tested networks respectively.

The general formulae generated with the assistance of NeuralWorks® [17] for the curvature and moment respectively at yield are:

\[
y_{78,y}^{out.f} = x_{78,y}^{out.f} \cdot (608.856) + 1950.375
\]  

(8.3)
\[ x_{78,y}^{\text{out},f} = \tanh(\sum x_{78,y}^f) \]  
(8.4)

\[ y_{79,y}^{\text{out},f} = x_{79,y}^{\text{out},f} \times (3033.731) + 4073.395 \]  
(8.5)

\[ x_{79,y}^{\text{out},f} = \tanh(\sum x_{79,y}^f) \]  
(8.6)

where \( y_{78,y}^{\text{out},f} \) and \( y_{79,y}^{\text{out},f} \) are the scaled curvature and moment respectively, and \( x_{78,y}^{\text{out},f} \) and \( x_{79,y}^{\text{out},f} \) are the curvature and moment within the network. Note that \( \sum x_{78,y}^f \) and \( \sum x_{79,y}^f \) are the summation of all the previous PE's. Other important observations were that equations (8.3) and (8.5) are similar to equations (6.95) and (6.97) obtained in chapter 6. However since the values for \( x_{78,y}^{\text{out},f} \) and \( x_{79,y}^{\text{out},f} \) are different, the values for \( \sum x_{78,y}^f \) and \( \sum x_{79,y}^f \) are different from equations (6.96) and (6.98).
Table 8.8. Comparison of Moments and Curvatures at Yield by the Preparatory and Final Stages. \( \% \text{Error} = \text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \)

<table>
<thead>
<tr>
<th>Specimen</th>
<th>( f'_c )</th>
<th>( \rho )</th>
<th>( \text{Preparatory Stage} )</th>
<th>( \text{Final Stage} )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specimen</td>
<td>( N / \text{mm}^2 )</td>
<td>(%)</td>
<td>Nirjar [1]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( M_y )</td>
<td>( kN - \text{mm} )</td>
<td>( M_y )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( \text{Compted} )</td>
<td>( \text{mm}^{-1} )</td>
<td>( kN - \text{mm} )</td>
</tr>
<tr>
<td>NN_1</td>
<td>30.0</td>
<td>1.28</td>
<td>2880</td>
<td>2615.82</td>
</tr>
<tr>
<td>NN_2</td>
<td>30.0</td>
<td>1.28</td>
<td>2940</td>
<td>2615.82</td>
</tr>
<tr>
<td>NN_3</td>
<td>30.0</td>
<td>1.28</td>
<td>2850</td>
<td>2615.82</td>
</tr>
<tr>
<td>NN_4</td>
<td>30.0</td>
<td>1.28</td>
<td>2790</td>
<td>2615.82</td>
</tr>
<tr>
<td>NN_5</td>
<td>30.0</td>
<td>1.28</td>
<td>2760</td>
<td>2615.82</td>
</tr>
</tbody>
</table>
Table 8.8. (continued)

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$NN_6$</td>
<td>30.0</td>
<td>1.28</td>
<td>2760</td>
<td>2615.82</td>
<td>5.22</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_7$</td>
<td>30.0</td>
<td>1.28</td>
<td>2880</td>
<td>2615.82</td>
<td>9.17</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_8$</td>
<td>30.0</td>
<td>1.28</td>
<td>2850</td>
<td>2615.82</td>
<td>8.22</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_9$</td>
<td>30.0</td>
<td>1.28</td>
<td>2820</td>
<td>2615.82</td>
<td>724</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_{10}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2820</td>
<td>2615.82</td>
<td>7.24</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{11}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2760</td>
<td>2615.82</td>
<td>5.22</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{12}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2760</td>
<td>2615.82</td>
<td>5.22</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{13}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2760</td>
<td>2615.82</td>
<td>5.22</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{14}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2910</td>
<td>2615.82</td>
<td>10.11</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NP_{15}$</td>
<td>30.0</td>
<td>0.72</td>
<td>1650</td>
<td>1632.30</td>
<td>1.07</td>
<td>1523.23</td>
<td>1734.55</td>
<td>1523.01</td>
<td>1732.01</td>
<td>0.01</td>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 8.8. (continued)

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 8.8. (continued)

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NR26</td>
<td>30.0</td>
<td>2.99</td>
<td>5760</td>
<td>6057.88</td>
<td>5.17</td>
<td>2412.72</td>
<td>6150.13</td>
<td>2412.71</td>
<td>6149.92</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>NR27</td>
<td>30.0</td>
<td>2.99</td>
<td>2700</td>
<td>2615.82</td>
<td>3.12</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
</tr>
<tr>
<td>NR28</td>
<td>30.0</td>
<td>2.99</td>
<td>2730</td>
<td>2615.82</td>
<td>4.18</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
</tr>
<tr>
<td>NR29</td>
<td>30.0</td>
<td>2.99</td>
<td>2700</td>
<td>2615.82</td>
<td>3.12</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
</tr>
<tr>
<td>NR30</td>
<td>30.0</td>
<td>2.99</td>
<td>2760</td>
<td>2615.82</td>
<td>5.22</td>
<td>1684.21</td>
<td>2992.84</td>
<td>1666.72</td>
<td>2804.84</td>
<td>1.04</td>
<td>6.28</td>
</tr>
<tr>
<td>NT31</td>
<td>40</td>
<td>1.28</td>
<td>2910</td>
<td>2641.02</td>
<td>9.24</td>
<td>1558.71</td>
<td>3062.02</td>
<td>1555.27</td>
<td>2876.28</td>
<td>0.22</td>
<td>6.07</td>
</tr>
<tr>
<td>NT32</td>
<td>35</td>
<td>1.28</td>
<td>2850</td>
<td>2623.67</td>
<td>7.94</td>
<td>1628.03</td>
<td>3022.49</td>
<td>1610.46</td>
<td>2828.19</td>
<td>1.08</td>
<td>6.43</td>
</tr>
<tr>
<td>NT33</td>
<td>25</td>
<td>1.28</td>
<td>2790</td>
<td>2607.58</td>
<td>6.54</td>
<td>1769.71</td>
<td>2951.32</td>
<td>1754.23</td>
<td>2766.97</td>
<td>0.87</td>
<td>6.25</td>
</tr>
<tr>
<td>NT34</td>
<td>20</td>
<td>1.28</td>
<td>2700</td>
<td>2623.14</td>
<td>2.85</td>
<td>1915.57</td>
<td>2889.06</td>
<td>1894.96</td>
<td>2718.76</td>
<td>1.08</td>
<td>5.89</td>
</tr>
</tbody>
</table>
Figure 8.2. Curvature and Moment Networks at Yield for the Final Stage.
8.2.2.2. Moment-Curvature Relationship at Ultimate

This is the moment-curvature relationship at ultimate for the final stage. A similar network was adopted as for the preparatory stage at ultimate but the training and test files conform only to the procedure adopted by Park and Ruitong [99] and the analysis provided by the ACI code [81].

Topology configuration, architecture representation and neurodynamics similar to that applied in the preparatory stage were applied at ultimate, to provide consistent comparisons. The train file contained 243 patterns and the test file 29. The network ceased learning at 24184 cycles with excellent confusion matrices of 1.0 for both the curvature and moment. A value of 1.0 was also obtained for the classification rate. Table 8.9 shows the results for the preparatory stage was as well as the final stage. The full networks for the trained and tested networks are shown in Figures 8.3(a) and (b).

The final stage equations provided by the NeuralWorks® [4] by means of the flashcode are:

(i) for the curvature at ultimate:

\[ y_{78,\mu}^{\text{out},f} = x_{78,\mu}^{\text{out},f} \times (20813.693) + 23738.135 \]  
\[ x_{78,\mu}^{\text{out},f} = \tanh(\sum x_{78,\mu}^f) \]  

(ii) for the moment at ultimate:

\[ y_{79,\mu}^{\text{out},f} = x_{79,\mu}^{\text{out},f} \times (3219.13) + 4261.97 \]  
\[ x_{79,\mu}^{\text{out},f} = \tanh(\sum x_{79,\mu}^f) \]  

where \( y_{78,\mu}^{\text{out},f} \) and \( y_{79,\mu}^{\text{out},f} \) are the curvature and moment respectively. Note that equations (8.7) and (8.9) for \( y_{78,\mu}^{\text{out},f} \) and \( y_{79,\mu}^{\text{out},f} \) respectively are similar to equations...
(6.107) and (6.109) for $y_{78,u}^{out,r}$ and $y_{79,u}^{out,r}$ in chapter 6. However, the scales values represented by $x_{78,u}^{out,f}$ and $x_{79,u}^{out,f}$ in the final stage are different from those obtained in chapter 6 as these depend on summations of the variables in $\sum x_{78,u}^{f}$ and $\sum x_{79,u}^{f}$.

