A RECOMMENDED NEURAL TRIP DISTRIBUTON MODEL

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

SERKAN TAPKIN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

THE DEPARTMENT OF CIVIL ENGINEERING

JANUARY 2004

Approval of the Graduate School of Natural and Applied Sciences

_____

Prof.Dr. Canan Özgen

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

_____

Prof.Dr. Erdal Çokça

Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

_____

Prof.Dr. Özdemir Akyılmaz

Supervisor

Examining Committee Members

Prof.Dr. Özdemir Akyılmaz _____

Prof.Dr. Ayhan İnal _____

Prof.Dr. Ali Türel _____

Asst.Prof.Dr. Hikmet Bayırtepe _____

Asst.Prof.Dr. Rifat Sönmez _____

ABSTRACT

A RECOMMENDED NEURAL TRIP DISTRIBUTON MODEL

TAPKIN, Serkan

Ph.D., Department of Civil Engineering

Supervisor: Prof.Dr. Özdemir Akyılmaz

January 2004, 167 pages

In this dissertation, it is aimed to develop an approach for the trip distribution element which is one of the important phases of four-step travel demand modelling. The trip distribution problem using back-propagation artificial neural networks has been researched in a limited number of studies and, in a critically evaluated study it has been concluded that the artificial neural networks underperform when compared to the traditional models. The underperformance of back-propagation artificial neural networks appears to be due to the thresholding the linearly combined inputs from the input layer in the hidden layer as well as thresholding the linearly combined outputs from the hidden layer in the output layer. In the proposed neural trip distribution model, it is attempted not to threshold the linearly combined outputs from the hidden layer in the output layer. Thus, in this approach, linearly combined

inputs are activated in the hidden layer as in most neural networks and the neuron in the output layer is used as a summation unit in contrast to other neural networks. When this developed neural trip distribution model is compared with various approaches as modular, gravity and back-propagation neural models, it has been found that reliable trip distribution predictions are obtained.

Keywords: Trip Distribution, Gravity Model, Back-Propagation Artificial Neural Networks, Neural Trip Distribution Model, Modular Neural Network

ÖZ

SEYAHAT DAĞILIMI İÇİN YENİ BİR SİNİR AĞI MODELİ

TAPKIN, Serkan

Doktora, İnşaat Mühendisliği Bölümü

Tez yöneticisi: Prof.Dr. Özdemir Akyılmaz

Ocak 2004, 167 sayfa

Bu çalışmada, dört basamaklı ulaşım talep modellemesinin en önemli unsurlarından biri olan seyahat dağılım modeli için yeni bir yaklaşım modelinin geliştirilmesi amaçlanmıştır. Çok sınırlı sayıda mevcut olan çalışmalardan yola çıkılarak, geriye yayılma yapay sinir ağları kullanılarak seyahat dağılımlarını hesap etme problemi ele alınmış ve geriye yayılma yapay sinir ağları ile elde edilen neticelerin yetersiz olduğu görülmüştür. Geriye yayılma yapay sinir ağları kullanılarak yapılmış olan çalışmalarda, bahsedilen sinir ağı tipinin mimarisinden dolayı, ağırlıklandırılmış girdileri toplayan nöronlar, bu sonuçları lineer olmayan bir eşik fonksiyondan geçirmekte ve bu işlem hem saklı katmanda ve hem de çıktı katmanında olmak üzere iki defa yapılmaktadır. Bu modellerde, çıktı katmanında bir kez daha eşik fonksiyonundan geçirilen ağırlıklandırılmış girdiler, hata

fonksiyonu sonucu hesap edilen değerlere etkimekte ve bu değerler olduğundan farklı bir hale gelmektedir. Önerilen yeni sinirsel seyahat dağılımı modelinde bu sorun, girdi katmanı değerlerinin, gizli katmandan sonra, bu katmandan çıkan eşik fonksiyonundan geçen değerlerle bir kez daha çarptırılıp, bu yeni ağırlıklandırılmış değerlerin ise tekrar bir eşik fonksiyonundan geçirilmemesiyle çözümlenmiştir. Geliştirilen bu sinirsel seyahat dağılım modeli ile, bu yaklaşıma benzer modüler modelden, gravite modelinden ve geriye yayılma yapay sinir ağlarından çok daha iyi ve güvenilir seyahat dağılımı tahminleri yapılabildiği gözlemlenmektedir.

Anahtar Kelimeler: Seyahat Dağılımı, Gravite Modeli, Geriye Yayılma Yapay Sinir Ağları, Sinirsel Seyahat Dağılım Modeli, Modüler Sinir Ağı

TO MY GRANDMOTHER

AND MOTHER

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude and appreciation to my supervisor Prof.Dr. Özdemir Akyılmaz, whose assistance, advice and critical review were invaluable in carrying out and completing this study.

I also deeply thank my grandmother Nazmiye Tapkın and mother Serpil Tapkın whose encouragement and continuous moral support are highly appreciated.

TABLE OF CONTENTS

Page

ABSTRACT .....…… iii

ÖZ .....……..... v

ACKNOWLEDGEMENTS ……..... viii

TABLE OF CONTENTS ...…….... ix

LIST OF TABLES ……..…….…… xii

LIST OF FIGURES ……..…… xvi

CHAPTER

1. INTRODUCTION ……... 1

2. SPATIAL INTERACTION THEORY AND TRANSPORTATION PLANNING ……... 3

2.1 General Information on Spatial Interaction Theory ……... 3

2.2 A Family of Spatial Interaction Models ……..…….…... 5

2.2.1 The Unconstrained Model ……..... 6

2.2.2 The Production-Constrained Model …….. 7

2.2.3 The Attraction-Constrained Model ……..... 9

2.2.4 The Production-Attraction Constrained Model
or Doubly Constrained Model) ……..... 12
2.3 Other Forms of Spatial Interaction Models ……. 14
2.4 Spatial Interaction Modelling in Urban Transportation Planning ... 15
2.5 Trip Distribution Models ……. 17
2.5.1 Growth-factor Methods ……..……..... 19
2.5.2 Theoretically Based Models ……..... 25
3. BASIS OF ARTIFICIAL NEURAL NETWORKS …….…….... 36
3.1 General ……..... 36
3.2 Basic Concepts of Artificial Neural Networks ……..... 39
3.3 Types of Artificial Neural Networks ….…….. 43
3.4 Neural Networks As Function Approximators……. 44
3.5 Back-propagation Network Model …….. 47
4. DEVELOPMENT OF THE NEURAL TRIP DISTRIBUTION MODEL
(NETDIM) ….....…….……. 59
4.1 Neural Network Background in Transportation Engineering ….. 59
4.2 Development of Neural Trip Distribution Model ….……..... 70
4.3 Modular Neural Networks ……..…….……... 71
4.4 Neural Trip Distribution Model (NETDIM) ….……. 74
5. ANALYSIS OF VARIOUS APPROACHES USING THE DIFFERENT
SIZED NETWORKS …….... 78
5.1 Data Acquisition and Generation of Different Sized Networks .. 79
5.2 Training and Testing the Models ……. 82
5.2.1 Training and Testing of Back-Propagation Neural Models .. 82

5.2.2 Training and Testing of NETDIM and Modular Model Approaches 90

5.2.3 Calibrating and Testing the Gravity Model 91

5.3 Testing Results for all Models 92

5.4 Comparison of Results 94

6. SUMMARY AND CONCLUSIONS 100

REFERENCES..... 103

APPENDICES 108

VITA 167

LIST OF TABLES

TABLE

2.1 Possible measures of origin propulsiveness and destination attractiveness .. 5

5.1 Flow data of a three-node sized network used in training 83

5.2 Normalised flow data of a three-node sized network used in training …..... 83

5.3 Distance data of a three-node sized network used in training ……..... 84

5.4 Normalised distance data of a three-node sized network used in training 84

5.5 Flow data of a three-node sized network used in testing ……..... 85

5.6 Distance data of a three-node sized network used in testing ……..... 85

5.7 Normalised flow data of a three-node sized network used in testing ……..... 85

5.8 Normalised distance data of a three-node sized network used in testing 85

5.9 Neural network training and testing parameters ……..... 88

5.10 A sample distance matrix ……..... 92

5.11 The predicted flow values obtained through the simulation of back-propagation neural networks .....……. 93

5.12 The predicted flow values obtained through testing of NETDIM network .. 93

5.13 The predicted flow values obtained through testing of modular network …. 94

5.14 The predicted flow values obtained through testing of unconstrained gravity model …….. 94

5.15 Test results of trained models ……..... 95

A.1 Literature Survey ……... 108

B.1 Zonal trips taken from Bursa Transportation Master Plan ……..... 136

B.2 Zonal distances taken from Bursa Transportation Master Plan ……..... 137

C.1 The predicted flow values for three-node sized network obtained through the simulation of back-propagation neural networks ……... 139

C.2 The predicted flow values for three-node sized network obtained through testing of the NETDIM network ……... 139

C.3 The predicted flow values obtained for three-node sized network through testing of the modular network …….. 140

C.4 The predicted flow values for three-node sized network obtained through testing of the unconstrained gravity model ……..... 140

C.5 The predicted flow values obtained for six-node sized network through the simulation of back-propagation neural networks ……..... 141

C.6 The predicted flow values obtained for six-node sized network through testing of the NETDIM network ……..... 141

C.7 The predicted flow values obtained for six-node sized network through testing of the modular network …….. 142

C.8 The predicted flow values obtained for six-node sized network through testing of the unconstrained gravity model ……... 142

C.9 The predicted flow values obtained for nine-node sized network through the simulation of back-propagation neural networks ……..... 143

C.10 The predicted flow values obtained for nine-node sized network through testing of the NETDIM network ……..... 144

C.11 The predicted flow values obtained for nine-node sized network through testing of the modular network …….....…… 144

C.12 The predicted flow values obtained for nine-node sized network through testing of the unconstrained gravity model ……..... 145

C.13 The predicted flow values obtained for twelve-node sized network through the simulation of back-propagation neural networks …..……..... 146

C.14 The predicted flow values obtained for twelve-node sized network through testing of the NETDIM network …….....…….. 147

C.15 The predicted flow values obtained for twelve-node sized network through testing of the modular network ……...…… 148

C.16 The predicted flow values obtained for twelve-node sized network through testing of the unconstrained gravity model ...……..... 149

C.17 The predicted flow values obtained for fifteen-node sized network through the simulation of back-propagation neural networks ...…….…. 151

C.18 The predicted flow values obtained for fifteen-node sized network through testing of the NETDIM network ..……...….... 152

C.19 The predicted flow values obtained for fifteen-node sized network through testing of the modular network .....…….....…….. 153

C.20 The predicted flow values obtained for fifteen-node sized network through testing of the unconstrained gravity model ...…….....…… 154

C.21 The predicted flow values obtained for eighteen-node sized network through the simulation of back-propagation neural networks ...…….....…… 155

C.22 The predicted flow values obtained for eighteen-node sized network through testing of the NETDIM network ..……...…….. 156

C.23 The predicted flow values obtained for eighteen-node sized network through testing of the modular network ..……...…….. 157

C.24 The predicted flow values obtained for eighteen-node sized network through testing of the unconstrained gravity model ..…… 158

C.25 The predicted flow values obtained for twenty-one-node sized network through the simulation of back-propagation neural networks ...……....…….. 159

C.26 The predicted flow values obtained for twenty-one-node sized network through testing of the NETDIM network ..…….....…….. 160

C.27 The predicted flow values obtained for twenty-one -node sized network through testing of the modular network ..…….....…… 161

C.28 The predicted flow values obtained for twenty-one-node sized network through testing of the unconstrained gravity model ...…….. 162

C.29 The predicted flow values obtained for twenty-four-node sized network through the simulation of back-propagation neural networks ...……...…… 163

C.30 The predicted flow values obtained for twenty-four-node sized network through testing of the NETDIM network ..……....…….. 164

C.31 The predicted flow values obtained for twenty-four-node sized network through testing of the modular network ..……...…….. 165

C.32 The predicted flow values obtained for twenty-four-node sized network through testing of the unconstrained gravity model ...……..…… 166

LIST OF FIGURES

FIGURE

3.1 Outline of the perceptron model ……....… 39

3.2 Neural information processing as depicted in Fu ……...… 40

3.3 Linear perceptron architecture ……..... 48

3.4 Back-propagation architecture ……..... 50

3.5 A sigmoid transfer function used in back-propagation network ……..… 52

3.6 The shape of the first derivative of the sigmoid transfer function ……..… 54

4.1 A system architecture of modular neural networks ……..….. 72

4.2 Schematical representation of the recommended NTDM ……..… 75

5.1 Training and testing the models ……....…. 81

5.2 Network size vs. RMSE values for all models ……....….. 96

CHAPTER 1

INTRODUCTION

Continuing expansion of cities with the development of societies and technology through the ages make the existing transportation systems inadequate to meet the increasing demands due to the difficulty and the complexity of the daily movements of goods and people. In order to provide the flow of traffic freely and safely from one place to another without encountering any congestion problem, it might be necessary to improve the existing facilities or to provide new facilities.

The first thing that should be tried is to look for alternatives of improving the existing facilities. Building of new facilities is always the case that should be well studied. In other words, it may not be desirable to introduce new facilities which may be due to the harmony already established in the existing system or which may not be desirable to disturb it. In general, with the increase of travel demand, there is a growing interest to preserve the nature and the environment and this problem is particularly crucial in the urban areas. Second thing that should be focused on is the factor of economy. Construction of new facilities needs very large expenses. It is often argued that the resources required for new transport facilities would return higher community benefits if they were invested in other types of services.

Construction of a new facility needs long span of time and generally, this type of improvement is included in the long-range plans.

All these points show the necessity of a comprehensive transportation planning within a regional area. The society demands increasing care and professional competence in the planning and operation of all transportation systems.