The full equations at the final stage for both the moment and the curvature at ultimate are provided on disk at the end of thesis.
Table 8.9. Comparison of Moments and Curvatures at Ultimate for the Preparatory and Final Stages. \( \% Error = \text{Abs}(1 - \frac{\text{Network}}{\text{Desired}}) \)

<table>
<thead>
<tr>
<th>Specimens</th>
<th>( f'_c )</th>
<th>( \rho )</th>
<th>( f'_c )</th>
<th>( \rho )</th>
<th>( \text{Preparatory Stage} )</th>
<th>( \text{Final Stage} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( NN_1 )</td>
<td>30.0</td>
<td>1.28</td>
<td>3180</td>
<td>2780.98</td>
<td>12.55</td>
<td>15946.15 3008.74</td>
</tr>
<tr>
<td>( NN_2 )</td>
<td>30.0</td>
<td>1.28</td>
<td>3240</td>
<td>2780.98</td>
<td>14.17</td>
<td>15946.15 3008.74</td>
</tr>
<tr>
<td>( NN_3 )</td>
<td>30.0</td>
<td>1.28</td>
<td>3300</td>
<td>2780.98</td>
<td>15.73</td>
<td>15946.15 3008.74</td>
</tr>
<tr>
<td>( NN_4 )</td>
<td>30.0</td>
<td>1.28</td>
<td>3000</td>
<td>2780.98</td>
<td>7.30</td>
<td>15946.15 3008.74</td>
</tr>
<tr>
<td>( NN_5 )</td>
<td>30.0</td>
<td>1.28</td>
<td>2970</td>
<td>2780.98</td>
<td>6.36</td>
<td>15946.15 3008.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>----</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>$NN_{6}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2940</td>
<td>2780.98</td>
<td>5.41</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NM_{7}$</td>
<td>30.0</td>
<td>1.28</td>
<td>3060</td>
<td>2780.98</td>
<td>9.12</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NM_{8}$</td>
<td>30.0</td>
<td>1.28</td>
<td>3090</td>
<td>2780.98</td>
<td>10.00</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NM_{9}$</td>
<td>30.0</td>
<td>1.28</td>
<td>3210</td>
<td>2780.98</td>
<td>13.37</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NM_{10}$</td>
<td>30.0</td>
<td>1.28</td>
<td>3270</td>
<td>2780.98</td>
<td>14.95</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NO_{11}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2790</td>
<td>2780.98</td>
<td>0.32</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NO_{12}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2820</td>
<td>2780.98</td>
<td>1.38</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NO_{13}$</td>
<td>30.0</td>
<td>1.28</td>
<td>2910</td>
<td>2780.98</td>
<td>4.43</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NO_{14}$</td>
<td>30.0</td>
<td>1.28</td>
<td>3060</td>
<td>2780.98</td>
<td>9.12</td>
<td>15946.15</td>
</tr>
<tr>
<td>$NP_{15}$</td>
<td>30.0</td>
<td>0.72</td>
<td>2040</td>
<td>2780.98</td>
<td>7.79</td>
<td>28348.71</td>
</tr>
</tbody>
</table>
Table 8.9. (continued)

<p>| | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$NP_{16}$</td>
<td>30.0</td>
<td>2.00</td>
<td>4500</td>
<td>1881.03</td>
<td>16.03</td>
<td>10205.54</td>
<td>4532.71</td>
<td>9805.13</td>
<td>5028.92</td>
<td>3.92</td>
<td>10.95</td>
</tr>
<tr>
<td>$NP_{17}$</td>
<td>30.0</td>
<td>2.55</td>
<td>5275</td>
<td>5221.43</td>
<td>15.44</td>
<td>7973.08</td>
<td>5634.17</td>
<td>7782.63</td>
<td>5954.40</td>
<td>2.39</td>
<td>5.68</td>
</tr>
<tr>
<td>$NP_{18}$</td>
<td>30.0</td>
<td>2.99</td>
<td>5775</td>
<td>6089.44</td>
<td>9.65</td>
<td>6803.69</td>
<td>6448.13</td>
<td>6874.65</td>
<td>6304.38</td>
<td>1.04</td>
<td>2.23</td>
</tr>
<tr>
<td>$NQ_{19}$</td>
<td>30.0</td>
<td>0.72</td>
<td>2010</td>
<td>6332.57</td>
<td>6.42</td>
<td>28348.71</td>
<td>1739.58</td>
<td>28348.33</td>
<td>1742.31</td>
<td>0.0</td>
<td>0.16</td>
</tr>
<tr>
<td>$NQ_{20}$</td>
<td>30.0</td>
<td>2.00</td>
<td>4400</td>
<td>1881.03</td>
<td>18.67</td>
<td>10205.54</td>
<td>4532.71</td>
<td>9805.13</td>
<td>5028.92</td>
<td>3.92</td>
<td>10.95</td>
</tr>
<tr>
<td>$NQ_{21}$</td>
<td>30.0</td>
<td>2.55</td>
<td>5170</td>
<td>5221.43</td>
<td>17.67</td>
<td>7973.08</td>
<td>5634.17</td>
<td>7782.63</td>
<td>5954.40</td>
<td>2.39</td>
<td>5.68</td>
</tr>
<tr>
<td>$NQ_{22}$</td>
<td>30.0</td>
<td>2.99</td>
<td>5700</td>
<td>6089.44</td>
<td>11.10</td>
<td>6803.69</td>
<td>6448.13</td>
<td>6874.65</td>
<td>6304.38</td>
<td>1.04</td>
<td>2.23</td>
</tr>
<tr>
<td>$NR_{23}$</td>
<td>30.0</td>
<td>2.99</td>
<td>6300</td>
<td>6284.95</td>
<td>0.24</td>
<td>7087.18</td>
<td>6230.63</td>
<td>7085.38</td>
<td>6230.37</td>
<td>0.03</td>
<td>0.0</td>
</tr>
<tr>
<td>$NR_{24}$</td>
<td>30.0</td>
<td>2.99</td>
<td>6210</td>
<td>6284.95</td>
<td>1.21</td>
<td>7087.18</td>
<td>6230.63</td>
<td>7085.38</td>
<td>6230.37</td>
<td>0.03</td>
<td>0.0</td>
</tr>
<tr>
<td>$NR_{25}$</td>
<td>30.0</td>
<td>2.99</td>
<td>5775</td>
<td>6284.95</td>
<td>8.83</td>
<td>7087.18</td>
<td>6230.63</td>
<td>7085.38</td>
<td>6230.37</td>
<td>0.03</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>( NR_{26} )</td>
<td>30.0</td>
<td>2.99</td>
<td>5775</td>
<td>6284.95</td>
<td>8.83</td>
<td>7087.18</td>
<td>6230.63</td>
<td>7085.38</td>
<td>6230.37</td>
<td>0.03</td>
</tr>
<tr>
<td>----</td>
<td>---------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>---------</td>
<td>------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>------</td>
</tr>
<tr>
<td></td>
<td>( NT_{31} )</td>
<td>40</td>
<td>1.28</td>
<td>3090</td>
<td>2859.76</td>
<td>7.45</td>
<td>22718.87</td>
<td>3065.87</td>
<td>24138.30</td>
<td>2927.72</td>
<td>6.25</td>
</tr>
<tr>
<td></td>
<td>( NT_{32} )</td>
<td>35</td>
<td>1.28</td>
<td>3000</td>
<td>2867.57</td>
<td>4.41</td>
<td>18292.64</td>
<td>3033.32</td>
<td>18514.48</td>
<td>2890.64</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>( NT_{33} )</td>
<td>25</td>
<td>1.28</td>
<td>2880</td>
<td>16956.43</td>
<td>18.80</td>
<td>13510.75</td>
<td>2974.19</td>
<td>13800.13</td>
<td>2803.13</td>
<td>2.14</td>
</tr>
<tr>
<td></td>
<td>( NT_{34} )</td>
<td>20</td>
<td>1.28</td>
<td>2730</td>
<td>15894.50</td>
<td>4.24</td>
<td>10986.43</td>
<td>2922.22</td>
<td>11266.89</td>
<td>2.55</td>
<td>5.01</td>
</tr>
</tbody>
</table>
Figure 8.3. Curvature and Moment Networks at Ultimate for the Final Stage.
8.2.3. Shear Forces at the Final Stage

This stage consisted of applying the ACI-ASCE Committee 352 recommendations [3] for shear force to the analysis based on the ACI Code of 1989 [81].

The architecture, topology and neurodynamics used in section 7.8 was maintained for the preparatory stage. The training data contained 7739 patterns and the test data 965 patterns. The networks ceased learning at 591904 cycles with excellent confusion matrices and classification rates of 1.0 for both the trained and tested networks. The reliability of the network was checked by passing the data obtained for the 34 specimens not seen by the network. The results tabulated in Table 8.10 are for the preparatory stage, as obtained in chapter 7, and the final stage, together with values of the percentage errors.

The procedure once more illustrates the full potential of predicting data usually obtained from an experimental study and provides an alternative way for obtaining data which is usually obtained by testing.

The shear force obtained by the neural network at the final stage is expressed in the form:

\[
y_{95}^{out,f} = x_{95}^{out,f} \times (65276.40) + (-1131.129)
\]

\[
x_{95}^{out,f} = \frac{1.0}{1.0 + e^{-\Sigma x_{95}}}
\]

The above equations are similar to equations (7.78) and (7.79) given in chapter 7. However the internal representations are again different as shown previously.

The final networks obtained at the final stage for the trained and tested networks are shown in Figure 8.4.
Table 8.10. Shear Force Strength for the Model Specimens at the Final Stage as Predicted by Neural Networks.

\[
\%Error = \text{Abs}(1 - \frac{\text{Network}}{\text{Desired}})
\]

<table>
<thead>
<tr>
<th>Specimens</th>
<th>( f_c )</th>
<th>( \rho )</th>
<th>Preparatory Stage</th>
<th>Final Stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>( NN_1 )</td>
<td>30</td>
<td>1.28</td>
<td>Nirjar [1] ( V_u, \text{Newtons} )</td>
<td>Neural ( V_u, \text{Newtons} )</td>
</tr>
<tr>
<td>( NP_{15} )</td>
<td>30</td>
<td>0.72</td>
<td>21613.96</td>
<td>20357.93</td>
</tr>
<tr>
<td>( NP_{16} )</td>
<td>30</td>
<td>2.00</td>
<td>12026.15</td>
<td>11993.21</td>
</tr>
<tr>
<td>( NP_{17} )</td>
<td>30</td>
<td>2.55</td>
<td>34242.18</td>
<td>39858.11</td>
</tr>
<tr>
<td>( NP_{18} )</td>
<td>30</td>
<td>2.99</td>
<td>44298.28</td>
<td>48082.03</td>
</tr>
<tr>
<td>( NQ_{20} )</td>
<td>30</td>
<td>2.00</td>
<td>52343.23</td>
<td>50798.46</td>
</tr>
</tbody>
</table>

337
Table 8.10. (Continued)

<table>
<thead>
<tr>
<th></th>
<th>$NQ_{21}$</th>
<th>$NQ_{22}$</th>
<th>$NR_{23}$</th>
<th>$NT_{31}$</th>
<th>$NT_{32}$</th>
<th>$NT_{33}$</th>
<th>$NT_{34}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30</td>
<td>2.55</td>
<td>30</td>
<td>2.99</td>
<td>30</td>
<td>2.99</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>44298.28</td>
<td>52343.23</td>
<td>50136.61</td>
<td>21291.59</td>
<td>21472.77</td>
<td>21819.08</td>
<td>22142.21</td>
</tr>
<tr>
<td></td>
<td>48082.03</td>
<td>50798.46</td>
<td>50191.02</td>
<td>20015.09</td>
<td>20200.48</td>
<td>20605.74</td>
<td>21051.30</td>
</tr>
<tr>
<td></td>
<td>8.54</td>
<td>2.95</td>
<td>0.11</td>
<td>6.00</td>
<td>5.93</td>
<td>5.56</td>
<td>4.93</td>
</tr>
<tr>
<td></td>
<td>45051.57</td>
<td>53377.71</td>
<td>51089.99</td>
<td>21561.11</td>
<td>21698.92</td>
<td>21946.98</td>
<td>22164.03</td>
</tr>
<tr>
<td></td>
<td>48893.63</td>
<td>51753.70</td>
<td>51113.43</td>
<td>20376.98</td>
<td>20388.86</td>
<td>20644.05</td>
<td>20999.42</td>
</tr>
<tr>
<td></td>
<td>8.53</td>
<td>3.04</td>
<td>0.05</td>
<td>5.49</td>
<td>6.04</td>
<td>5.94</td>
<td>5.25</td>
</tr>
</tbody>
</table>
Figure 8.4. Trained and Tested Networks for Shear Force at Final Stage.
8.3. ADJUSTMENTS FOR THE SHEAR STRENGTH OF THE MODEL

The preparatory stage for model specimen evaluation was discussed in chapter 7 (section 7.7) where the results for ultimate shear stress, $v_u$, the shear stress carried by the concrete, $v_c$, and the shear stress carried by the stirrups, $v_s$, were tabulated together with the percentage errors (Table 7.5). The percentage error obtained for the shear stress carried by the stirrups, $v_s$, are large compared to the other results. In an attempt to improve the performance of the network by recognising a better relationship, an intermediate spacing of 60 mm was introduced between the values of 40 and 80 mm used in the preparatory stage. The output of the numerical analysis by FORTRAN produced 11552 training patterns of input data and a test file of 1442 compared to values of 7739 and 965 respectively for the preparatory stage. Similar networks were used with the same learning algorithm and momentum. The network trained by presenting the data randomly for the training set converged after 188747 cycles with excellent confusion matrices for the trained network of 0.9998, 0.9987 and 0.9997 for $v_u$, $v_c$ and $v_s$ respectively and a classification rate of 1.0 for all. A test file of 1442 was then presented to the trained network sequentially to improve the network performance and to adjust to new data not seen before. The reliability of the network data was checked by introducing data for the 34 specimens which contained all the parameters used in the experimental study. The results obtained for $v_s$ were excellent compared to those from the preparatory results. The values of $v_u$ and $v_c$ were also improved. The full results are shown in Table 8.11 together with the percentage errors. The shaded area indicates all those under investigation.

This procedure demonstrates the reliability of neural networks to predict results usually obtained by testing. It is a procedure which can enhance the data by selecting additional variables within the specific domain. The amount of data obtained is considerable compared to the data obtained by testing, confirming the benefits of numerical computation with neural networks.
The modified formulae for the shear strength obtained after adjusting the spacing of the links are:

(i) for ultimate shear stress, $v_u$:

\[ y_{97}^{\text{out},f} = x_{97}^{\text{out},f} \times (8.733) + (-0.117) \]  
\[ x_{97}^{\text{out},f} = \frac{1.0}{1.0 + e^{-\Sigma e_{97}}} \]

(ii) for the shear stress carried by the concrete, $v_c$:

\[ y_{98}^{\text{out},f} = x_{98}^{\text{out},f} \times (16.40) + (0.10) \]
\[ x_{98}^{\text{out},f} = \frac{1.0}{1.0 + e^{-\Sigma e_{98}}} \]

(iii) for the shear stress carried by the stirrups, $v_s$:

\[ y_{99}^{\text{out},f} = x_{99}^{\text{out},f} \times (2.883) + (-0.147) \]
\[ x_{99}^{\text{out},f} = \frac{1.0}{1.0 + e^{-\Sigma e_{99}}} \]

Note that equations (8.13), (8.15) and (8.17) for $y_{97}^{\text{out},f}$, $y_{98}^{\text{out},f}$ and $y_{99}^{\text{out},f}$ respectively are similar to equations (7.70), (7.72) and (7.74) for $y_{97}^{\text{out},p}$, $y_{98}^{\text{out},p}$ and $y_{99}^{\text{out},p}$ obtained in chapter 7. However, the scale values represented by $x_{97}^{\text{out},f}$, $x_{98}^{\text{out},f}$ and $x_{99}^{\text{out},f}$ for the final stage are different from those obtained in chapter 7 as these depend on the summation of variables in $e^{-\Sigma e_{r}}$, $e^{-\Sigma e_{c}}$ and $e^{-\Sigma e_{s}}$ which are different for each case, due to weight adjustments between the PE's at the final stage of the network. The full equations for $v_u$, $v_c$ and $v_s$ respectively are provided on a disk at the end of the thesis.