Transportation planning is the study of present transportation patterns in relation to present population, economy and land use of an area, the estimation of future transportation patterns related to prediction of future population, land use and economy, the design of alternative transportation networks and facilities, the evaluation of alternatives and the adoption of a transportation plan with proposals for its implementation, scheduling and financing.

During the last decade, neural networks have been used as alternative tools to traditional planning efforts. In these studies, some important ideas and application procedures are developed which are all discussed in this thesis.

This dissertation is arranged in the following manner; in the second chapter, it is aimed at familiarising the neural network practitioners with spatial interaction modelling and transportation planning, the aim of the inclusion of the third chapter into the thesis is to acquaint the transportation planners with the basic concepts of neural networks, in the fourth chapter, neural network background in transportation engineering is presented and a recommended neural network approach is proposed, chapter five is dealt with the analyses of various approaches as well as the recommended approach and finally, summary of the study and conclusions are drawn in chapter six.

CHAPTER 2

SPATIAL INTERACTION THEORY AND TRANSPORTATION PLANNING

2.1 General Information on Spatial Interaction Theory

Over the past decades, a number of modelling approaches have been developed to apprehend the spatial processes associated with the development of a transportation system. All these approaches, commonly known as spatial interaction modelling efforts are aimed at determining the mobility of individuals over space that results from their utilities. Such activity related movements, which are in the form of trips undertaken to achieve particular objectives, as getting to work, are all derived from the utilities of individuals. These purpose-oriented trips largely determine the spatial mobility over spatially separated origins and destinations.

Thus, spatial interaction models can be regarded as mathematical descriptions of those spatial flows. These models serve a dual purpose upon their calibration: one

is to predict flows between origins and destinations and the other is to analyse the marginal changes in those predicted flows.

The starting point for spatial interaction modelling is a matrix of flows between a set of origins and a set of destinations, which can be labelled as i and j respectively. Thus, the task is to represent the number of trips T_{ij} between any i and j, as a function of origin-destination characteristics (i.e. zonal attributes) and the spatial separation between those origins and destinations.

Essentially, spatial interaction models contain three basic elements: a measurement of the spatial separation between origins and destinations; a measurement of the propulsiveness of each origin; a measurement of the attractiveness of each destination.

Each of the above-mentioned elements can be measured in different ways depending on the type of spatial interaction being modelled. Usually, distances between origins and destinations are used to measure spatial separation although cost and time can also be used. Regardless of how spatial separation is measured, it is one of the most important determinants of spatial interaction patterns. For example, while an individual is likely to choose a supermarket close to home, it may not always be the one that is the most attractive. A supermarket farther away but offering a wider range of products or cheaper prices might be more attractive. This trade-off between zonal attributes and separation is the essence of interaction modelling.

Propulsiveness of origin and attractiveness of destinations can be described by the zonal attributes (i.e. origin and destination characteristics) such as population, income and employment as can be seen in Table 2.1. These attributes together with

zonal separation can be used for spatial interaction modelling to determine the number of zonal movements (trips).

Table 2.1 Possible measures of origin propulsiveness and destination attractiveness

Type of spatial interaction	**Origin Attributes**	**Destination Attributes**
Trips, Communication, Migration, Others	Population, Household, Income, Car Ownership Preferences, Others	Employment, Establishments (size), Land use, Parking charges, Others

2.2 A Family of Spatial Interaction Models

In order to predict the level of interaction between origin-destination (O-D) pairs, it is necessary to have the following information about the zones: a measure of origin propulsiveness, a measure of destination attractiveness and a measure of spatial separation. Different levels of information available on propulsiveness, attractiveness and separation results four different forms of spatial interaction modelling which are collectively referred to as a "family of spatial interaction models" [1].

2.2.1 The Unconstrained Model

If spatial separation is known but no information on how many trips originate at each origin and how many trips terminate at each destination is available, the latter two values need to be estimated within the model by surrogate variables. The appropriate model in such a situation is known as an unconstrained model with the following general form [2]:

$$T_{ij} = kV_{i1}^{\alpha 1}V_{i2}^{\alpha 2}.....V_{ip}^{\alpha p}W_{j1}^{\lambda 1}W_{j2}^{\lambda 2}.....W_{jq}^{\lambda q}d_{ij}^{\delta} \qquad \textbf{2.1}$$

where;

T_{ij} = number of trips that begin at origin i and end at destination j

V_i = origin attribute that is a surrogate for the number of trips beginning at i

W_j = destination attribute that is a surrogate for the number of trips ending at j

d_{ij} = spatial separation (assumed here to be measured by distance) between origin i and destination j

The number of origin and destination variables included in the model depends on the type of spatial interaction being modelled and the information available. Here the number of origin variables is denoted by i and the number of destination variables is denoted by j. In some situations, there might only be one origin attribute and one destination attribute in the model. The parameters α, λ and β reflect the influence of each of the exogenous variables in the model. For example, if V_{i1} were

population, it would be expected that $\alpha 1$ would be positive: as the population of an origin increased, a greater volume of interaction would emanate from it, with everything else being equal.

Similarly, β would normally be negative: as the distance between any two zones increases, the volume of interaction between them decreases, everything else being equal. The exact rate at which interaction decreases or "decays" as distance increases, is a matter for empirical evaluation. It might be interesting to estimate these parameters in different systems and to compare spatial behaviour in different systems and over time. For instance, the value of the distance-decay parameter, β, has been shown to vary across different economic systems, over time, by income group and by gender [3]. Generally, the deterrence of distance for interaction is less in more developed economies and there is a gradual reduction in distance-decay over time as transportation facilities improve and as knowledge about different locations increases. Finally, the parameter k in the model is a scale factor that simply balances the units on both sides of the equation and is of no great behavioural importance.

2.2.2 The Production-Constrained Model

If spatial separation and the number of trips emanating from each origin are known but no information on the number of trips that terminate at each destination is available, the appropriate model for this situation is known as a production-constrained model which has the form [2] as:

$$T_{ij} = O_i \frac{W_{j1}^{\lambda 1} W_{j2}^{\lambda 2} .....W_{jq}^{\lambda q} d_{ij}^{\beta}}{\sum_j W_{j1}^{\lambda 1} W_{j2}^{\lambda 2} .....W_{jq}^{\lambda q} d_{ij}^{\beta}} \qquad \textbf{2.2}$$

where;

O_i = total number of trips beginning at origin i

O_i replaces the surrogate origin propulsiveness variables in the unconstrained version. The model essentially distributes the number of trips from an origin in proportion to the relative attractiveness of each destination as measured by the numerator of the above equation. The denominator acts as a constraint on the predicted trips so that [2];

$$\sum_j T_{ij}^{*} = O_i \text{ for every origin} \qquad \textbf{2.3}$$

where T_{ij}^{*} is the predicted value of T_{ij} from the model. Alternatively, the model can also be represented as [2]:

$$T_{ij} = A_i O_i W_{j1}^{\lambda 1} W_{j2}^{\lambda 2} .....W_{jq}^{\lambda q} d_{ij}^{\beta} \qquad \textbf{2.4}$$

where;

$$A_i = \left[\sum_j W_{j1}^{\lambda 1} W_{j2}^{\lambda 2} .....W_{jq}^{\lambda q} d_{ij}^{\beta} \right]^{-1} \qquad \textbf{2.5}$$

Production-constrained forms of spatial interaction models are encountered very frequently, particularly in retailing where estimates of the number of trips leaving each origin can be obtained from census population data. The models are used to discover the characteristics of stores that attract consumers and to predict the proportion of people in each residential neighbourhood shopping at each store. Once such a store's potential market and its impact on the sales at existing stores are known, then the model can be used to estimate sales at a variety of possible locations to aid in the optimal strategy to attract more customers to a store. For example, more customers would be attracted to a store if the number of parking spaces were doubled or if parking charges were reduced slightly.

2.2.3 The Attraction-Constrained Model

If spatial separation and the numbers of trips terminating at each destination are known but there is no information about the number of trips beginning at each origin, then an attraction-constrained model is the most appropriate one since the known inflow totals into each destination and a set of surrogate variables are used to represent the total number of trips beginning at each origin. The model has the general form [2] as:

$$T_{ij} = D_j \frac{V_{i1}^{\alpha 1} V_{i2}^{\alpha 2} .....V_{ip}^{\alpha p} d_{ij}^{\beta}}{\sum_j V_{i1}^{\alpha 1} V_{i2}^{\alpha 2} .....V_{ip}^{\alpha p} d_{ij}^{\beta}} \qquad \textbf{2.6}$$

where;

D_j = total number of trips terminating at destination j

D_j replaces the destination attractiveness surrogates in the unconstrained model. The attraction-constrained model is the complement of the production-constrained model. In this case, the model allocates the number of trips terminating at a destination across the various origins in proportion to the relative propulsiveness of each origin as measured by the numerator of the above equation. The denominator again acts as a constraint on the predicted trips but in this case the constraint is on the number of predicted trips into each destination so that [2];

$$\sum_i T_{ij}^* = D_j \text{ for every destination} \qquad \textbf{2.7}$$

where T_{ij}^* again represents the predicted value of T_{ij} from the model. Alternatively, the model can also be represented as [2]:

$$T_{ij} = B_j D_j V_{i1}^{\alpha 1} V_{i2}^{\alpha 2} .....V_{ip}^{\alpha p} d_{ij}^{\beta} \qquad \textbf{2.8}$$

where;

$$B_j = \left[\sum_i V_{i1}^{\alpha 1} V_{i2}^{\alpha 2} .....V_{ip}^{\alpha p} d_{ij}^{\beta} \right]^{-1} \qquad \textbf{2.9}$$

The terms A_i and B_j in the production-constrained and attraction-constrained equations are sometimes referred to as "balancing factors" since they constrain the row and column totals of the predicted flow matrix respectively.

Attraction-constrained spatial interaction models are generally used less frequently than production-constrained models. However, they are found useful in some applications [2]. Such applications are:

- to estimate the demand for housing in different residential neighbourhoods due to the location of a new enterprise.
- to analyse geographic patterns in student enrollments where the total number of students attending a university is known.
- to determine the characteristics of places that affect the number of students attending from different origins.
- to derive "expected" numbers of students enrolling from each origin (the predicted trips from the model).
- to identify areas from which the university is drawing more students than expected, given the characteristics of these areas.

Obviously, these applications could easily be extended to retailing or any other area where there is competition to attract individuals.

2.2.4 The Production-Attraction Constrained Model (or Doubly Constrained Model)

If the total number of trips beginning at each origin, and the total number of trips terminating at each destination are known, then the production-attraction constrained, or doubly constrained model is the most appropriate one. This has the form [2] as:

$$T_{ij} = A_i O_i B_j D_j d_{ij}^{\beta} \qquad \textbf{2.10}$$

where;

$$A_i = \left[\sum_j B_j D_j d_{ij}^{\beta} \right]^{-1} \qquad \textbf{2.11}$$

and

$$B_j = \left[\sum_j A_i O_i d_{ij}^{\beta} \right]^{-1} \qquad \textbf{2.12}$$

The A_i and B_j terms have to be estimated iteratively and they ensure that the constraints in equations 2.3, 2.4 and 2.5 are met.

The doubly constrained model allocates trips to the cells of the flow matrix (the origin-destination links) given row and column totals. It does this solely on the basis of the distance between origins and destinations. Its use has mainly been confined to transportation applications where the total numbers of commuters

leaving each residential neighbourhood are known and the total numbers of employment opportunities in each destination are also known. What is unknown, and can be predicted from the model, is the distribution of trips between origins and destinations.

Although a detailed discussion of the nature and methods of calibrating spatial interaction models is beyond the scope of this dissertation, it is worth mentioning very briefly how the parameters in a spatial interaction model might be estimated.

In order to use a spatial interaction model to answer questions such as those discussed above, it is necessary to obtain values for the model's parameters. If the purpose of the model is to obtain information on the way either origin or destination characteristics affect spatial interaction patterns, the model must be calibrated with some known flow data. This is done by finding the values of the parameters that maximise some criterion, such as the accuracy with which the model replicates the known flow pattern. Parameter values that are far from their expected values will tend to produce very poor estimates of the flows, while those close to their expected values will tend to produce estimates that are more accurate. It is sometimes useful to determine how sensitive are a model's predictions to variations in the parameters used to derive the predictions.

If flow data are not available, model calibration as described above is impossible and "educated guesses" have to be made for parameter values. This could be done, for example, by surveying the results of previous calibrations of similar models. However, in practice this is impeded by difficulties and should be avoided where possible as it is rare to find exactly the same spatial interaction model

with the same combinations of explanatory variables being used in different applications. In addition, it appears that the parameters of spatial interaction models are quite context dependent [4].

2.3 Other Forms of Spatial Interaction Models

The forms of spatial interaction models described above are referred to as "global" models because they provide a single set of parameter estimates for the whole flow system being analysed. For instance, in each of the above-mentioned models, a single estimate of distance-decay is obtained when the model is calibrated. In most circumstances, it would be useful to obtain more detailed information on the flow system and to obtain separate distance-decay parameter estimates for each origin in the system or for each destination. This is quite easy to achieve by calibrating what are known as origin-specific models. An origin-specific form of the production-constrained model is given in the following form [2]:

$$T_{ij} = A_i O_i W_{j1}^{\lambda 1(i)} W_{j2}^{\lambda 2(i)} .....W_{jq}^{\lambda q(i)} d_{ij}^{\beta(i)} \qquad \textbf{2.13}$$

where;

$$A_i = \left[\sum_j W_{j1}^{\lambda 1(i)} W_{j2}^{\lambda 2(i)} .....W_{jq}^{\lambda q(i)} d_{ij}^{\beta(i)} \right]^{-1} \qquad \textbf{2.14}$$

In this model, each parameter is specific to origin *i* so that the spatial distribution of a parameter can be mapped to examine any spatial patterns in the

behaviour represented by that matter. Interestingly, there is a long history of estimated origin-specific distance-decay parameters exhibiting spatial patterns in which centrally located origins have less negative parameter estimates than peripheral ones [5]. The interpretation of such a pattern has provoked intense debate at times, but it appears that the distribution is due to a miss-specification bias which can easily be eliminated by the addition of a variable measuring the proximity of a destination to all the other destinations [6]. The models that result are termed "competing destination models" and have been shown to remove the previously observed spatial pattern of origin-specific, distance-decay parameters. It is interesting to note that the miss-specification of spatial interaction models became evident only when local forms of the models were calibrated [2].

For completeness, a destination-specific form of the attraction-constrained model described previously, is [2]:

$$T_{ij} = B_j D_j V_{i1}^{\alpha 1(j)} V_{i2}^{\alpha 2(j)} .....V_{ip}^{\alpha p(j)} d_{ij}^{\beta(j)} \qquad \textbf{2.15}$$

where;

$$B_j = \left[\sum_i V_{i1}^{\alpha 1(j)} V_{i2}^{\alpha 2(j)} .....V_{ip}^{\alpha p(j)} d_{ij}^{\beta(j)} \right]^{-1} \qquad \textbf{2.16}$$

2.4 Spatial Interaction Modelling in Urban Transportation Planning

Attempts at understanding regularities in patterns of spatial flows began with the observation that the movement of people between cities was analogous to the

gravitational attraction between solid bodies. That is, greater numbers of migrants were observed to move between larger cities than smaller ones, ceteris paribus (other things being equal), and between cities that were closer together than between cities that were farther apart, ceteris paribus. This led to the proposal of a simple mathematical model to predict migration flows between origins and destinations based on the Newtonian gravity model.

However, there was no theoretical justification for this form of model until Wilson [7] derived doubly constrained gravity model based on entropy-maximising principles.

This model has been extensively used by planners to analyse an individual segment or a link of a metropolitan transportation system. These efforts experienced by planners are found not to be adequate, and the model is often misleading when perceived from the point of view of long-range planning efforts of transportation facilities. This systems approach to transportation planning and development, where flows in transport networks are analysed in an integrated manner, is considered necessary for a proper comprehensive and coordinated transportation plan. This approach is commonly termed as a continuous, coordinated and comprehensive transportation planning process with the well-known four-step procedure of trip generation, trip distribution, modal split and traffic assignment. Since the essence of spatial interaction modelling is to predict level of interaction between zones, then the zonal interaction in this four-step planning process can be described by the number of trips distributed between pairs of zones.

The four-step planning process can also be termed as segmental aggregate demand modelling. In this modelling, trip generation is concerned with the number

of trips per time period made to or from a given areal unit or zone (regardless of the trips' origins or destinations), trip distribution is involved with finding the zones to or from which the generated trips are directed, mode choice is related with the determination of the particular mode of transportation used for the zone-to-zone trips, and finally, trip assignment is dealt with the particular route selected by travellers going between each pair of zones on each mode of transport.

2.5 Trip Distribution Models

An important step in the sequential four-step procedure is to distribute the zonal trips between all pairs of zones *i* and *j*, where *i* is the trip-producing zone and *j* is the trip-attracting zone of the pair. The rationale of trip distribution is that all trip-attracting zones *j* in the region are in competition with each other to attract trips produced by each zone *i*. Everything else being equal, more trips will be attracted by zones that have higher levels of attractiveness. However, other intervening factors affect the choice of *j* as well. Consider, for example, the case of two identical shopping centers (i.e., of equal attractiveness) competing for the shopping trips produced by a given zone *i*. If the distances between zone *i* and each of the two centers are different, shoppers residing in zone *i* will show a preference for the closer of the two identical centers [8]. Thus the intervening difficulty of travel between the producing zone *i* and each of the competing zones *j* has a definite effect on the choice of attraction zone. In the shopping center example above, distance is used as a measure of the difficulty of travel, but other measures of this effect may be used, such as travel time or some generalised cost that includes travel time, out-of-pocket

cost and so on. When applying a specific model for predictive purposes, care must be taken to use the same measure of impedance that was employed to calibrate the model.

Obviously, there are a multitude of reasons why one destination would be chosen over another in the trip distribution analysis. Among these are: better highways between certain locations, lack of jobs in certain zones, dangerous neighbourhoods that must be traversed and so forth. Generally speaking, then, traffic distribution can be considered as a function of: (a) the type and extent of transportation facilities available, (b) the pattern of land use in an area, including the location and intensity of land use, (c) the various social and economic characteristics of the population.

The measure of separation between zones most commonly used for trip distribution is roadway travel time, calculated from the computerised transportation networks. Most transportation planning efforts use peak-period travel times as a measure of zonal separation for home-based work and home-based school models. Recent studies have tried to incorporate travel cost and transit travel time into the separation measure. Cost has been considered in an attempt to estimate effects on trip distribution of parking costs, vehicle operating costs and tolls.

Many mathematical models have been developed to describe and predict the distribution pattern of trips. They are generally divided into two groups: (1) growth-factor methods and (2) theoretically based methods. In the first group, there are four basic types of models. These can be stated as the Detroit, Fratar, the uniform and the average-factor methods. In the second group, the most well known two models are

the gravity model and the intervening opportunities model. These approaches are explained briefly below.

2.5.1 Growth-factor Methods

Growth-factor methods work on the argument that the future number of trips between a pair of zones can be found by proportioning the relative increases (growth) in trip ends in those zones. This proportioning process is iterative in nature. That means, a first proportion is worked out based on initial conditions, new trip end totals are computed, a new proportion established and so on until some stable numbers are obtained. Mathematically, this process is described below.

The initial growth factor for zone i, F_i, is computed by dividing the forecasted trips by actual trip ends:

$$F_i^k = \frac{T_i^*}{t_i^k} \qquad \textbf{2.17}$$

For the whole study area, the trip ends over all zones are summed to get the corresponding area-wide growth factors, F^k.

$$F^k = \frac{\sum_i T_i^*}{\sum_i t_i^k} \qquad \textbf{2.18}$$

Total trip ends in zone i are obtained through:

$$t_i^k = \sum_j t_{ij}^k \qquad i \neq j \qquad \textbf{2.19}$$

On the other hand, the Detroit models are as follows:

$$t_{ij}^{'} = t_{ij} \frac{F_i F_j}{F} \qquad \textbf{2.20}$$

$$t_{ij}^k = t_{ij}^{k-1} \frac{F_i^{k-1} F_j^{k-1}}{F^{k-1}} \qquad \textbf{2.21}$$

where k denotes the k th iteration, T_i^* denotes the predicted trips, t_i^k denotes actual trip ends, F^k denotes corresponding area-wide growth factors.

In these models, the number of trips between zones i and j increases in proportion to the growth of trip ends in the origin zone *(i)* and the growth of trip ends in the destination zone *(j)*.

Another growth-factor method is the Fratar model. This model is often used to estimate external trips, that is, trips that are either produced and/or attracted outside the boundaries of the region under study from outlying areas whose character is not explicitly analysed.

The Fratar model begins with the base-year trip-interchange data. Usually this model does not distinguish between productions and attractions and considers the interzonal trips irrespective of their direction [9]. Since no distinction is made between productions and attractions, the trip generation of each zone is denoted by

T_i. The following trip balance equation provides the necessary relationship between the trip generation of a zone *i* and the trip interchanges that involve zone *i*:

$$T_i = \sum_j T_{ij} \qquad \textbf{2.22}$$

The estimate of the target-year trip generation $T_i(t)$, which precedes the trip-distribution phase, is computed by multiplying the base-year trip generation, $T_i(b)$ by a simple growth factor, namely G_i. This growth factor is based on the anticipated land-use changes that are expected to occur within the zone between the base year and the target year. Thus;

$$T_i(t) = G_i[T_i(b)] \qquad \textbf{2.23}$$

Subsequently, the Fratar model estimates the target-year trip-distribution $T_{ij}(t)$ that satisfies the trip balance (equation 2.22) for that year. Mathematically, the model consists of successive approximations and a test of convergence in an iterative procedure. During each iteration, the target-year trip-interchange volumes are computed based on the anticipated growth of the two zones at either end of each interchange. The implied estimated target-year trip generation of each zone is then computed according to equation 2.22 and compared to the expected target-year trip generation of equation 2.23 [9]. A set of adjustment factors, R_i, are then computed by:

$$R_i = \frac{T_i(t)}{T_i(current)} \qquad \textbf{2.24}$$

If the adjustment factors are all sufficiently close to unity, the trip balance constraint is satisfied and the procedure is terminated. Otherwise the adjustment factors are used along with the current estimate of trip distribution Q_{ij} (current) to improve the approximation. A comparison of equations 2.23 and 2.24 shows that the adjustment factors used in all but the first iteration and the original growth factors applied during the first iteration play the same mathematical role. Their interpretation, however, is not the same: The growth factors constitute a prediction of the actual growth of each zone between the base year and the target year, but the subsequent adjustment factors are merely mathematical adjustments that facilitate the convergence of the solution to the predicted zonal trip generation [9].

The basic equation employed by the Fratar model to calculate the portion of the target-year generation of zone *i* that will interchange with zone *j* is:

$$T_{ij}(new) = \frac{(T_{ij}(current))R_i}{\sum_j (T_{ij}(current))R_j} T_i(t) \qquad \textbf{2.25}$$

This equation is similar to that of the gravity model, which will be presented later in this chapter. The expected trip generation of zone *i* is distributed among all zones so that a specific zone *j* receives its share according to a zone-specific term divided by the sum of these terms for all "competing" zones *j*. When equation 2.25 is applied to all zones, two estimated values result for each pair of zones: The first

represents the portion of the generation of zone *i* chosen to the interchange due to the influence of zone *j* (or T_{ij}), and the second is the portion of the generation of zone *j* chosen to the interchange due to the influence of zone *i* (or T_{ij}).

An asymmetric form of the Fratar model begins with a base-year trip table in the production-attraction format. In this case the sum of each row represents the base-year productions, whereas the sum of each column represents the base-year attractions of the corresponding zone. Each zone is given two growth factors: one associated with the expected growth in residential activity (and therefore productions), whereas the second captures the zone's non-residential growth (i.e., attractions).

Uniform growth factor method can be summarised in a compact form as:

$$E = \frac{\sum T_i^G}{\sum\sum T_{ij}^T} \qquad \textbf{2.26}$$

$$T_{ij}^F = T_{ij}^T \times E \qquad \textbf{2.27}$$

where;

E = Uniform Growth (Adjustment) factor

T_i^G = Trip generation output for future

T_{ij}^T = Total trips today

T_{ij}^F = Flow from *i* to *j* in future

The steps that should be followed are straightforward. First the uniform growth factor will be calculated. Then this factor will be applied to all current flows.

Also the average growth factor method can be presented mathematically as:

$$E_i^{k-1} = \frac{T_i^G}{T_i^{k-1}} = \frac{T_i^G}{\sum T_{ij}^{k-1}} \qquad \textbf{2.28}$$

$$T_{ij}^k = T_{ij}^{k-1} \times \left[\frac{\left(E_i^{k-1} + E_j^{k-1} \right)}{2} \right] \qquad \textbf{2.29}$$

where;

E_i^{k-1} = Average growth (adjustment) factor

T_i^G = Total trip generation at *i* in future date

T_i^{k-1} = Total trips for iteration *k* at *i*

T_{ij}^k = Flow from *i* to *j* for iteration *k* (represents future)

The steps that should be followed to calibrate the model are: (1) run a trip generation model; (2) determine the first estimate of "average growth factors"; (3) apply the first set of average growth factors to all current flows; (4) check for closure.

The Fratar and Detroit models are considered to have better mathematical expressions and to be computationally more efficient than the uniform growth and average growth factor models. In any case, the growth-factor models find most applications in estimating trips from external to internal or other external zones since there are no land-use data available for the external areas outside the study region.

These models are advantageous in that they: (1) are simple, inexpensive and easy to apply; (2) are well-tested; (3) require no distance variables; (4) need no calibration; (5) can be applied to peak directional flows; (6) are useful in updating origin-destination surveys.

However, the disadvantages are: (1) only a single growth factor for each zone, and assumed stable to the horizon year; (2) inability to account adequately for major changes in land use or interzonal activity; (3) no explicit term relating to any form of travel cost, time or other impedances; (4) zones having zero interchanges in the base will show zero interchanges in the horizon year; (5) errors in the original distribution due to sampling or other factors will be carried forward and magnified.

2.5.2 Theoretically Based Models

The gravity model, as its name implies, gets its name from the fact that it is conceptually based on Newton's law of gravitation, which states that the force of attraction between two bodies is directly proportional to the product of the masses of the two bodies and inversely proportional to the square of the distance between them or;

$$F_{12} = G \times \frac{M_1 \times M_2}{{d_{12}}^2} \qquad \textbf{2.30}$$

where;

F_{12} = the gravitational force between bodies 1 and 2

M_1 = mass of body 1

M_2 = mass of body 2

d_{12} = distance between bodies 1 and 2

G = a constant

When analysing this model, travel researchers noted an interesting analogy, especially in regard to shopping travel: M_1 might represent the "mass" of trips available at, say, a residential area; M_2 the "mass" or attractiveness of a shopping area; d_{12} the distance between the two areas; and F_{12} the number of trips between the two areas. These interpretations would imply through the gravity model that the greater the size or attractiveness of the two areas (masses) and the less the distance between them, the more would be the number of interarea trips. This was found to resemble many real world situations [10]. For example, the volume of long-distance telephone calls between cities may be modelled in this manner, with the population sizes of the cities replacing the masses of particles and the distance between cities or the cost of telephone calls taking the place of d_{12} . The application of this concept to trip distribution takes the form which can be expressed as:

$$T_{ij} = k \frac{P_j A_i}{d_{ij}^{\ c}} \qquad \textbf{2.31}$$

where;

T_{ij} = number of trips produced in zone i and attracted to zone j

k = a specific zone-to-zone adjustment factor for taking account the effect on travel patterns of defined social or economic linkages

P_j = number of trips produced by zone j

A_i = number of trips attracted to zone i

d_{ij} = distance between zone i and zone j

c = an empirically determined exponent that expresses the average area-wide effect of spatial separation between zones on trip interchange

Equation 2.31 states that the interchange volume between a trip-producing zone j and a trip-attracting zone i is directly proportional to the magnitude of the trip productions of zone j and the trip-attractiveness of zone i and is inversely proportional to a function of the impedance d_{ij} between the two zones.