The adjusted network architectures for both the training and test networks are shown in Figure 8.5 (a) and 8.5 (b) respectively.
<table>
<thead>
<tr>
<th>Specimens</th>
<th>$P_u$</th>
<th>$P_a$</th>
<th>Preparatory Stage</th>
<th>Final Stage</th>
<th>% Error $= \left(1 - \frac{\text{Network}}{\text{Desired}}\right)$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$kN$</td>
<td>$kN$</td>
<td>$v_u$ $N/mm^2$</td>
<td>$v_c$ $N/mm^2$</td>
<td>$v_s$ $N/mm^2$</td>
</tr>
<tr>
<td>$NN_1$</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
</tr>
<tr>
<td>$NN_2$</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>5.47</td>
<td>1.44</td>
</tr>
<tr>
<td>$NN_3$</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>6.30</td>
<td>1.44</td>
</tr>
<tr>
<td>$NN_4$</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>7.03</td>
<td>1.44</td>
</tr>
</tbody>
</table>
Table 8.11. (continued)

<p>| | | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$NN_5$</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>7.70</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>2.74</td>
<td>7.23</td>
<td>1.44</td>
<td>7.43</td>
<td>6.10</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NN_6$</td>
<td>294.37</td>
<td>29.44</td>
<td>2.96</td>
<td>8.31</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>2.74</td>
<td>8.13</td>
<td>1.44</td>
<td>7.43</td>
<td>2.15</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_7$</td>
<td>380.99</td>
<td>38.10</td>
<td>2.96</td>
<td>4.80</td>
<td>1.44</td>
<td>1.77</td>
<td>25.00</td>
<td>2.74</td>
<td>4.45</td>
<td>1.44</td>
<td>7.09</td>
<td>7.25</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_8$</td>
<td>364.37</td>
<td>34.25</td>
<td>2.96</td>
<td>4.74</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>2.74</td>
<td>4.44</td>
<td>1.44</td>
<td>7.09</td>
<td>6.39</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_9$</td>
<td>342.49</td>
<td>34.25</td>
<td>2.96</td>
<td>4.66</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>2.74</td>
<td>4.42</td>
<td>1.44</td>
<td>7.43</td>
<td>5.19</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NM_{10}$</td>
<td>311.00</td>
<td>31.10</td>
<td>2.96</td>
<td>4.55</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>2.74</td>
<td>4.39</td>
<td>1.44</td>
<td>7.43</td>
<td>3.46</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{11}$</td>
<td>380.99</td>
<td>190.50</td>
<td>2.96</td>
<td>8.58</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>2.74</td>
<td>8.38</td>
<td>1.45</td>
<td>7.43</td>
<td>2.38</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{12}$</td>
<td>364.37</td>
<td>182.18</td>
<td>2.96</td>
<td>8.42</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>2.74</td>
<td>8.15</td>
<td>1.45</td>
<td>7.43</td>
<td>3.18</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{13}$</td>
<td>342.49</td>
<td>171.25</td>
<td>2.96</td>
<td>8.20</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>2.74</td>
<td>7.86</td>
<td>1.45</td>
<td>7.43</td>
<td>4.14</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$NO_{14}$</td>
<td>311.00</td>
<td>155.50</td>
<td>2.96</td>
<td>7.87</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>2.74</td>
<td>7.45</td>
<td>1.45</td>
<td>7.43</td>
<td>5.40</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 8.11. (continued)

<table>
<thead>
<tr>
<th>$NP_{15}$</th>
<th>294.37</th>
<th>29.44</th>
<th>1.65</th>
<th>4.49</th>
<th>1.44</th>
<th>1.75</th>
<th>21.53</th>
<th>1.62</th>
<th>4.44</th>
<th>1.43</th>
<th>1.82</th>
<th>1.20</th>
<th>0.69</th>
</tr>
</thead>
<tbody>
<tr>
<td>$NP_{16}$</td>
<td>294.37</td>
<td>29.44</td>
<td>4.69</td>
<td>4.49</td>
<td>1.44</td>
<td>1.75</td>
<td>21.53</td>
<td>5.43</td>
<td>4.43</td>
<td>1.46</td>
<td>15.78</td>
<td>1.37</td>
<td>1.39</td>
</tr>
<tr>
<td>$NP_{17}$</td>
<td>294.37</td>
<td>29.44</td>
<td>6.07</td>
<td>4.49</td>
<td>1.44</td>
<td>1.74</td>
<td>20.83</td>
<td>6.54</td>
<td>4.53</td>
<td>1.45</td>
<td>7.74</td>
<td>0.95</td>
<td>0.69</td>
</tr>
<tr>
<td>$NP_{18}$</td>
<td>294.37</td>
<td>29.44</td>
<td>7.17</td>
<td>4.49</td>
<td>1.44</td>
<td>1.73</td>
<td>20.14</td>
<td>6.89</td>
<td>4.61</td>
<td>1.44</td>
<td>3.91</td>
<td>2.63</td>
<td>0.0</td>
</tr>
<tr>
<td>$NO_{19}$</td>
<td>294.37</td>
<td>176.62</td>
<td>1.65</td>
<td>8.31</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>1.62</td>
<td>8.08</td>
<td>1.45</td>
<td>1.82</td>
<td>2.71</td>
<td>0.69</td>
</tr>
<tr>
<td>$NO_{20}$</td>
<td>294.37</td>
<td>176.62</td>
<td>4.69</td>
<td>8.31</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>5.48</td>
<td>8.10</td>
<td>1.47</td>
<td>16.84</td>
<td>2.50</td>
<td>2.08</td>
</tr>
<tr>
<td>$NO_{21}$</td>
<td>294.37</td>
<td>176.62</td>
<td>6.07</td>
<td>8.31</td>
<td>1.44</td>
<td>1.78</td>
<td>23.61</td>
<td>6.62</td>
<td>8.03</td>
<td>1.46</td>
<td>9.06</td>
<td>3.32</td>
<td>1.39</td>
</tr>
<tr>
<td>$NO_{22}$</td>
<td>294.37</td>
<td>176.62</td>
<td>7.17</td>
<td>8.31</td>
<td>1.44</td>
<td>1.78</td>
<td>23.61</td>
<td>6.96</td>
<td>7.99</td>
<td>1.44</td>
<td>2.93</td>
<td>3.81</td>
<td>0.0</td>
</tr>
<tr>
<td>$NR_{23}$</td>
<td>342.49</td>
<td>34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>1.08</td>
<td>1.08</td>
<td>0.00</td>
<td>6.79</td>
<td>4.77</td>
<td>1.09</td>
<td>1.16</td>
<td>2.42</td>
<td>0.93</td>
</tr>
<tr>
<td>$NR_{24}$</td>
<td>342.49</td>
<td>34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>0.86</td>
<td>0.85</td>
<td>1.16</td>
<td>6.84</td>
<td>4.62</td>
<td>0.88</td>
<td>0.44</td>
<td>0.76</td>
<td>2.33</td>
</tr>
</tbody>
</table>
Table 8.11 (continued)

<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>343.49 &amp; 34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>0.43</td>
<td>0.44</td>
<td>2.33</td>
<td>6.76</td>
<td>4.74</td>
<td>0.45</td>
<td>1.60</td>
<td>1.79</td>
<td>4.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NR_{25}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>342.49 &amp; 34.25</td>
<td>6.87</td>
<td>4.66</td>
<td>0.43</td>
<td>0.44</td>
<td>2.33</td>
<td>6.76</td>
<td>4.74</td>
<td>0.45</td>
<td>1.60</td>
<td>1.79</td>
<td>4.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>294.37 &amp; 29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>2.74</td>
<td>4.48</td>
<td>1.44</td>
<td>7.43</td>
<td>2.46</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS_{27}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>294.37 &amp; 29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>2.74</td>
<td>4.38</td>
<td>1.44</td>
<td>7.43</td>
<td>2.46</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>294.37 &amp; 29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>2.74</td>
<td>4.38</td>
<td>1.44</td>
<td>7.43</td>
<td>2.46</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>294.37 &amp; 29.44</td>
<td>2.96</td>
<td>4.49</td>
<td>1.44</td>
<td>1.77</td>
<td>22.92</td>
<td>2.74</td>
<td>4.38</td>
<td>1.44</td>
<td>7.43</td>
<td>2.46</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>420.07 &amp; 210.03</td>
<td>2.92</td>
<td>10.96</td>
<td>1.44</td>
<td>1.79</td>
<td>24.31</td>
<td>2.73</td>
<td>11.09</td>
<td>1.45</td>
<td>6.51</td>
<td>0.85</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NT_{31}</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>336.27 &amp; 168.14</td>
<td>2.94</td>
<td>8.79</td>
<td>1.44</td>
<td>1.80</td>
<td>25.00</td>
<td>2.72</td>
<td>8.55</td>
<td>1.45</td>
<td>7.48</td>
<td>2.75</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>252.47 &amp; 126.24</td>
<td>2.99</td>
<td>6.60</td>
<td>1.44</td>
<td>1.79</td>
<td>24.31</td>
<td>2.79</td>
<td>6.05</td>
<td>1.45</td>
<td>6.69</td>
<td>8.28</td>
<td>0.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>210.57 &amp; 105.29</td>
<td>3.03</td>
<td>5.50</td>
<td>1.44</td>
<td>1.78</td>
<td>23.61</td>
<td>2.89</td>
<td>5.08</td>
<td>1.44</td>
<td>4.62</td>
<td>7.62</td>
<td>0.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 8.5. Adjustment Networks for the Model Specimens.
CHAPTER 9

SUMMARY AND CONCLUSION

9.1. GENERAL SUMMARY

The fundamental aspects of symbolic based learning was stressed in chapter 2 which explained the role of Artificial Intelligence in general and expert systems in particular with the emphasis on machine learning. The theoretical mathematical foundation of parallel distributed processing was explained in chapter 3 and the roles of neural networks were examined, especially the backpropagation algorithm. The emergence of neurocomputations software and hardware was discussed in chapter 4 and detailed information was given on the implementation of NeuralWorks® Professional II/Plus version 5 [17].