In this model, the interzonal volume is the dependent variable; the productions, attractions and impedances are the independent variables; and the constants k and c are the parameters of the model that must be estimated through calibration using base-year data.

However, the parameter k can be eliminated from equation 2.31 by applying the trip-production balance constraint, which states that the sum over all trip-attracting zones i of the interchange volumes that share j as the trip-producing zone must equal the total productions of zone i, or;

$$P_j = \sum_i T_{ij}$$

and

$$A_i = \sum_j T_{ij} \qquad \textbf{2.32}$$

After carrying out some mathematical operations, the classical form of the gravity model can be obtained as:

$$T_{ij} = P_j \left[\frac{A_j / d_{ij}^{\ c}}{\sum_j \left(A_j / d_{ij}^{c} \right)} \right] \qquad \textbf{2.33}$$

The bracketed term is the proportion of the trips produced by zone *i* that will be attracted by zone *j* in competition with all trip-attracting zones. Note that the numerical value of this fraction would not be affected if all attraction terms were multiplied by a constant. This implies that the attraction terms can measure the relative attractiveness of zones. For example, one employment zone may be said to be twice as attractive as another, based on the number of employment opportunities available.

The gravity formula can be written alternatively as:

$$T_{ij} = P_j \left(\frac{A_j F_{ij}}{\sum_j A_j F_{ij}} \right)$$

where;

$$F_{ij} = \frac{1}{d_{ij}^{c}} \qquad \textbf{2.34}$$

where F_{ij} is known as the travel-time (or friction) factor [9].

Finally, a set of interzonal socioeconomic adjustment factors K_{ij} are introduced during calibration to incorporate effects that are not captured by the limited number of independent variables included in the model. The resulting gravity formula becomes:

$$T_{ij} = P_i \frac{A_j F_{ij} K_{ij}}{\sum_{j=1}^{n} A_j F_{ij} K_{ij}} \qquad \textbf{2.35}$$

where;

T_{ij} = number of trips produced in zone i and attracted to zone j

P_i = number of trips produced by zone i

A_j = number of trips attracted to zone j

F_{ij} = travel-time factor

K_{ij} = a specific zone-to-zone adjustment factor

This gravity modelling is known as the unconstrained gravity model where row totals and column totals are not constrained. As this is required, constrained gravity models can be used instead of unconstrained models.

The calibration of the gravity model in the form of equation 2.33 involves the determination of the numerical value of the parameter *c* that fixes the model to the one that replicates the base-year observations. Hence knowledge of the proper value of *c* fixes the relationship between the travel-time factor and the interzonal impedance.

Unlike the calibration of a simple linear regression model where the parameters can be solved by a relatively easy minimisation of the sum of squared deviations, the calibration of the gravity formula is accomplished through an iterative procedure: An initial value of *c* is assumed and equation 2.33 is applied using the known base-year productions, attractiveness and impedances to compute the interzonal volumes T_{ij}. These results are then compared with those observed during the base year. If the computed volumes are sufficiently close to the observed volumes, the current value of *c* is retained as the calibrated value. Otherwise an adjustment to c is made and the procedure is continued until an acceptable degree of convergence is reached. Most commonly, the friction-factor function *F* rather than the parameter *c* is used in the calibration procedure.

In urban transport planning, once the minimum time paths are established from each zone centroid (center) to all others to obtain the current zone to zone travel times (distances, or any other cost element), the gravity model would be calibrated to distribute the trips using the total trip attractions and trip productions of each zone obtained from trip-generation analysis.

In spite of the common use of the gravity model, there are some advantages and disadvantages. Some of the advantages of the gravity model are: (1) it accounts for competition of trips between land uses by emphasising trip attractions versus productions; (2) it is sensitive to changes in travel times between zones, is able to recognise trip purposes as affecting zonal interchanges; (3) it is easy to understand and therefore easy to apply in particular areas [9, 11].

Shortcomings of the gravity model are: (1) it is very unlikely that the travel-time factors by trip purpose would remain constant throughout the urban area to the horizon period; (2) the changing nature of travel times between zones with time of day makes the use of single values for the travel time factors questionable; (3) it tends to overestimate short trips and underestimate long trips; (4) it usually focuses on impedance (or zonal separation) which lacks a behavioural basis explaining the choices made by individuals among alternatives; (5) it does not include variables that reflect the characteristics of the individuals or households who decide which destinations to choose in order to satisfy their activity needs [9,11].

In its original application, the interzonal impedance was measured only in terms of highway travel time which fails the incorporation of the effect of the presence of other modes of travel explicitly. However, this can be eliminated by incorporating additional variables, such as parking and toll charges, in the expression of impedance. The term “generalised cost” is often used in place of impedance to show up this practice. Moreover, measures of composite impedance have been considered that combined travel times and costs associated with all modes providing services between pairs of zones, including transit and, more recently, non-motorised modes. In some instances, the generalised costs of the various modes would be

weighted by the expected proportions of trips that each attracted. As another approach would be the use of composite utility which is computed by certain choice models. Alternative to gravity modelling, a more recent practice has been to abandon in favour of more behaviourally based destination choice models [9].

The use of *K*-factors to adjust for discrepancies between the observed base-year trip-length frequency distribution and that resulting from the use of the final friction factors alone has been an interest for two reasons. The first reason is related to difficulties arising from attempts to interpret the effects captured by the *K*-factors and the second has to do with the question of whether these effects would hold true between the base and target years. The need for the *K*-factors has been explained in detail in [9] as capturing special conditions between some zonal pairs such as the need to cross a river. Other findings showed that *K*-factors were needed to rectify a mismatch between the types of jobs in which residents of producing zones were engaged and the type of employment available in the trip-attracting zones. For example, workers in zone *i* could be sent by the gravity model to jobs in zone *j* because the latter is closer to *i* than a third zone. To minimise this difficulty, some gravity-model applications resort to stratifying jobs by industry and employment type or income at the cost of added computational load. Experience has also shown that the causes of the problem may be rooted in historical and cultural factors that are unique to the local area. A good understanding of local conditions and their likelihood to persist over time can provide invaluable insights that can potentially aid the modeller in interpreting and applying *K*-factors with good judgement [9].

Despite all these shortcomings, the gravity modelling has extensively been employed in most urban transportation planning packages.

The distance variable involves two choices: a physical measure of distance, travel time, or the cost of travel; or a mathematical distance decay function. The best-known distance decay functions are power functions of the form [12]:

$$f(d_{ij}) = d_{ij}^{-\beta} \qquad \textbf{2.36}$$

and exponential functions in the forms of Equation 2.37 and Equation 2.38.

$$f(d_{ij}) = e^{-\beta d_{ij}} \qquad \textbf{2.37}$$

$$f(d_{ij}) = d_{ij} e^{-\beta d_{ij}} \qquad \textbf{2.38}$$

where β is an empirical constant representing the severity of the inhibiting effects of distance (d_{ij}) on trip making. Other things being equal, the higher the value of beta, the faster the fall in the number of trips with distance. The second negative exponential function presented above produces a distance decay that is less rapid as distance increases, allowing for a situation where the number of trips actually increases over short distances, but then decreases thereafter. One disadvantage of the power function form is that when distance is zero, this function equals zero, and the gravity model will predict infinite number of trips [13]. When a study concerns region-to-region, city-to-city or zone-to-zone trips, the intra-region, intra-city or intra-zonal trips are usually excluded from the spatial interaction modelling analysis and the distance for these trips will be zeroes. Applying a power distance decay function or a type of exponential functions to such problems may cause the gravity

model to produce infinite trips in these intra areas. A common practice to deal with the problem is to assign some very large values to these distances so that the inhibiting effects is so big that no trips will be distributed within these areas.

The traditional family of gravity models consists of unconstrained and constrained gravity models including unconstrained, production constrained, attraction constrained and fully constrained gravity models.

These traditional gravity models can be listed as follows:

1) Unconstrained gravity model
2) Production constrained gravity model
3) Destination constrained gravity model
4) Doubly constrained gravity model

Among the above-mentioned models, the fully constrained gravity model is the most widely used one in terms of accuracy and pattern recognition [14]. Other gravity models have some problems such as inconsistency, miss-specification, multi-collinearity or data distortion associated with the transformation of the equations into a linear relationship and operational form [15].

As an alternative to the gravity model, the intervening opportunities model is also available. This model is based on an attractively simple hypothesis that there is a constant probability that a traveller will be satisfied at the next opportunity. The even simpler assumption is that the number of trips from an origin to a destination zone is proportional to the number of opportunities at the destination zone and inversely proportional to the number of intervening opportunities [12].

To compute intervening opportunities, the order of destination zones away from any origin zone should be identified. The number of trip origins at zone *i* (O_i) multiplied by the probability of a trip terminating in *j* is given in the following equation as [16]:

$$T_{ij} = O_i\left(P(V_{j+1}) - P(V_j)\right) \qquad \textbf{2.39}$$

$$T_{ij} = O_i\left(e^{-LV_{j+1}} - e^{-LV_j}\right) \qquad \textbf{2.40}$$

where;

$P(V_j)$ = total probability that a trip will terminate before the *j* th possible destination is considered,

V_j = subtended volume or the possible destinations already considered, that is, reached before reaching zone *j*,

L = constant probability of a possible destination being accepted if it is considered

The above equations 2.39 and 2.40 are the usual statement of the intervening opportunities model.

One of the main differences between intervening opportunities model and the gravity model is that the former has a probabilistic where the latter has a deterministic structure.

CHAPTER 3

BASIS OF ARTIFICIAL NEURAL NETWORKS

3.1 General

A human brain is composed of different regions, where each region is responsible for the accomplishment of complex tasks. It has always been an important area of research to understand the functioning of the brain. In one of the major attempts to understand the brain, the idea of neurons as structural constituents of a brain was introduced [17]. These neurons are organised as a biological network with soma (cell body), dendrites and synapses. The soma is the body of the neuron and dendrites are connections to synapses. It is estimated that there are 10 billion neurons and 60 trillion synapses and dendrites in a human brain [18]. Thus, the brain has a truly amazing number of neurons with massive interconnections between them.

Each neuron is analogous to a logical processing unit where inputs are received and processed as outputs. It is this input and output structure that

constitutes the essence of artificial neural networks. When a number of processing units are interconnected, an artificial neural network is formed. In this dissertation, artificial neural networks and neural networks are used synonymously.

The development of artificial neural networks dates back to the 1940s. One of the first abstract models of a neuron was presented by McCulloch and Pitts [19]. Hebb [20] put forward a learning rule in 1949 that explained the way a network of neurons learned. Some other researches followed this idea through the next two decades. Minsky [21] and Rosenblatt [22] were two of the most important researchers who pursued Hebb's ideas. Rosenblatt is distinguished with his formulation of the perceptron learning algorithm. Widrow and Hoff developed an important variation of perceptron learning during the same time span, which is called the Widrow-Hoff rule [23]. In 1969, Minsky and Papert [24] has put forward the theoretical limitations of single-layer neural network models in their milestone book called "Perceptrons". This book had a pessimistic perspective on neural networks so the research taking place about artificial neural networks fell into an indefinable era for nearly 20 years. In the year 1977, Anderson et al. [25] and in 1980, Grossberg [26] carried out important works on psychological models. In addition, Kohonen [27] developed associative memory models.

In 1980s, a renewal in the neural network approach can be seen. Hopfield [28] initiated the basic idea of energy minimisation with artificial neural networks theory. Later, with his paper, this idea gained great momentum in optimisation applications. The term "connectionist" had been made popular by Feldman and Ballard [29]. Connectionism, sometimes, also named as "subsymbolic processes", that have become the study of informative and artificial intelligence systems stimulated by neural networks [30]. Connectionism lays emphasis on the capability

of learning and discovering representations, unlike symbolic artificial intelligence. In the light of the above-mentioned facts, connectionism has become a common foundation between traditional artificial intelligence and artificial neural networks research.

In the mid 1980s, the book "Parallel Distributed Processing", written by Rumelhart and McClelland [31] produced great impacts on biological, information and computer sciences. Rumelhart, Hinton and Williams [32] developed the back-propagation learning algorithm. This important algorithm suggests a powerful solution for training a neural network. The success of this approach was demonstrated by the NETtalk system, which is developed by Sejnowski and Rosenberg [33]. This system converts English text into highly intelligible speech. It is an interesting note to consider that the idea of back-propagation had been developed by Werbos [34] and Parker [35] independently.

The essence of neural networks stemmed from the basics of a biological neuron. The biological neuron is the basic unit of the brain and is working on a principle of a logical processing unit. The types of neurons are mainly: 1) Interneuron cells, which process locally and have their input and output connections at about 100 microns; 2) Output cells that connect different regions of the brain to each other, namely, connect the brain to muscles or connect other organs to brain [36].

The operation of the neuron is a complicated and not fully understood process, although the basic details are relatively clear. The neuron accepts many inputs, which are all added up in some manner. If enough active inputs are received at once, then the neuron will be activated and transmit a signal; otherwise the neuron will remain in its inactive quiet state.

3.2 Basic Concepts of Artificial Neural Networks

The influence of the synapses, coupled with the incoming signal into the soma (cell body), can be modelled by a linear combination of the inputs to the processing unit. The more influential the synapse, the larger the signal, the less influential the synapse, the smaller the signal.