In chapter 5, a theoretical analysis of the ultimate flexural strength of reinforced concrete members was presented to demonstrate the benefits of neural network in the evaluation of several parameters. The results clearly indicated that the neural network application represents a method of evaluating the ultimate moment of a cross-section which is comparable with established methods using experimental and analytical procedures. Experimental investigations however demand extensive data recording over an extended period of time.

The application of neural networks to predict the moment-curvature parameters from analytical and experimental data was discussed in chapter 6. The author demonstrated the concept of backpropagation in artificial neural networks to assist
experimental testing. The results demonstrated that neural network methodology offers potential and valuable alternatives to obtaining curvatures and moments from experimental data. Instead of carrying out extensive experimental tests, neural networks offer an alternative procedure which requires less experimental specimen, concrete strengths, specific bar arrangements and maximum and minimum bar sizes. By constructing a model that allows for different experimental testing combinations with numerical analysis potentiality, neural network paradigms provide an advance in the prediction of parameters that have previously required repetition of the experimental work.

Chapter 7 investigates the general feasibility of shear behaviour and the strength characteristics of beam-column joints by using both existing and recently published data. Having established an appropriate shear computational procedure, the work was extended to neural network implementation to predict several variables that are essential in the assessment of beam-column joints.

In chapter 8, the final stage is proposed to carry out a procedure similar to that adopted previously, namely using the updated code of practice plus any new relevant information to perform preliminary experimental work as a secondary objective. This provides information on materials, loading and geometrical relationships before detailed experimental work is carried out.

The author therefore strongly advocates the use of computer technology to increase progress in research and obtain more rapid and reliable solutions to problems that confront the engineer. Its use must be balanced by an understanding of the behaviour of the model so that its performance can be simulated by artificial modelling. The research conducted in this study has involved the integration of software from word-processors, spreadsheets, graphical interfaces, programming languages, mathematical tools, operating systems and neural network simulators to encourage the use of computer technology in research and obtain results previously
unobtainable. However, analysis by computer depends on an accurate understanding of mathematical modelling and material behaviour idealisation.

The most accurate information is produced by experimental testing which reproduces the actual structural behaviour, monitored through instrumentation. The changes in structural behaviour are observed as the experiment progresses. However, with increases in the cost of experimental testing, neural networks can be implemented to assist with or speed up the testing procedures by providing an alternative method of generating the required data which will consequently reduce the overall cost.

The scope of the author's research was limited to the implementation of neural networks to predict specific variables and determine appropriate equations to perform the necessary computation. The study concentrated on concept and methodology rather than on full implantation of an expert system followed by procedure and implementation. The main concept was to view the experimental set-up and limit the amount of work by reducing the following: number of specimen, choice of concrete strengths, types of reinforcement in the beam, column and joint, types of lateral reinforcement and the column loading.

The concept can be applied to any engineering field or area which requires extensive experimental work and produces data based on analytical or numerical methods. The future as foreseen by the author is discussed, general conclusions are drawn and finally recommendations for future research are outlined.

9.2. FUTURE FORECAST

The computer industry is changing so rapidly that researchers can be confused by the methods and algorithms that have developed over recent years. The move from AI technology to neural network to parallelism and future promises for data visualisation and virtual reality have increased the responsibilities on the scientists,
researchers and the engineers. The field of computers is characterised by the rapid movement from one technology to another without real commitment or the fertilisation of technology to obtain the best results. Over the past forty years, tremendous changes in computer technology have occurred resulting in a provision of large amounts of data and resources. Figure 9.1 presents the author's view of past, current and future research trends in computational technology activity. The shaded areas compress the study conducted in the author's current research and important computational paradigms are discussed below.

9.2.1. Hybrid Knowledge Neural Expert System

The integration of a knowledge-based system and neural networks has provided an incredibly powerful tool that manipulates the potential of symbolic reasoning with aggressive pattern recognition, to reach an ultimate system that contains the best of both systems. In fact the knowledge-based system and the neural network compliment each other by a mechanism that can be bridged, because of their diverse nature. The implantation together of these two systems, can be of great benefit to the engineer due to decision making that is characterised by symbolic based and neural network that can manipulate data. The system in reality requires team work to provide the best solutions for a specific case.

9.2.2. Data Visualisation

Nowadays computers can analyse and create complex graphical systems that allow one to examine and manipulate complex non-linearity relationships in ways that never existed previously. The development of power and flexible visualisation software will allow researches to look beyond the tradition of data graphical packages and spreadsheets. The potential to identify and analyse underlying patterns in data provides a technique which can display abstract numerical data, neural network patterns data or statistics in comprehensive graphical form. It is a technique which not only generates quantities of data but also qualitative information that other
techniques have failed to achieve, such as mathematical software or spreadsheet packages; thus, offering a colour picture of internal relationships that makes information which is complex to perceive numerically more easily to visualise. Data visualisation techniques will provide new possibilities for analysing data provided by experimental instrumentations or through the manipulation of numerical analysis. Further development of the PC, both in hardware and software, will make it possible to implement visualisation with the PC whereas access was only possible before through the workstation or Mainframe. Several elemental components contributed to the development of the data visualisation technique, among them 3-dimensional graphics which provide an illusion of 3-D objects and the simulation techniques.

The use of the visualisation technique will be challenging since it addresses the most meaningful way of representing the data by using colour, animations, texture and sound, and is represented by the data into multi-dimensional space of more than 100 variables. The growing awareness of the data complexity information will encourage researchers to represent the data in 3-dimensions to show their full implication and usefulness.

The application of data visualisation will encompass areas where it can be used to visualise and analyse algorithms and programming techniques by using animations which guide the researchers in understanding the most complex algorithms. In the field of civil engineering it will guide the engineer in visualising the data from perspectives that have not been seen before.

Visualisation will provide the engineer and science researchers with an invaluable tool to probe into the relationship of complex mathematical non-linearity while the data processes its computations and also provides instant effects for changing parameters, resolutions or colour representations. With high technological computers, investigations of experimental testing simulations can dynamically alter and modify parameters that can monitor the progress of experimental simulation and
alleviate or terminate improper parameters. From the PC, the engineer will view and
design models and select material properties and invoke crash structural experimental
simulation testing without leaving his desk and observe the structural behaviour by
monitoring the stress and strain levels on sophisticated graphical presentations and
perform modifications instantly. From a series of test processes, the engineer will be
able to observe results on contoured superimposed graphics of the deformed shape
and distil huge amounts of numeric data into a series of snapshots or images that can
be executed over time in an animation process.

9.2.3. Virtual Reality: Is it Reality?

Computer technology dictates change faster than people can cope with it,
providing a more of data than we can cope with. Virtual reality is a computer
programming screen that gives you a feeling of depth by placing a helmet around the
head or small visor. With the introduction of virtual reality the engineer will be able
to design, redesign, construct landscape, furniture, light, live in the house without
leaving his design desk. Virtual reality will provide the ultimate in-depth 3-
dimension that is closest to reality. The implementation of a virtual reality system
requires a high performance specific computer and a headset surrounded by a cabinet.
With the introduction of the RISC processor and improvements in software, the
implementation of virtual reality will continue to be introduced in many areas.
Research in civil engineering will benefit from these technological achievements by
building full virtual structural models and performing testing and manoeuvres within
these models to observe the behaviour of the structure.
9.3. CONCLUSION

Neural networks provide new platforms for approaching engineering problems by exploring the benefits of this technique in reducing the amount of experimental work and increasing the number of specimens hypothetically tested using numerical manipulations. Previous studies indicate that the neural network is a new computational technology that can assist the testing procedure. It requires a delicate model set-up which takes into account the test procedure, specific criteria and
indicates boundary conditions by identifying the domain which is vital in neural network techniques.

Data obtained by neural networks were compared with experimental results, validated by theoretical formulations, and found to be satisfactory. Three stages were investigated to determine the beam-column joint parameters to illustrate the benefits and advantages of using neural networks together with manipulation by numerical analyses. The preliminary requirement of a neural network is the selection of a feasible and appropriate domain for the training and test data. These are essential for any further neural network manipulation. The selection of the neural network model requires a delicate procedure using a trial and error process to optimise the network by monitoring the output results from the instruments provided by the tool as the network evolves. The general conclusions drawn from implementing neural networks to analyse a reinforced concrete structure are summarised below:

1. The benefit of using a neural network analog method with three or more layers to be trained, was to provide the opportunity to conduct a structural analysis incorporating an alternative method with a solid mathematical foundation.

2. Neural networks capability lies in training sets of examples without programming by repeatedly showing the patterns to the network and predicting the relationship between variables by itself.

3. The relationship between variables is learned rather than by instructing the relationship in advance.

4. Backpropagation is an algorithm for training neural networks by relying on a gradient descent method to minimise the network error.

5. Overtraining the network can result in memorisation of the patterns which is undesirable because a network memorises a training pattern which typically does a poor job of generalisation.
6. The results indicate that the neural network approach is a step forward in predicting the missing data that relies partially on data that can be trained into the network.

7. A neural network does not depend on the formulation of a mathematical algorithm but bases its entire process on a set of examples presented to the network.

8. The use of neural network models to predict parameters for certain tasks incorporates knowledge within its domain in a form of example.

9. The automatic learning capability monitors the trends of patterns not seen before.

10. A major limitation of neural networks is the lack of intelligent "rule of thumb" which in certain situations are essential.

11. The quality of training and test sets are important. Better learning results are achieved by using more training sets but at the expense of training time.

9.4. RECOMMENDATION FOR FUTURE RESEARCH

The n-dimensional variables that neural networks use to manipulate data are still in fact primitive for presentation in graphical format. The promise of visualisation will instantly view multi-dimensional variables that provide an in-depth feeling, a sense of experimental work and will enhance numerical computation.

Aerospace Engineering which is young compared to Civil Engineering has experienced exponential advances in its research techniques and become more productive and reliable, mainly due to mass production, large amounts of investment as well as the use of aluminium materials for the structural shape. Finite element analysis has played a major part in the research of aircraft structures mainly due to the implementation of excellent hardware and software. One of the dilemmas that face civil engineering researchers in general, and structural analysis in particular, is
the variation in structural shapes and material properties which make progress
difficult and research time-consuming. The structural engineer can benefit from the
technology available to aerospace engineers by implementing 'instant' finite analysis.
This can be accomplished by carrying out a full finite element analysis and attaching
a graphical interface through the virtual reality technique to observe an instant 3-
dimensional structural behaviour and generating data that can be passed to neural
networks. The data can then be analysed with the aid of data visualisation to
represent the stress behaviour in the structure and observe it in several colourful
formats.

The technology is here and the future promises tremendous advances in both
hardware and software technology. What was impossible 10 years ago with
computers with only 640 kilobytes of memory, can be dealt with computer nowadays
with \textbf{RAM} in gigabytes and hard drives in tera, peta or exabytes!!!
REFERENCES


Knowledge Acquisition for Use in Learning Expert Systems", *Int. Man-Machine

57. McCulloch, W. C., Pitts, W., (1943), "A logical Calculus of the Ideas Immanent


Storage and Orginization in the Brain", *Psychological Review*, Vol. 65, pp. 386-
408.


Collective Computational Abilities", *Proceedings of the National Academy

Representations by Error Propagation", In Rumelhart, D. E., Hinton, G. E.,
Williams, R. J., (editors), *Parallel Distributed Processing: Explorations in the

WESCON Convention Record*, Part 4, pp. 96-104.


65. Ross, T., (1993), "Feature: The DTI Neural Computing Technology Transfer


117. American Concrete Institute, (1963), "Building Code Requirements for Reinforced Concrete", *ACI Code 318-63*, Detroit, USA.

118. American Concrete Institute, (1977), "Building Code Requirements for Reinforced Concrete", *ACI Code 318-77*, Detroit, USA.

119. American Concrete Institute, (1983), "Building Code Requirements for Reinforced Concrete", *ACI Code 318-83*, Detroit, USA.


APPENDIX A

Neural Networks

Manual Computation
NEURAL NETWORKS MANUAL COMPUTATION

This appendix provides simple manual neural network computation using backpropagation algorithm and omitting the first derivative to simplify calculations.

**Given:** Sample training example consists of:

<table>
<thead>
<tr>
<th>$x_{out}(2)$</th>
<th>$x_{out}(3)$</th>
<th>$d_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Network architecture is shown in Figure A.1

A sigmoidal activation function is assumed with backpropagation learning algorithm. The computation does not include a *Bias*.

The first training example consists of an input of 1.0 for $x_{out}(2)$ and 0.0 for $x_{out}(3)$ with desired output of $d_i = 1.0$.

**Procedure:**

1. Randomise the weights, $w_{pE}$, with arbitrary values, starting from the output layer back to input layer as:

   $w_{42} = -0.1456$  \quad $w_{43} = -0.4134$  \quad $w_{74} = -0.5565$

   $w_{52} = -0.2794$  \quad $w_{53} = -0.6520$  \quad $w_{75} = +0.5415$

   $w_{62} = +0.4604$  \quad $w_{63} = -0.1221$  \quad $w_{76} = -0.1679$
2. Compute the PE output starting from the input layer, such that the input to the PE of the input layer is known, i.e. $x_{out}(2)$ and $x_{out}(3)$.

For PE number 4:

\[ x_{sum}(4) = (x_{out}(2)) \cdot w_{42} + (x_{out}(3)) \cdot w_{43} \]
\[ = 1 \cdot (-0.1456) + 0 \cdot (-0.4134) \]
\[ x_{sum}(4) = -0.1456 \]

Activation function:

\[ x_{out}(4) = \frac{1.0}{1 + e^{-(x_{sum}(4))}} = \frac{1.0}{1 + e^{-0.1456}} \]

Therefore the output at PE 4 is:

\[ x_{out}(4) = 0.4637 \]

For PE number 5:

\[ x_{sum}(5) = (x_{out}(2)) \cdot w_{52} + (x_{out}(3)) \cdot w_{53} \]
\[ = 1 \cdot (-0.2794) + 0 \cdot (-0.6520) \]
\[ x_{sum}(5) = -0.2794 \]

Activation function:

\[ x_{out}(5) = \frac{1.0}{1 + e^{-(x_{sum}(5))}} = \frac{1.0}{1 + e^{-0.2794}} \]

Therefore the output at PE 5 is:

\[ x_{out}(5) = 0.4306 \]

For PE number 6:

\[ x_{sum}(6) = (x_{out}(2)) \cdot w_{62} + (x_{out}(3)) \cdot w_{63} \]
\[ = 1 \cdot (0.4604) + 0 \cdot (-0.1221) \]
\[ x_{sum}(6) = 0.4604 \]
Activation function:
\[ x_{out}(6) = \frac{1.0}{1 + e^{-(x_{out}(6))}} = \frac{1.0}{1 + e^{-0.4604}} \]

Therefore the output at PE 6 is:
\[ x_{out}(6) = 0.6131 \]

For PE number 7:
\[ x_{sum}(7) = (x_{out}(4)) * w_{74} + (x_{out}(5)) * w_{75} + (x_{out}(6)) * w_{76} \]
\[ = (0.4637)*(-0.5565) + (0.4306)*(0.5415) + (0.6131)*(-0.1679) \]
\[ x_{sum}(7) = -0.1278 \]

Activation function:
\[ x_{out}(7) = \frac{1.0}{1 + e^{-(x_{out}(7))}} = \frac{1.0}{1 + e^{0.1278}} \]

Therefore the output at PE 6 is:
\[ x_{out}(7) = 0.4681 \]

3. Compute the error \( E_{PE} \) for the network:
\[ E_7 = \text{Desired output at PE 7 - Actual Network output at PE 7} \]
\[ E_7 = d_7 - y_{out}(7) \]
\[ E_7 = 1.0 - 0.4681 = 0.5319 \]

4. The errors at previous layer, i.e. hidden layer for \( E_4 \), \( E_5 \) and \( E_6 \) are computed as:
\[ E_4 = E_7 * w_{74} \]
\[ = 0.5319*(-0.5565) \]
\[ E_4 = -0.2960 \]
\[ E_5 = E_7 * w_{75} \]
\[ = 0.5319*(0.5415) \]
\[ E_5 = 0.2880 \]

\[ E_6 = E_7 \ast w_{76} = 0.5319 \ast (-0.1679) \]

\[ E_6 = -0.0893 \]

5. Now compute the 'new' values for weight, \( w_{PE}^{new} \), by propagating the errors backward from the output layer through the hidden layer to the input layer.