This basic model, which is analogous to a biological neuron, is shown in Figure 3.1. This model, which is called a perceptron simply performs a weighted sum of inputs (a linear combination), compares this to a threshold value in the processing unit and turns on if this value is exceeded, otherwise it stays off. Since the inputs are passed through the model neuron to produce the output once, the system is known as a feedforward one.

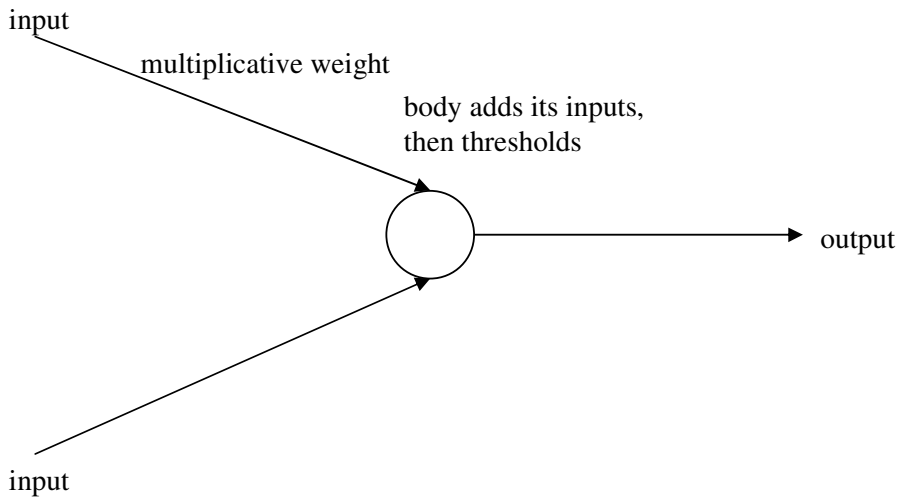

Figure 3.1 Outline of the perceptron model

Notationally, each input X_i is modulated by a weight W_i and the total input is expressed as:

$$\sum_i X_i W_i \qquad \textbf{3.1}$$

This input signal is further processed in the processing unit by an activation function to produce the output. The activation function, which is a threshold function, can be linear, as a straight line, or non-linear, as a sigmoid function.

Generally, a neural network is represented by a set of nodes and arrows, which corresponds to the fundamental concepts in graph theory. A node corresponds to a processing unit, and an arrow corresponds to a connection between those units. As illustrated in Figure 3.2, some nodes are connected to the system inputs and others are connected to the system outputs for information processing. In neural network terminology, this framework is called neural network architecture.

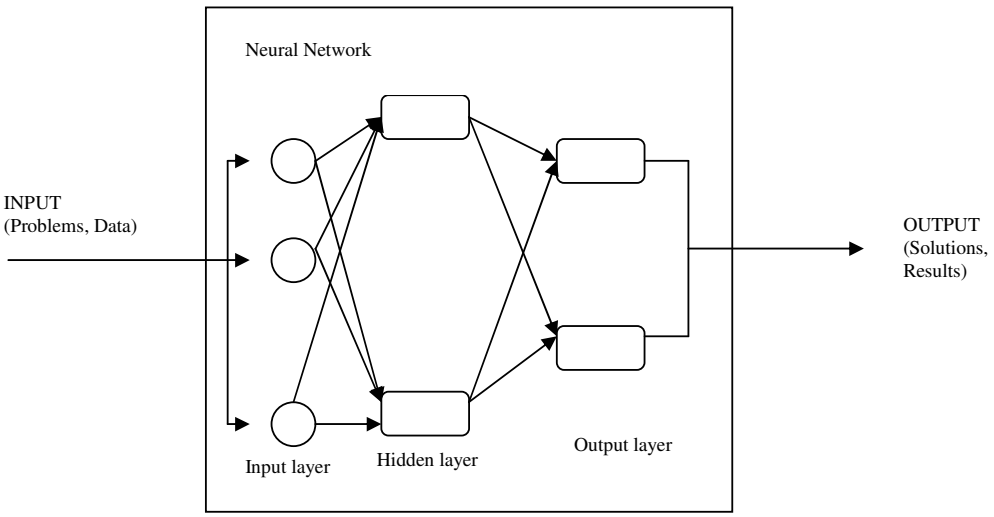

Figure 3.2 Neural information processing as depicted in Fu [40]

The architecture of a neural network refers to its framework as well as its interconnection scheme. The architecture is often specified by a number of layers and a number of nodes per layer. The types of layers are:

a) The input layer is the one that includes input units, which specify the set of input data (instance) presented to the network for processing. For example, each input unit may be designated by an attribute value possessed by the instance. Referring to Figure 3.2, the first row of nodes do not process information. They simply distribute information to other units. Schematically, input units there are drawn as circles as distinguished from processing elements like hidden units.

b) The hidden layer is the one that includes hidden units. These hidden units are processing units and called hidden because they are internal to the network and whose output has no relation with its surroundings.

c) The output layer is the one that includes output units which are associated with possible values to be assigned to the instance under consideration.

According to the connectivity, a network can be either feed-forward or recurrent where the connections can either be symmetrical or asymmetrical. These can be listed as:

a) Feed-forward networks: All connections point in one direction (from the input toward the output layer)

b) Recurrent networks: There are feedback connections or loops

c) Symmetrical connections: If there is a connection pointing from node i to node j, then there is also a connection from node j to node i, and the weights associated with the two connections are equal, or notationally, $W_{ji} = W_{ij}$

d) Asymmetrical connections: If connections are not symmetrical as defined above, then they are asymmetrical.

According to the way the perceptrons are architectured; a neural network can be single layered perceptron as can be seen in Figure 3.1 or multi layer perceptron as in Figure 3.2.

Neural networks attack problems by self-organisation and self-learning. They derive their intelligence from the mutual behaviour of simple computational mechanisms of individual neurons. The computational advantages of neural networks can be summarised as [32]:

a) Knowledge acquisition under noise and uncertainty: neural networks can carry out generalisation, abstraction and extraction of statistical properties from the data.

b) Flexible knowledge representation: neural networks can build up their own representation by using self-organisation techniques.

c) Efficient knowledge processing: neural networks can carry out computations in a parallel manner which is known as parallel-distributed processing.

d) Fault tolerance: through distributed knowledge representation and redundant information encoding, the system's performance degrades slowly in response to errors.

If a formal and universally accepted definition for neural networks is to be given, it can be stated as follows:

"An artificial neural network is a parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use" [35].

A connection between nodes in different layers is called an interlayer connection. A connection between nodes within the same layer is called an intralayer connection. A connection pointing from a node to itself is called a self-connection. In addition, a connection between nodes in distant (nonadjacent) layers

is called a supralayer connection. The term connectivity refers to how nodes are connected. For example, full connectivity often means that every node in one layer is connected to every node in its adjacent layer. A high order connection is a connection that combines inputs from more than one node, often by multiplication. The number of the inputs determines the order of the connection [37].

Connection weights can be real numbers or integers. They can be confined to a range. They are adjustable during network training, but some can be fixed deliberately. When training is completed, all of them should be fixed.

3.3 Types of Artificial Neural Networks

There are various ways of classifying the neural networks. Generally these classifications are based on processing units and interconnection configured, the learning algorithms used or the forms of transmitting information in the network. Any given network, classified as above, can either be a feedforward or a recurrent network.

The information about the inputs propagates through the network in a forward direction from one layer to another until the output information is achieved. Later, the output information is used as the input to the network, which propagates again through the network in the forward direction, which presumably yields better estimates about the output. Whereas in recurrent networks, the processed information, either from the output or the hidden layer is recursively used, that is they are fed back to the network as a part of their inputs.

Whether the specific case is a feedforward or recurrent network, the networks can be either single layer perceptron or multilayer perceptron models. Here, the examples for these types of models are discussed.

• **Adaptive Linear Element (ADALINE)** is a single layer perceptron and feedforward network that accepts several inputs and produces one output.

• **Multiple Adaptive Linear Element (MADALINE)** is a multilayer perceptron and feedforward network, composed of more than one adaptive linear elements.

• **Back-propagation network** is a multi layer perceptron and feedforward network that employs the back-propagation algorithm which uses the gradient descent technique with the error propagated backwards.

• **Hopfield, Kohonen and Adaptive Resonance Networks** are all recurrent and multilayer perceptron models each of which can be used in various disciplines.

• **Hybrid networks** are those composed of certain networks each performing its own function. These models could both be feedforward or recurrent, depending on the type of the network configured. Examples for these can be parallel network models and differentiation models.

All these models can be used as classifiers or function-approximating models.

3.4 Neural Networks As Function Approximators

As is known, a function is a mathematical model where dependent and independent variables are related through some constants. In order to understand the interactions among those variables, the values of the constants should be determined

by calibrating the model. Once the model is calibrated, it should be tested to validate its general applicability.

Dependent variables, independent variables and calibration of the model correspond to input data, output data and training the network in neural network terminology respectively. The trained network should also be tested for its generalisation.

Neural networks that approximate a function exhibit some unique properties. They can learn from input data, recognise new data from previous data and then if properly trained, they can generalise. One unique aspect of a neural network is that no assumptions are made regarding the variables used, whereas they learn and recognise by the way the values are associated with the input data. This advantage is due to the hidden layers that play an important role in training and transforming data. Another unique aspect of a neural network is to handle nonlinear and/or complex functions analysed. For example, any mathematical model selected for a particular problem such as a logistic or a regression model depends very much on the distribution of the data, the relationships among the variables involved and the nature of the problem in question. In contrast, neural networks do not require information about the structural form of the model, since no assumptions are made regarding the distributions of the data. A neural network to approximate a function should first be trained by input and output data. The training of neural networks requires some decisions regarding the number of hidden layers and number of neurons in those layers. There is no any general theoretical procedure to determine the number of layers. However, it has been shown that neural networks with single hidden layers may approximate any functional relationship to a desired level of accuracy [38,39]

whereas, the number of neurons in a hidden layer is usually determined by a trial and error procedure.

In order to train a network, an input set associated with an output set is required. This output set are called target output values. Together with input set, target values specify a pattern. The presentation of all patterns to the network during the training process is called a cycle.

The networks are trained essentially by sequentially altering the connection weights among the neurons after each pattern is presented to the network. This process continues until an error term of the computed output value with the target output value is minimised. This error term is generally the average sum squared error. The networks become better approximators as more patterns are presented to the network during the training process where an error term is minimised. In comparing and contrasting neural networks with traditional function approximating approaches, neural network models have some advantages over them.

To validate the general applicability of a trained network, it should be tested by a set of patterns which are different than those used in the training process. This testing procedure is simply representing a set of patterns to the network and comparing the output values with target values to reach decisions about generalisations of networks. Thus, in this procedure the connection weights among neurons are not altered as in the training process.

A network which is feedforward and multilayer perceptron as mentioned above employs supervised learning in the sense that network is trained on a set of input output pairs and weights are adjusted to minimise the error of outputs and then another set of input output pairs is used to test the effects of training.

In summary, because of the unique properties mentioned above, neural networks generally perform better in approximating a complex function than those traditional techniques such as curve fitting.

3.5 Back-propagation Network Model

A single layered perceptron with an input and output layer classifies a set of inputs by comparing the sum of those weighted inputs with a threshold value. Notationally this can be expressed as:

$$\sum_{i=1}^{n} W_i X_i = \theta \qquad \textbf{3.2}$$

where W_i are the connection weights, X_i are the inputs and θ is the threshold value.

If a threshold is considered as an offset added to the weighted sum, an alternative way of writing above expression is by using a bias neuron which is an extra input to the network with a weight associated. This can be expressed as:

$$\theta_j = \sum_{i=1}^{n} W_i X_i - W_0 \qquad \textbf{3.3}$$

where W_0 is the node weight associated with the bias neuron.

Schematically, such perceptron can be shown as:

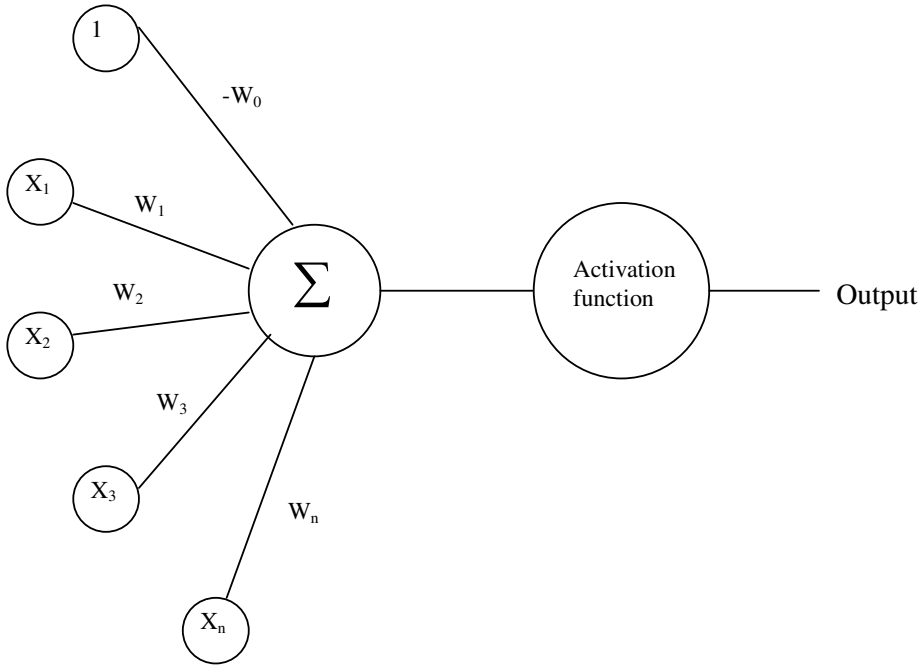

Figure 3.3 Linear perceptron architecture

Single layer perceptrons are generally used to classify linearly separable inputs. When nonlinearly separable conditions exist, it is necessary to use multilayer perceptrons.