From PE 7 to PE 4

\[ w_{74}^{new} = (E_7 \ast x_{out}(7)) + w_{74} = (0.5319 \ast 0.4681) + (-0.5565) \]

\[ w_{74}^{new} = -0.3075 \]

From PE 7 to PE 5

\[ w_{75}^{new} = (E_7 \ast x_{out}(7)) + w_{75} = (0.5319 \ast 0.4681) + (0.5415) \]

\[ w_{75}^{new} = 0.7905 \]

From PE 7 to PE 6

\[ w_{76}^{new} = (E_7 \ast x_{out}(7)) + w_{76} = (0.5319 \ast 0.4681) + (-0.1679) \]

\[ w_{76}^{new} = 0.0811 \]

From PE 4 to PE 2

\[ w_{42}^{new} = (E_4 \ast x_{out}(4)) + w_{42} = (-0.2960 \ast 0.4637) + (-0.1456) \]
\[ w_{42}^{\text{new}} = -0.2829 \]

From PE 4 to PE 3

\[ w_{43}^{\text{new}} = (E_4 \times x_{\text{out}}(4)) + w_{43} \]
\[ = (-0.2960 \times 0.4637) + (-0.4134) \]
\[ w_{43}^{\text{new}} = -0.5507 \]

From PE 5 to PE 2

\[ w_{52}^{\text{new}} = (E_5 \times x_{\text{out}}(5)) + w_{52} \]
\[ = (0.2880 \times 0.4306) + (-0.2794) \]
\[ w_{52}^{\text{new}} = -0.1554 \]

From PE 5 to PE 3

\[ w_{53}^{\text{new}} = (E_5 \times x_{\text{out}}(5)) + w_{53} \]
\[ = (0.2880 \times 0.4306) + (-0.6520) \]
\[ w_{53}^{\text{new}} = -0.5280 \]

From PE 6 to PE 2

\[ w_{62}^{\text{new}} = (E_6 \times x_{\text{out}}(6)) + w_{62} \]
\[ = (-0.0893 \times 0.6131) + (0.4604) \]
\[ w_{62}^{\text{new}} = 0.4057 \]

From PE 6 to PE 3

\[ w_{63}^{\text{new}} = (E_6 \times x_{\text{out}}(6)) + w_{63} \]
\[ = (-0.0893 \times 0.6131) + (-0.1221) \]
Therefore the 'new' weights for the network are:

\[ w_{63}^{new} = -0.1768 \]

\[ w_{42}^{new} = -0.2829 \quad w_{43}^{new} = -0.5507 \quad w_{74}^{new} = -0.3075 \]

\[ w_{52}^{new} = -0.1554 \quad w_{53}^{new} = -0.5280 \quad w_{75}^{new} = +0.7905 \]

\[ w_{62}^{new} = +0.4057 \quad w_{63}^{new} = -0.1768 \quad w_{76}^{new} = +0.0811 \]

6. Repeat the procedure from step 2 to 5 until the 'new' weights are the 'old' weights. This is continued for several cycles until an acceptable error is reached, i.e. a value close to the desired output.

This is very simple manual computation for a network using a backpropagation algorithm with a supervised learning. The NeuralWorks® professional package provides several alternative algorithms, the use of Bias, several instruments to monitor the network performance and other important tools.
Forward Propagation

At each PE a summation and activation

Input Layer  Hidden Layer  Output Layer

Figure A.1. Simple Backpropagation Network.
APPENDIX B

Full Generation Code for Modulus of Rupture

This Appendix contains the full generation code equation predicted by neural networks for the concrete strength versus the modulus of rupture obtained in chapter 5, equations (5.6) and (5.7).
/* Subroutine produced by NeuralWorks for Concrete strength versus Modus of Rupture*/

/* Recall-Only Run-time for <fcftp> */

/* Control Strategy is: <backprop> */

#if __STDC__
#define ARGS(x) x
#else
#define ARGS(x)
#endif

/* --- External Routines --- */

extern double exp ARGS((double));

/* *** MAKE SURE TO LINK IN YOUR COMPILERS MATH LIBRARIES *** */

#if __STDC__
int NN_Recall( void *NetPtr, float Yin[1], float Yout[1] )
#else
int NN_Recall( NetPtr, Yin, Yout )
#endif

void *NetPtr; /* Network Pointer (not used) */
float Yin[1], Yout[1]; /* Data */

{ float Xout[25], Xsum[25]; /* work arrays */
  long ICmpT; /* temp for comparisons */

  /* *** WARNING: Code generated assuming Recall = 0 *** */

  /* Read and scale input into network */
  Xout[2] = Yin[0] * (0.042105263) + (-1.1052632);

LAB107:
/* Generating code for PE 0 in layer 3 */
Xsum[3] = (float)(-0.76620275) + (float)(-0.99918461) * Xout[2];

/* Generating code for PE 1 in layer 3 */
Xsum[4] = (float)(-0.60372019) + (float)(-0.78286338) * Xout[2];

/* Generating code for PE 2 in layer 3 */
Xsum[5] = (float)(0.14605978) + (float)(1.0793796) * Xout[2];

/* Generating code for PE 3 in layer 3 */
Xsum[6] = (float)(-1.0665023) + (float)(-1.665948) * Xout[2];

/* Generating code for PE 4 in layer 3 */
Xsum[7] = (float)(-0.7186563) + (float)(-1.2456799) * Xout[2];

/* Generating code for PE 5 in layer 3 */
Xsum[8] = (float)(-0.32231224) + (float)(-0.76984376) * Xout[2];

/* Generating code for PE 6 in layer 3 */
Xsum[9] = (float)(0.55516249) + (float)(-1.2007288) * Xout[2];

/* Generating code for PE 7 in layer 3 */
Xsum[10] = (float)(-0.65146434) + (float)(-1.3861005) * Xout[2];

/* Generating code for PE 8 in layer 3 */
Xsum[11] = (float)(-0.16407533) + (float)(-0.64738035) * Xout[2];

/* Generating code for PE 9 in layer 3 */

/* Generating code for PE 10 in layer 3 */
Xsum[13] = (float)(-0.25128219) + (float)(-1.1231488) * Xout[2];

/* Generating code for PE 11 in layer 3 */
Xsum[14] = (float)(0.0058753048) + (float)(1.1885237) * Xout[2];

/* Generating code for PE 0 in layer 3 */
Xout[3] = 1.0 / (1.0 + exp(-Xsum[3]));
/* Generating code for PE 1 in layer 3 */
Xout[4] = 1.0 / (1.0 + exp(-Xsum[4]));
/* Generating code for PE 2 in layer 3 */
Xout[5] = 1.0 / (1.0 + exp(-Xsum[5]));
/* Generating code for PE 3 in layer 3 */
Xout[6] = 1.0 / (1.0 + exp(-Xsum[6]));
/* Generating code for PE 4 in layer 3 */
Xout[7] = 1.0 / (1.0 + exp(-Xsum[7]));
/* Generating code for PE 5 in layer 3 */
Xout[8] = 1.0 / (1.0 + exp(-Xsum[8]));
/* Generating code for PE 6 in layer 3 */
Xout[9] = 1.0 / (1.0 + exp(-Xsum[9]));
/* Generating code for PE 7 in layer 3 */
Xout[10] = 1.0 / (1.0 + exp(-Xsum[10]));
/* Generating code for PE 8 in layer 3 */
Xout[11] = 1.0 / (1.0 + exp(-Xsum[11]));
/* Generating code for PE 9 in layer 3 */
Xout[12] = 1.0 / (1.0 + exp(-Xsum[12]));
/* Generating code for PE 10 in layer 3 */
Xout[13] = 1.0 / (1.0 + exp(-Xsum[13]));
/* Generating code for PE 11 in layer 3 */
Xout[14] = 1.0 / (1.0 + exp(-Xsum[14]));
/* Generating code for PE 0 in layer 4 */
Xsum[15] = (float)(0.23606284) + (float)(-0.34144795) * Xout[3] +
(float)(-0.088125587) * Xout[4] + (float)(0.78090584) * Xout[5] +
(float)(-0.50964046) * Xout[6] + (float)(-0.33817393) * Xout[7] +
(float)(-0.14980897) * Xout[8] + (float)(-0.056997679) * Xout[9] +
(float)(-0.5251022) * Xout[10] + (float)(0.29646865) * Xout[11] +
(float)(0.52169693) * Xout[12];

Xsum[15] += (float)(-0.28413373) * Xout[13] +
(float)(0.75337368) * Xout[14];

/* Generating code for PE 1 in layer 4 */

Xsum[16] = (float)(-0.23317058) + (float)(-0.095218442) * Xout[3] +
(float)(-0.0027557316) * Xout[4] + (float)(-0.5768171) * Xout[5] +
(float)(0.62342918) * Xout[6] + (float)(0.41889441) * Xout[7] +
(float)(0.075185463) * Xout[8] + (float)(0.029678525) * Xout[9] +
(float)(0.20248657) * Xout[10] + (float)(0.25342181) * Xout[11] +
(float)(-0.74407566) * Xout[12];

Xsum[16] += (float)(0.26482615) * Xout[13] +
(float)(-0.12570286) * Xout[14];

/* Generating code for PE 2 in layer 4 */

Xsum[17] = (float)(-0.14074276) + (float)(-0.14202476) * Xout[3] +
(float)(0.0055213231) * Xout[4] + (float)(-0.057532519) * Xout[5] +
(float)(-0.017825821) * Xout[6] + (float)(0.074725471) * Xout[7] +
(float)(0.21745886) * Xout[8] + (float)(-0.19344166) * Xout[9] +
(float)(-0.31087324) * Xout[10] + (float)(-0.29493266) * Xout[11] +
(float)(0.3222793) * Xout[12];

Xsum[17] += (float)(-0.052200522) * Xout[13] +
(float)(0.21744959) * Xout[14];

/* Generating code for PE 3 in layer 4 */

Xsum[18] = (float)(0.21248251) + (float)(-0.062593833) * Xout[3] +
(float)(-0.2872259) * Xout[4] + (float)(0.042985857) * Xout[5] +
(float)(-0.17981297) * Xout[6] + (float)(-0.10749779) * Xout[7] +
(float)(-0.005183947) * Xout[8] + (float)(-0.39645219) * Xout[9] +
(float)(-0.5195424) * Xout[10] + (float)(-0.013343313) * Xout[11] +
(float)(0.47206047) * Xout[12];

Xsum[18] += (float)(-0.11497389) * Xout[13] +
(float)(0.083029673) * Xout[14];

/* Generating code for PE 4 in layer 4 */

Xsum[19] = (float)(0.0066736671) + (float)(0.080481082) * Xout[3] +
(float)(-0.15034163) * Xout[4] + (float)(0.029113185) * Xout[5] +
(float)(-0.085055746) * Xout[6] + (float)(-0.30220094) * Xout[7] +
(float)(0.087724298) * Xout[8] + (float)(0.25097185) * Xout[9] +
(float)(-0.19879386) * Xout[10] + (float)(0.30397969) * Xout[11] +
(float)(-0.059420008) * Xout[12];

Xsum[19] += (float)(-0.14391561) * Xout[13] +
(float)(0.33417496) * Xout[14];

/* Generating code for PE 5 in layer 4 */

Xsum[20] = (float)(-0.040932432) + (float)(0.31569684) * Xout[3] +
(float)(-0.24787149) * Xout[4] + (float)(0.073610328) * Xout[5] +
(float)(0.076208606) * Xout[6] + (float)(0.017323652) * Xout[7] +
(float)(0.27031592) * Xout[8] + (float)(0.0070682899) * Xout[9] +
(float)(0.31263754) * Xout[10] + (float)(0.23203792) * Xout[11] +
(float)(-0.36220065) * Xout[12];

Xsum[20] += (float)(0.36745802) * Xout[13] +
(float)(-0.47743654) * Xout[14];

/* Generating code for PE 6 in layer 4 */

Xsum[21] = (float)(-0.33379903) + (float)(-0.028827731) * Xout[3] +
(float)(0.17283779) * Xout[4] + (float)(-0.36399084) * Xout[5] +
(float)(0.40099321) * Xout[6] + (float)(0.3094174) * Xout[7] +
(float)(-0.017143365) * Xout[8] + (float)(0.19775648) * Xout[9] +
(float)(0.16214246) * Xout[10] + (float)(0.054619297) * Xout[11] +
(float)(-0.35757539) * Xout[12];
Xsum[21] += (float)(0.23889661) * Xout[13] +
(float)(-0.22647703) * Xout[14];

/* Generating code for PE 7 in layer 4 */
Xsum[22] = (float)(-0.18841542) + (float)(0.19754705) * Xout[3] +
(float)(0.20955321) * Xout[4] + (float)(-0.51844031) * Xout[5] +
(float)(0.54046267) * Xout[6] + (float)(0.0038342031) * Xout[7] +
(float)(0.0047981148) * Xout[8] + (float)(0.19214241) * Xout[9] +
(float)(0.34254292) * Xout[10] + (float)(0.35343501) * Xout[11] +
(float)(-1.0228976) * Xout[12];
Xsum[22] += (float)(0.0018936887) * Xout[13] +
(float)(-0.60670567) * Xout[14];

/* Generating code for PE 8 in layer 4 */
Xsum[23] = (float)(-0.066219144) + (float)(-0.29918492) * Xout[3] +
(float)(-0.31510872) * Xout[4] + (float)(0.35389706) * Xout[5] +
(float)(-0.34545916) * Xout[6] + (float)(-0.36934546) * Xout[7] +
(float)(0.0051398743) * Xout[8] + (float)(-0.05263469) * Xout[9] +
(float)(-0.10092045) * Xout[10] + (float)(0.039514996) * Xout[11] +
(float)(0.57129878) * Xout[12];
Xsum[23] += (float)(0.22911224) * Xout[13] +
(float)(0.24085249) * Xout[14];

/* Generating code for PE 0 in layer 4 */
Xout[15] = 1.0 / (1.0 + exp( -Xsum[15] ));
/* Generating code for PE 1 in layer 4 */
Xout[16] = 1.0 / (1.0 + exp( -Xsum[16] ));
/* Generating code for PE 2 in layer 4 */
Xout[17] = 1.0 / (1.0 + exp( -Xsum[17] ));
/* Generating code for PE 3 in layer 4 */
Xout[18] = 1.0 / (1.0 + exp( -Xsum[18] ));
/* Generating code for PE 4 in layer 4 */
Xout[19] = 1.0 / (1.0 + exp( -Xsum[19] ));
/* Generating code for PE 5 in layer 4 */
Xout[20] = 1.0 / (1.0 + exp( -Xsum[20] ));
/* Generating code for PE 6 in layer 4 */
Xout[21] = 1.0 / (1.0 + exp( -Xsum[21] ));
/* Generating code for PE 7 in layer 4 */
Xout[22] = 1.0 / (1.0 + exp( -Xsum[22] ));
/* Generating code for PE 8 in layer 4 */
Xout[23] = 1.0 / (1.0 + exp( -Xsum[23] ));
/* Generating code for PE 0 in layer 5 */
Xsum[24] = (float)(0.099840201) + (float)(0.91322863) * Xout[15] +
(float)(-0.80489773) * Xout[16] + (float)(0.142288) * Xout[17] +
(float)(0.44045058) * Xout[18] + (float)(0.028653085) * Xout[19] +
(float)(-0.46781901) * Xout[20] + (float)(-0.52230942) * Xout[21] +
(float)(-0.97042954) * Xout[22] + (float)(0.57755297) * Xout[23];
Xout[24] = 1.0 / (1.0 + exp( -Xsum[24] ));
/* De-scale and write output from network */
Yout[0] = Xout[24] * (7.4149996) + (-0.98199995);
/* Generating code for PE 0 in layer 5 */

return( 0 );

}
APPENDIX C

This Appendix contains the trained and tested networks of the stress-strain relationships for $f'_c = 20, 30, 35$ and $40 \text{ N/mm}^2$ of Figures 5.11, 5.12, 5.13 and 5.13 respectively.
Stress vs. Strain: $F_c = 28 N/mm^2$; Cycle $= 158853$; RMS $= .81$; CR $= .9745$; CR $= 1.0$

(a) Trained Network.

(b) Trained and Tested Network.

Figure C.1. Stress-strain Relationship Networks for $f_c = 20 N/mm^2$. 
Figure C.2. Stress-strain Relationship Networks for $f'_c = 30 N/mm^2$. 

(a) Trained Network.

(b) Trained and Tested Network.
Figure C.3. Stress-strain Relationship Networks for $f_c = 35N/mm^2$. 

(a) Trained Network.

(b) Trained and Tested Network.
Figure C.4. Stress-strain Relationship Networks for $f_c = 40 \text{N/mm}^2$. 
PUBLICATIONS