Linear separability refers to the case when a linear hyperplane exists to place one class on one side of the plane and those on the other. On the other hand, many classification problems are not linearly separable. To handle this problem, a multilayer perceptron is needed. In contrast to single layer perceptrons, back-propagation neural networks are multi-layered perceptrons that can be used to approximate any complex function.

As in all networks used to approximate functions, a back-propagation model has one input layer, one output layer and one hidden layer. Each layer is fully connected by means of the weights associated with the neurons.

Input values coupled with the weights associated with the neurons in the hidden layer is in fact a linear combination of the input variables. After this linear sum is activated with a threshold function in the hidden layer, the hidden layer outputs are produced. These outputs are again linearly combined with those connection weights between the hidden and output layers. This sum is further activated with a threshold function in the output layer to produce an output to be compared with the target output value.

The sigmoid type of functions (s-curve) are generally used as threshold functions. Using an error term, the initial weights are adjusted with propagating the error between the output and target value backward as in gradient descent technique. That is why this type of multilayer perceptron networks are named as back-propagation networks. Those adjusted weights are again used with the inputs for the next step of the training process. This procedure continues until an acceptable minimum error term is obtained between the output and target values. An architecture of a back-propagation network model is presented in Figure 3.4.

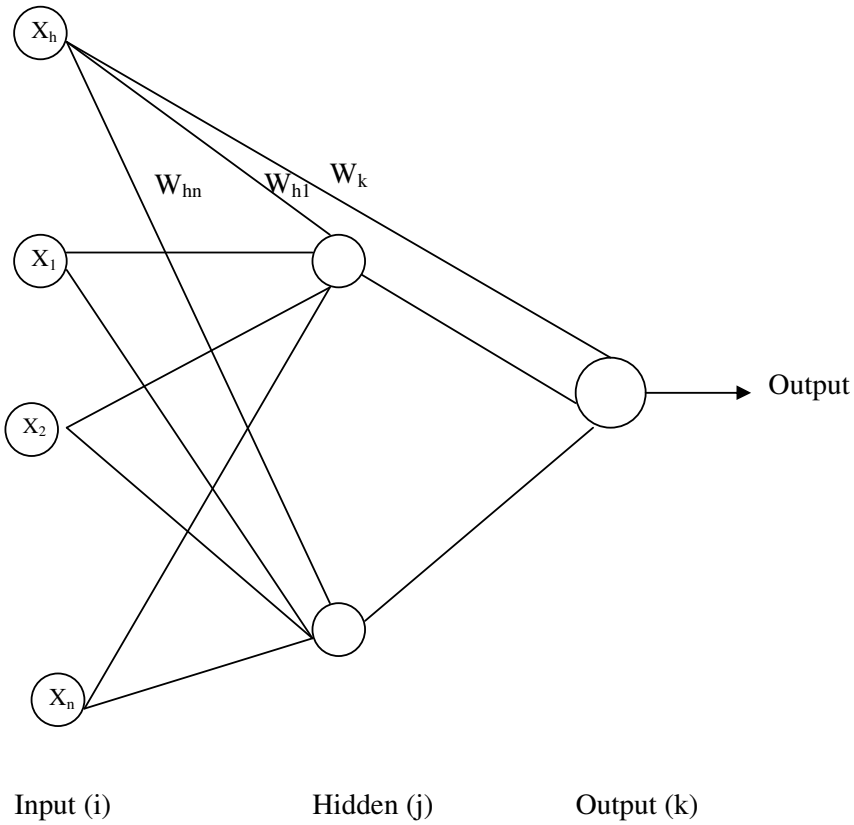

Figure 3.4 Back-propagation architecture

For this model, linearly combined inputs are:

$$y_j = \sum_i^n W_{ij} X_i - W_{hj} \qquad \textbf{3.4}$$

where W_{ij} are the connection weights between input and hidden layers, X_i are the input values, W_h is the bias value which is always equal to one with a negative connection weight.

This y_j value is thresholded with a sigmoid function as:

$$O_j = \frac{1}{1+e^{-y_j}} \qquad \textbf{3.5}$$

where O_j are the outputs from hidden layer.

Linearly combined outputs from the hidden layer is,

$$z_k = \sum_{i=1}^{m} W_{jk} O_j - W_k \qquad \textbf{3.6}$$

where W_{jk} are the connection weights between hidden and output layers, W_k is the bias value which is always equal to one with a negative connection weight.

This z_k value is further thresholded with another sigmoid function as:

$$O_k = \frac{1}{1+e^{-z_k}} \qquad \textbf{3.7}$$

where O_k are the outputs from output layer to be compared with target values.

A general shape of the sigmoid function is shown in Figure 3.5. Since a sigmoid function produces the values between zero and one, it plays a very important role in the back-propagation network.

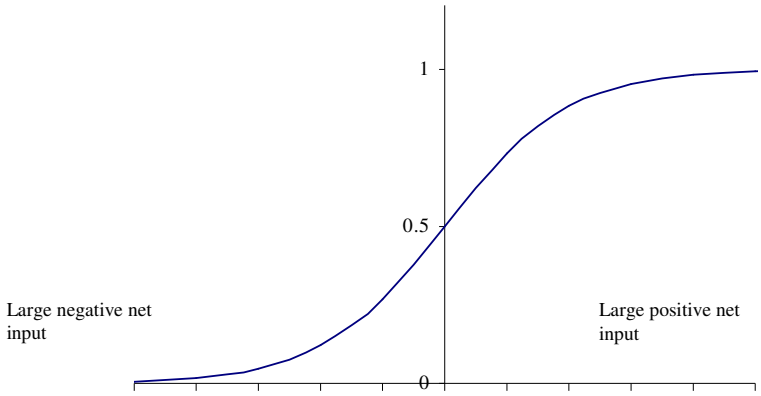

Figure 3.5 A sigmoid transfer function used in back-propagation network

As can be seen from the above figure, after being put through the sigmoid function, the range of the output value is compressed to values between 0 and 1.0. For large negative net input values, the neuron output approaches 0; for large positive values, it approaches 1.0.

A major use of the sigmoid function is that it has a very simple derivative. This fact makes the implementation of the back-propagation algorithm much easier.

If the following sigmoid function is considered,

$$y = \frac{1}{1 + e^{-x}} \qquad \textbf{3.8}$$

its derivative is,

$$\frac{d_y}{d_x} = \frac{e^{-x}}{\left(1+e^{-x}\right)^2}$$

which can be further simplified as;

$$\frac{d_y}{d_x} = y - y^2 \qquad \textbf{3.9}$$

The derivative of the sigmoid function is therefore a simple function of the outputs. Derivative is the slope of the tangent of a function at a specified point mathematically. In addition, this slope is the marginal rate of increase or decrease in outputs. The shape of the first derivative of the sigmoid transfer function in terms of outputs is shown in Figure 3.6. It can be seen that when the sigmoid function approaches zero or one, its derivative reaches a minimum value of zero whereas when it equals to 0.5, its derivative reaches its maximum value of 0.25. This means that the amount of change in a given weight is proportional to the derivative of the sigmoid function.

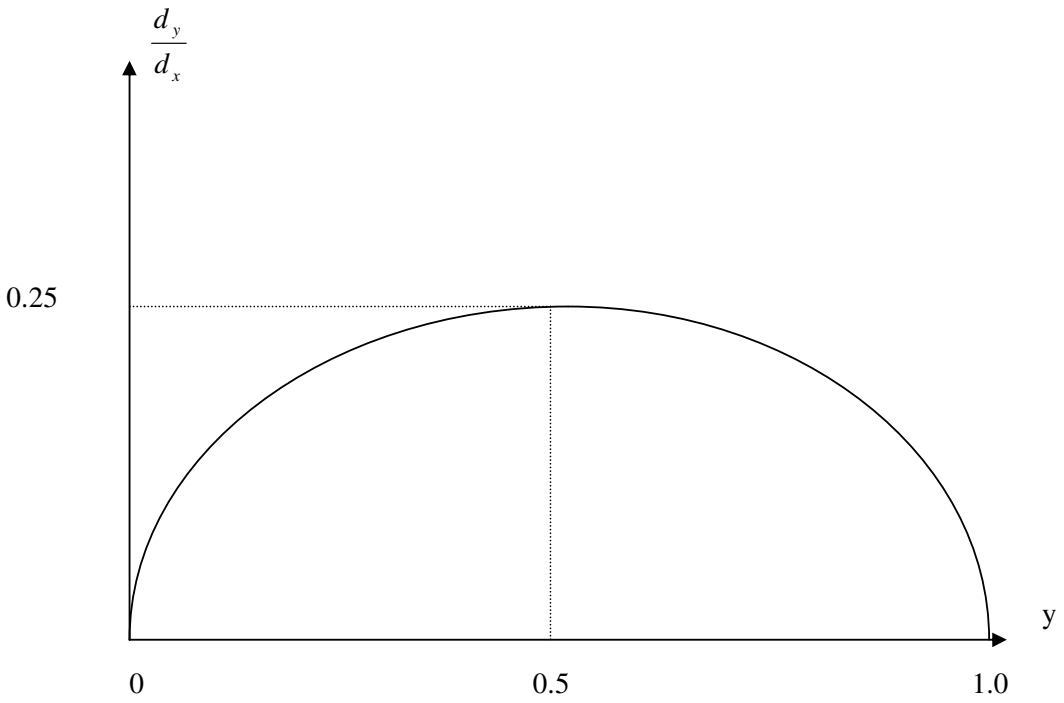

Figure 3.6 The shape of the first derivative of the sigmoid transfer function

The back-propagation algorithm trains the hidden layers by propagating the output error back through the network and adjusting the weights at each layer. Therefore, it is an error-feedback method. An error term is the difference between the value of the output produced by the network and the value of the desired (target) output. Most back-propagation neural networks use the average sum squared error that can be defined as the total sum of squared errors on a neuron-by-neuron basis over the whole set of patterns and divided by the number of patterns.

This error term can be expressed as:

$$E_i = 0.5\sum_{k=1}^{n}\left(T_{ki} - O_{ki}\right)^2 \qquad \textbf{3.10}$$

where T_{ki} is the target output and O_{ki} is the output produced by the network.

The essence of the training process is to minimise the average sum squared error over all training patterns. To minimise the overall mean squared error between desired and target output values for all output units over all input patterns, the weights are iteratively adjusted through a gradient descent algorithm that is referred to as a generalised delta rule. A gradient descent algorithm searches for the solution along the negative of the gradient (i.e. steepest descent). Through a gradient descent, the amount that the weights are to be adjusted for each input is determined by the derivative of the error function with respect to the weights as follows:

$$\Delta W_{ij} = -(CONSTANT)\left(\frac{\partial E}{\partial W_{ij}}\right) \qquad \textbf{3.11}$$

The gradient descent is generally slow and so is the speed of the convergence of the network when the training rate which is the constant term in the above equation is small. However, if a selected training rate is too large, the gradient descent will frequently oscillate widely. This event will slow down the convergence speed and increase the training time. Including another factor as a momentum term will allow for faster training probably with little oscillations. Proper determination of the momentum term will therefore improve the training time while enhancing the stability of the process. The above-mentioned procedure will continue for all the patterns presented to the network.

Since the weighted sum of the inputs are activated through the sigmoid transfer function which produces an output between 0 and 1, all those input and output values should be normalised. Although there is no general theoretical rule for normalisation, the logical approach is to divide them by the largest values in any given pattern.

The formal presentation of the back-propagation algorithm as given in Fu [40] is presented below.

- **Weight Initialisation**

First, the weights and node thresholds will be set to small random numbers. It is important to note that the threshold is the negative of the weight of the bias unit. In addition, the activation level of the bias is fixed to one.

- **Calculation of Activation**

Activation level of input unit is determined by the instance that it is presented to the network. The activation levels O_x's are determined according to the architecture given in Figure 3.4 and corresponding formulae.

- **Weight Training**

The procedure will start at the output units and work backward to the hidden layers recursively. The weights will be adjusted according to the formulation below:

$$W_{ji}(t+1) = W_{ji}(t) + \Delta W_{ji} \qquad \textbf{3.13}$$

where $W_{ji}(t)$ is the weight from unit i to unit j at time t (or tth iteration) and ΔW_{ji} is the weight adjustment.

The change in weight is computed by:

$$\Delta W_{ji} = \eta \delta_j O_i \qquad \textbf{3.14}$$

where η is a trial-independent learning rate ($0 < \eta < 1$). The learning rate η is a constant and it represents a measure of the speed of convergence of the current weight vector to the ideal weight vector. δ_j is the error gradient at unit j.

Convergence is sometimes made faster by including a momentum term as:

$$W_{ji}(t+1) = W_{ji}(t) + \eta \delta_j O_i + \alpha\left(W_{ji}(t) - W_{ji}(t-1)\right) \qquad \textbf{3.15}$$

where α is a momentum factor with the range $0 < \alpha < 1$.

The error gradient is given for the output units by:

$$\delta_j = O_j(1 - O_j)(T_j - O_j) \qquad \textbf{3.16}$$

where T_j is the desired (target) output activation and O_j is the actual output activation at output unit j.

The error gradient is given for the hidden units by:

$$\delta_j = O_j(1 - O_j)\sum_k \delta_k W_{kj} \qquad \textbf{3.17}$$

where δ_k is the error gradient at unit k to which a connection points from the hidden unit j.

The iterations will be repeated until the appearance of convergence in terms of the selected error criterion. Iteration consists of presenting an instance, calculating activations and modifying weights.

As a summary, back-propagation uses the generalised delta rule that is a gradient descent algorithm. The error back-propagation process has three major steps. In the first step, an input vector with associated weights is passed forward to the activation of the network as a whole. The movement is that the weights vector in the first layer connects the input units to the hidden units and the second layer connects the hidden units to the output units in the output layer. The sigmoid transfer function ranges the activation of a unit between zero and one. The first step also generates an error between the output of the network and the desired output (target) values. The second step computes the error factor and propagates this factor successively back through the network (error passed backward). The third step computes the changes for the connection weights and the biases, and updates the weights. The goal of the learning process in the generalised delta rule is to minimise the overall mean squared error for all input patterns by adjusting weights through successive learning iterations. This learning and adjusting process continues until the error is minimised to a satisfactory level so that the network model maps an input vector to an output vector.

CHAPTER 4

DEVELOPMENT OF THE NEURAL TRIP DISTRIBUTION MODEL (NETDIM)

4.1 Neural Network Background in Transportation Engineering

During the past decade, numerous studies have been conducted in transportation engineering using neural networks. These studies are documented in tabular form and presented in the Appendix A. In these tables, the application fields, the authors, a brief description about the studies, type of the network used in the study, the comparison model and finally the results that have been concluded are all briefly discussed. As can be seen from the tables, although there are various works related with the spatial interaction modelling, traffic control and freeway flows, there are few studies about neural network applications to trip distribution. These are, namely, "Spatial Interaction Modelling Using Artificial Networks" by Black [14], "A Neural Network Approach to Modelling and Predicting Intercity Passenger

Flows" by Ji-Rong Xie [12] and "Trip Distribution Forecasting with Multilayer Perceptron Neural Networks: A Critical Evaluation" by Mozolin, et al [41].

In the work by Black [14], an artificial neural network using the traditional unconstrained gravity model components is proposed as an alternative to the fully constrained gravity model. Traditional gravity model components are defined as trips produced, trips attracted and distances between production and attraction centers. The gravity artificial neural network model proposed by this author is basically a back-propagation network with three layers. The input and hidden layers contain three neurons whereas the output layer has one neuron as well as bias neurons attached to the hidden and output neurons. Such a network architecture (structure) has 16 connection weights to be estimated during the training process.

To compare the proposed model with the constrained and unconstrained gravity models, the data of commodity flows between nine regions were used. Inputs of the two models were flow production, flow attraction and distances between the nine flow regions. Input to the proposed model included regional flow production, flow attraction and the interregional distance between the origin and destination region. All input data normalised to ranges 0 and 1 by using total production and attraction flows (row and column totals of the flow matrix) and the longest distance as normalised factors. Normalised input values are fed to the proposed network resulted outputs as normalised flows and error is minimised by the back-propagation algorithm.

By comparing root mean squared errors (RMSE) obtained from the models, Black concluded that the application of the proposed model yields errors less than 30 % to 50 %. Based on this comparison, he argued that the errors of the proposed

model are as much as 50 % less than those general models. He further stated that the accuracy of the modelling improved as it is moved from the unconstrained gravity model to the fully constrained gravity model to the proposed model.

The work by Ji-Rong Xie is an extension of the work by Black in the sense that, using the similar neural network architecture, the predictive abilities of the neural and classical models were evaluated. She used the same normalisation procedure for the inputs like Black. The flows were assigned to a partial railroad network, flow maps were produced, and the assigned flows and the generated flow maps are further evaluated through statistical analyses.

In her dissertation, the actual Amtrak passenger flow data were used to predict and analyse regional Amtrak passenger flows and flow patterns. She stated that relatively little research effort has been directed toward Amtrak passenger flows though a great number of items in the literature were devoted to the analysis of region to region or city-to-city travel flows of people and goods by highways and by air. The data set, she used, included all zero flow values. She argued that the most studies and research efforts on traffic flow modelling and forecasting excluded the zero values in the diagonal cells for intra-city or intra-regional flows and those in the off-diagonal cells for inter-city or inter-regional flows. She further argued that this might be due to the inclusion of zero values in modelling are of little interest, whereas these values should also be predicted as well, and it would be interesting to compare the predictive capabilities of different models using the data set with these values.

In this study, a back-propagation neural network model method with the gradient descent search algorithm was used to predict monthly inter-city Amtrak

passenger flows between a sample of stations in order to evaluate the model's predictive ability. A gravity model-based regression model, a gravity model-based log-normal regression model and a fully-constrained gravity model were used for comparison with the neural network model as well. The predicted passenger trips were assigned to the railroad network and flow maps were generated for further analyses of the network flow patterns and the link flow volumes. The relative order of importance of all variables in the neural network model was also studied. The temporal stability of the neural network model was addressed by cross validating twelve months of Amtrak passenger flow data. Her findings were based on the neural network trained with a sample size of 3104 cases and tested with a data set of 97x97 cases, and the resulting root mean squared errors were compared with the other three models.

In her work, she summarised the findings as given below:

- When compared with the gravity model, the neural network ranked second in terms of the lowest root mean squared error, and it predicted only 78.6 % of the actual flows. However, the neural network outperformed the gravity model by accurately predicting 98.7 % of the actual origin-destination pairs with zero flows and by producing smaller root mean squared error with origin-destination pairs of small flows. These origin-destination pairs consisted of 9144 cases out of the 9409 cases. The neural network model missed many origin-destination pairs with small flows, but had a tendency to predict large flows well. The group root mean squared error and the percent root mean squared error for the volume groups of 100-800 and over were slightly smaller for the gravity model than for the neural network model. The gravity model overestimated many actual zero values and small values, and

underestimated the large flows slightly. Given that the data were not normally distributed and the relationships were nonlinear, the regression methods did not fit well.

- The flow assignment using the actual origin-destination flows was used in comparison with the flow assignments using the predicted origin-destination passenger flows from the four models under study. The flow assignment using the neural networks estimates gave up the best results on all counts including the total flows assigned, the total passenger miles travelled, and the average link flows assigned. The assignment using the gravity model estimated flows yielded the second best results, followed by the assignments using the regression estimates and log-normal regression estimates. By using the regression estimates, a great over-assignment of the total flows and total passenger miles travelled is done. On the other hand, using the log-normal regression estimates makes a great under-assignment of the total flows and total passenger miles travelled. These indicate that all predictive models tended to predict the origin-destination trips of longer length. This is especially the case for the neural network model.

- Further analyses were converged on using regression methods with the output of the assignments stated above. The regression using the assigned link volumes based on the neural network predictive results yielded the best results compared with the results of other three regressions based on the estimated model flows. The regression based on the neural network estimated flows yielded the lowest standard error of estimate, the highest R^2 of 0.977, and the mean link flow 10967 that is closest to the true mean link flows. The regression using the assigned link volumes based on the fully constrained gravity model predictive results, yielded

the second lowest standard error of estimate, the second highest R^2, the second best mean link flows and the smallest intercept value. Taken as a whole, statistics indicate that the neural network yielded the best assignment results, the gravity model was second best, the regression model ranked third and the log-normal regression model was last.

- The projecting ability of the models was also considered through an analysis of the predictive results with the link volume groups. The gravity model predicted flows yielded the best assignment results on link flow volumes between 20000 and 80000, the neural network yielded the second best results for this volume group; both models tended to overestimate and as a result over-assigned the flows for this link group. For the links with flow volumes between 10000 and 19999, and the volumes between 1 and 5000, the neural network predicted flows yielded the best assignment results. For the link volumes between 5000 and 9999, the results of the assignments are indefinite between the gravity and neural network predicted flows. For all link volume groups, the assignments using regression and log-normal regression predicted flows were worse than the assignments using the estimated flows from the gravity or neural network model.

- It was found that the input variable attraction was coupled with the largest weights (in absolute value); production was associated with the second largest weights and distance ranked the third. All weights coupled with the bias were large. Larger output weight magnitude may be due to several reasons. The one that was most evident with this study was the larger input variable values. Further, the magnitude of the weights related with the order of importance of the neuron. The variable associated with the largest weights added up the most to the model.

- All possible neuron combinations displayed that for simple single neuron models, the order of importance was attraction, production and distance. This order of importance was conserved in the multivariate models with an arrangement of different neurons. The typical gravity type model with three neurons of classical variables: production, attraction and distance created the best result among all other single-neuron or multivariate models in this study. The relative importance of the variables in multivariate regression models ranked approximately similar to the relative importance order determined by the approach of all possible combinations of neurons for the neural networks. However, it is difficult to explain exactly or naturally the minor differences between the above two rank orders. This is simply because the regression method is not a perfect alternative for all possible combinations approach.

- Temporal stability showed that the neural networks were generally as stable as the regression methods with temporal data, but had an intention to overestimate the flows. The predictions of Amtrak passenger flows for twelve months by both models followed the data and the seasonal patterns very well. While both models were almost equally stable with temporal data, the neural network model outperformed the regression model by producing a smaller root mean squared error and flow estimates of more accurate values for each of twelve months. The root mean squared errors from the neural network model were improved by 38% to 51% over the root mean squared errors from the regression model and the average root mean squared error over the twelve months was improved by 44%. In summary, the neural network model in this study achieved a level of temporal stability and

generalisation analogous to the regression model but with a much a higher correctness of predictive ability than the regression model.

In conclusion, she stated that the application of the neural networks to large data sets produced satisfactory performance results, the neural network greatly outperformed the regression methods by yielding much smaller errors and more accurate predictions given the advantage of no distributional or data requirement, the total root mean squared error from the interaction modelling by the neural network model was the second best compared with the total root mean squared error from the fully-constrained gravity model and the neural network model outperformed the fully-constrained gravity model in terms of root mean squared error for some volume groups.

M. Mozolin et al. researched the performance of multilayer perceptron neural networks and doubly-constrained gravity models for commuter trip distribution. They stated that a number of modelling approaches had been developed to distribute trips, freight or information among origins and destinations but one of the more successful one was the spatial interaction or gravity modelling where the matrix of flows were related to a matrix of interzonal impedances.

Since several studies [14, 42, and 43] have proposed the neural network architecture as a means to model the complexity of spatial interaction, the authors aimed in this study to compare the performance of a perceptron neural network spatial interaction model with that constrained gravity model. This comparison was conducted empirically on journey-to-work patterns in the Atlanta metropolitan area. The authors found the approach as differing significantly from others in several respects. These differences are:

- First of all, they consider the models in a predictive mode. In other words, calibration was executed on observed, base-year data, while testing was performed on data for the projection year.
- Second, the authors baseline model was a doubly-constrained gravity model. This differs from Fischer and Gopal [42] who chose the less accurate unconstrained spatial interaction model, and estimated the parameters by ordinary least squares regression which is considered less precise.
- Finally, they applied an adjustment factor to flows predicted by the neural network output to satisfy production and attraction constraints, and thus made it possible to unambiguously interpret any inconsistency with flows predicted by the baseline doubly-constrained model in terms of relative performance of the models.

In this research, network weights were adjusted only after all examples in the training set had been processed. The algorithm which was used provided quick convergence so the speed of neural network training was increased significantly. With this algorithm, the network mapped the function that best fitted the relationship between dependent variables (production, attraction and travel impedance) and the independent variables (flows). The mapping function was not bounded to either a power or exponential functional form as in the traditional models.

Networks with 5, 20 and 50 hidden nodes were tested in this study. Networks of larger sizes had not been used due to the excessive computational requirement of their training. Each network configuration was processed five times, each run starting with a random set of initial weights and a training set drawn randomly from the full data set.

The neural network model was further identified as follows. Since, in most instances, weights were changed according to the specific learning rule stated above, no momentum term was needed and the learning rate must be specified only for use with the gradient descent method. A 0.1 learning rate was used all throughout the analysis. Experiments with different rates led to highly similar weight estimates and learning speeds. Initial weights were randomly drawn from a uniform distribution within the range of [-0.01, +0.01].

All three network inputs were scaled by dividing the value observed for each example by the input's maximum value in the set. Whereas input scaling was optional, scaling of the output was required for successful learning. Scaling to fit the output within the [0.1, 0.9] range was usually utilised. However, because the networks were tested on data other than those used for training and validation, and that total flows had increased between base and prediction years, the interval was scaled to 0.75.

All networks were trained for a maximum of 100000 iterations. The authors trained and verified all networks on the 1980 data, while the testing was conducted on the 1990 data. Thus, the origin and destination totals found out by summing the flows predicted by the model are usually not equal to the actual origin and destination totals.

At the county level, a total of 225 data vectors were obtained. For each network processed, the training set was formed by randomly selecting 112 vectors without replacement, while the remaining 113 vectors were used for validation. In one experiment, the full set of vectors was used both for training and validation. The network weights that minimise validation error served to test the model on the 400

interactions from the 1990 trip matrix. At the census-tract level, the training set was selected by simple random sample without replacement of 200 examples from the 121104 origin-destination pairs in the 1980 tract-to-tract trip matrix. This was also true for the validation set. The optimal set of network weights was then tested on all 257049 vectors from the 1990 tract-level trip matrix.

This study compared the achievement of multilayer perceptron neural networks and doubly-constrained models for commuter trip distribution. Although Black [14] and Fischer and Gopal [42] noted that a neural network may perform well enough to estimate actual spatial interaction flows by using iterative proportional fitting procedure, M.Mozolin et al. emphasised that the neural network models might fit the data better but their predictive accuracy was poor in comparison to that of doubly-constrained models. They further claimed that their experiments produced evidence that the predictive accuracy of neural network spatial interaction models was inferior to that of doubly-constrained models with an exponential function of distance decay. They pointed to the several likely causes of neural network under-performance, as including "model non-transferability, insufficient ability to generalise and reliance on sigmoid activation functions". Future research was also proposed to use other perceptron formulations (i.e. spatial structure as neural network input) and other neural networks (radial basis functions, for instance) to predict spatial interaction flows with high level of accuracy.

In conclusion, they stated that while neural networks may perform better than conventional models in modelling spatial interaction for the base year, they fail to outperform doubly-constrained model for predicting purposes and current perceptron

neural networks do not provide an appropriate modelling approach to predict trip distribution over a planning horizon.

4.2 Development of Neural Trip Distribution Model

In his work, Black concluded that the accuracy of modelling is improved as one moves from the unconstrained to the fully constrained and to neural network gravity models. This conclusion seems limited within the context of the size of data. It can be stated from the conclusions drawn in the latter two studies that gravity modelling outperforms the back-propagation neural network model.

As discussed previously, a back-propagation neural network, which is a multilayer perceptron network, consists of one input layer, one hidden layer and one output layer. Linearly combined weights from input layer are thresholded with a sigmoid in the hidden layer and information received from the hidden layer is further thresholded with another sigmoid function. Sigmoid output nodes tend to generate S-shaped surfaces.

Due to the nature of sigmoid functions, over this S-shaped surface, the slope of the function for the two inputs (variables) all other things being equal (ceteris paribus) start from zero, reaches a constant and then decreases to zero again as can be seen from Figure 2.5 of the second chapter of this dissertation. In other words, the marginal rate of increase in one of the variables ranges between zero values. This contradicts the theory of spatial interaction phenomena. For example, marginal changes in flow decreases as distance increases. That is, it would not be expected that the marginal changes in flow would start from zero, then will reach to a constant

value in a gradual manner, then will decrease again to zero. Thus, it seems appropriate not to threshold the information received from the hidden layer with a transfer function in the output layer. This type of approach would probably eliminate the underperformance of neural networks used for distribution of trips.

4.3 Modular Neural Networks

One category of neural networks where sigmoid activation functions are not used in the output layer is modular neural networks developed by Jacobs et al [46]. In such networks, linear combination of the outputs from the hidden layer is not thresholded with any functions in the output layer.

The modular neural networks developed by Jacobs et al. [46] are composed of multiple different expert networks and a gating network that decides which of the experts should be invoked for each training case. This approach can be useful in approximating discontinuous function. For example, if the data suggest a discontinuity in the function being approximated, then it would be more accurate to use a separate model on each side of the discontinuity than to fit a single model across the discontinuity. The role of the gating network is to divide the training data set into different subsets (or more generally, to divide the input space into different subspaces), each best dealt with by a distinct expert network.

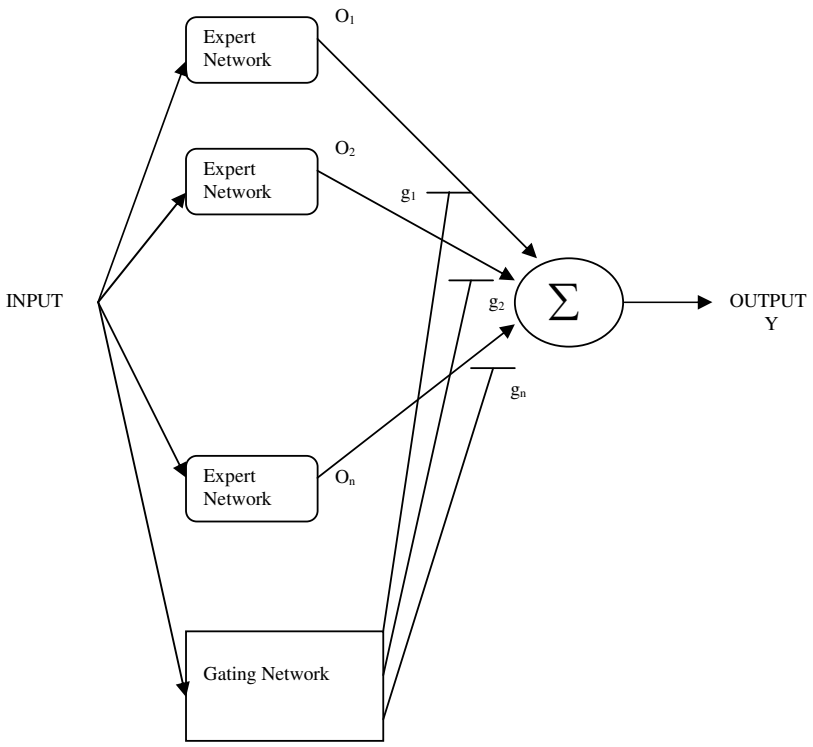

Figure 4.1 A system architecture of modular neural networks (Jacobs et al.) [46]

Each expert network is a feedforward network (a perceptron or a radial basis function network), and all experts have the same input and the same number of outputs. The gating network is also feedforward, typically receives the same input as the expert networks, and has the number of outputs equal to that of the expert networks. The overall output of the system Y is given by:

$$Y = \sum_i g_i O_i \qquad \textbf{4.1}$$

where O_i is the output vector of expert i, and g_i is the corresponding gating strength produced by output unit i of the gating network:

$$g_i = \exp(X_i) / \sum_j \exp(X_j) \qquad \textbf{4.2}$$

where X_i is the total weighted input received by output unit i of the gating network. Since,

$$\sum_i g_i = 1 \qquad \textbf{4.3}$$

then, g_i can be regarded as the probability of selecting expert i. Note that g_i is produced in each inference cycle in contrast to other weights, which are fixed during inference.

As long as the system behaviour is described by a set of continuous functions, it can be trained by gradient descent. The overall output error is propagated backward to the expert and gating networks. The weights inside both networks can be adjusted accordingly.

Multiple expert networks may cooperate or compete to solve a particular problem. Their interaction manner is very much influenced by the following error criterion:

$$E = \left\| T - \sum_i g_i O_i \right\|^2 \qquad \textbf{4.4}$$

where T is the desired output vector, O_i is the actual output vector from the i th expert network, and g_i is the gating strength representing the proportional contribution of expert i to the combined output vector. The strong combination between the experts due to the linear combination of their outputs causes them to cooperate and tends to produce solutions in which many experts are used for each case.

In modular neural network architecture, which can be seen in Figure 4.1, the gating network is used to determine the proportional contributions of the inputs to the output.

4.4 Neural Trip Distribution Model (NETDIM)

An alternative formulation to the modular network is a one where the outputs from hidden layer can be expressed as linear combination of inputs from the input layer. In other words, the input layer is connected to the outputs (activation levels) of hidden layer as well as to the hidden layer. The architecture of this proposed approach is presented in Figure 4.2.

As common to neural networks, there is a bias neuron connected to neurons in the hidden layer, but in contrast, this bias neuron is not connected to the output neuron in the output layer. Needless to say, this bias neuron has a value of 1 whose connection weight is negative as shown in Figure 4.2.

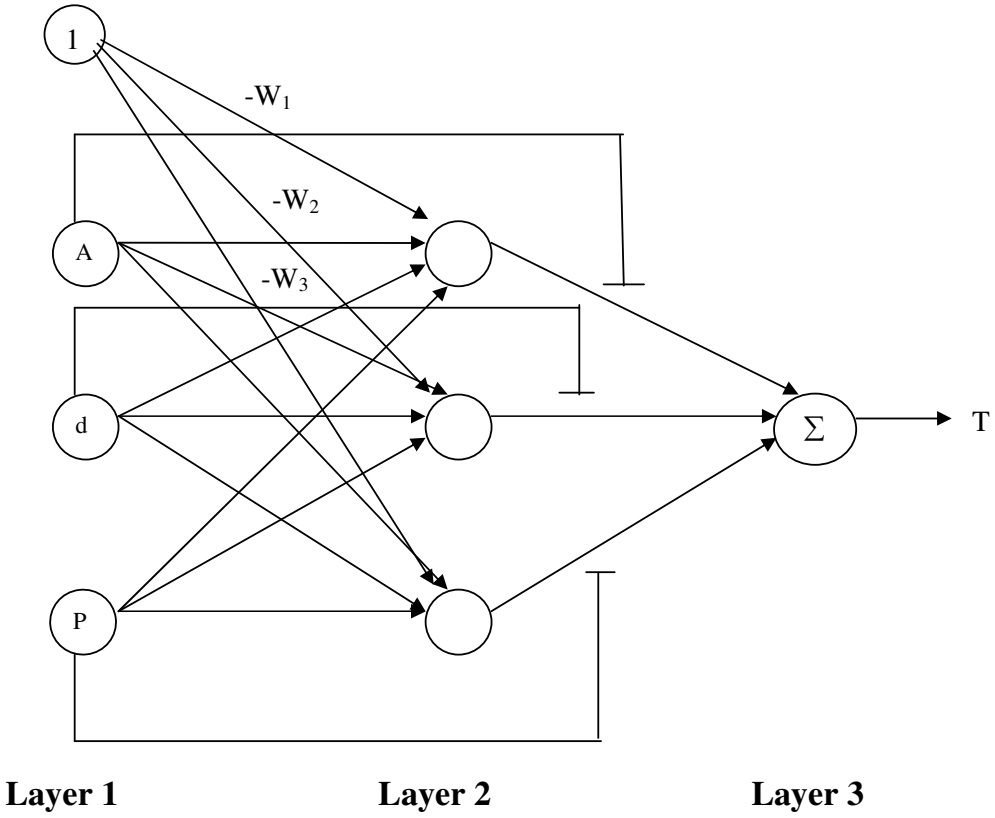

Figure 4.2 Schematical representation of the recommended NETDIM

The connection of the input layer to the activation levels of the hidden layer corresponds to the replacement of connection weights between hidden layer and output layer with the input variables. This implies that the outputs of hidden layer has relation with the external environment that is the input layer and the output layer. Since this is the case, the hidden layer here is termed Layer 2 whereas Layer 1 and Layer 3 are those, the input and output layers, respectively. This is a significant deviation from most of the neural network architectures. As connection weights

between the second layer and the third layer are replaced by the input variables, then the only weights to be optimised are those connection weights between the first layer and the second layer. The linearly combined inputs are thresholded in Layer 2 through a sigmoid transfer function in the nodes of the second layer yielding a single set of activation levels. These activation levels are multiplied by their associated input variables which are further combined linearly in the output layer. Thus the output neuron acts as a summation unit rather than a thresholding unit.

When the above architecture is used in distributing the trips, Layer 1 receives input variables A (Attractions), P (Productions), d (distances) which are the traditional components of gravity modelling. Layer 2 activates the linear combination of those input variables and Layer 3 yields the zonal trip as an output.

The mathematical description of the recommended approach is stated below.

The input variables are linearly combined as,

$$y_j = AW_{1j} + dW_{2j} + PW_{3j} - W_j \qquad \forall j \text{ node in Layer 2} \qquad \textbf{4.5}$$

This y_j value is thresholded with a sigmoid function in the Layer 2 resulting activation levels as outputs,

$$O_j = \frac{1}{1+e^{-y_j}} \qquad \forall j \text{ node in layer 2} \qquad \textbf{4.6}$$

These outputs are further combined linearly with the input variables yielding a zonal trip as output:

$$T_k = AO_1 + dO_2 + PO_3 \qquad \textbf{4.7}$$

The connection weights between input and second layer can be optimised with the following error function:

$$E = [T - T_k]^2 \qquad \textbf{4.8}$$

where;

E = error function

T = target value (actual output)

T_k = calculated value

The gradient descent that is used in optimising the weights is:

$$\Delta W = -\eta\left(\frac{\partial E}{\partial W}\right) \qquad \textbf{4.9}$$

where;

η = trial-independent learning rate ($0 < \eta < 1$)

$\frac{\partial E}{\partial W}$ = derivative of the error function with respect to the connection weights

CHAPTER 5

ANALYSIS OF VARIOUS APPROACHES USING THE DIFFERENT SIZED NETWORKS

In this chapter, the predictive capabilities of the recommended neural trip distribution model (NETDIM) as well as back-propagation neural network, modular neural network and unconstrained gravity model are discussed. The basis of this discussion is to demonstrate the prediction performance of these models by comparing their levels of prediction rather than to illustrate how well the models predict a given set of data. The prediction performances are compared with the Root Mean Squared Error (RMSE) values. The lesser the Root Mean Squared Error, the better the estimates are. RMSE values can be obtained by the following standard formula:

$$RMSE = \sqrt{\frac{\sum_{j=1}^{N}\left(X_j - \bar{X}\right)^2}{N}} \qquad \textbf{5.1}$$

where;

N = number of observations,

X_j = Predicted values, and

$\bar{X}$ = Observed values

In order to demonstrate the prediction performance of these models, three computer programs are developed. Two of these programs, used for neural trip distribution model (NETDIM) and modular approach are written in FORTRAN, and a program used for the unconstrained gravity model is developed in the source code of Matlab software. In addition, the Neural Network Toolbox of Matlab is used for the same purpose for the back-propagation neural networks.

5.1 Data Acquisition and Generation of Different Sized Networks

In order to demonstrate the predictive capability of the recommended NETDIM, it is compared with the back-propagation, modular and gravity models using different sized networks. For this purpose, thirty node-sized network is chosen and the data sets regarding the zonal trips and distances are taken from the Bursa Transportation Master Plan [47] which is shown in Table 1 and Table 2 in Appendix B. Each data set corresponds upper most left-hand portion of the original full set of

flow and distance matrices in that Plan. As there would always be problems common to all types of data such as symmetry of matrices, these problems are neither examined nor addressed in this work. The data sets are purely used for hypothetical purposes.

A thirty node-sized network is found sufficient in generating different sized networks to illustrate the salient features of those compared models. The networks with 3, 6, 9, 12, 15, 18, 21 and 24 nodes are generated from that thirty node sized largest network. This generation is simply done by assigning the upper most left hand and lower most right portions of the flow and distance matrices of that large network to those smaller ones. Thus, each any sized network is associated with two sets of data. The ones that correspond to the upper most left hand portions are used for training (i.e. calibrating) and those correspond to the lower most right portions are used for testing the models. This is shown schematically as in the Figure 5.1.

The generated small networks are in fact the sub networks of that large one. In the data sets associated with those sub networks, the row and column totals of zonal trips are the trips produced and trips attracted, respectively. The trip productions and attractions together with the distances are used as inputs whereas zonal trips are used as outputs for both training and testing the models.

The generation of the sub networks from that large one has nothing to do with the computational requirements of the models. This is solely done to illustrate the predictive performances of the models regarding the various sized networks. This is a significant departure from those previous works discussed in Chapter 3 in which one type of network with a specified size is used in both training and testing procedures.

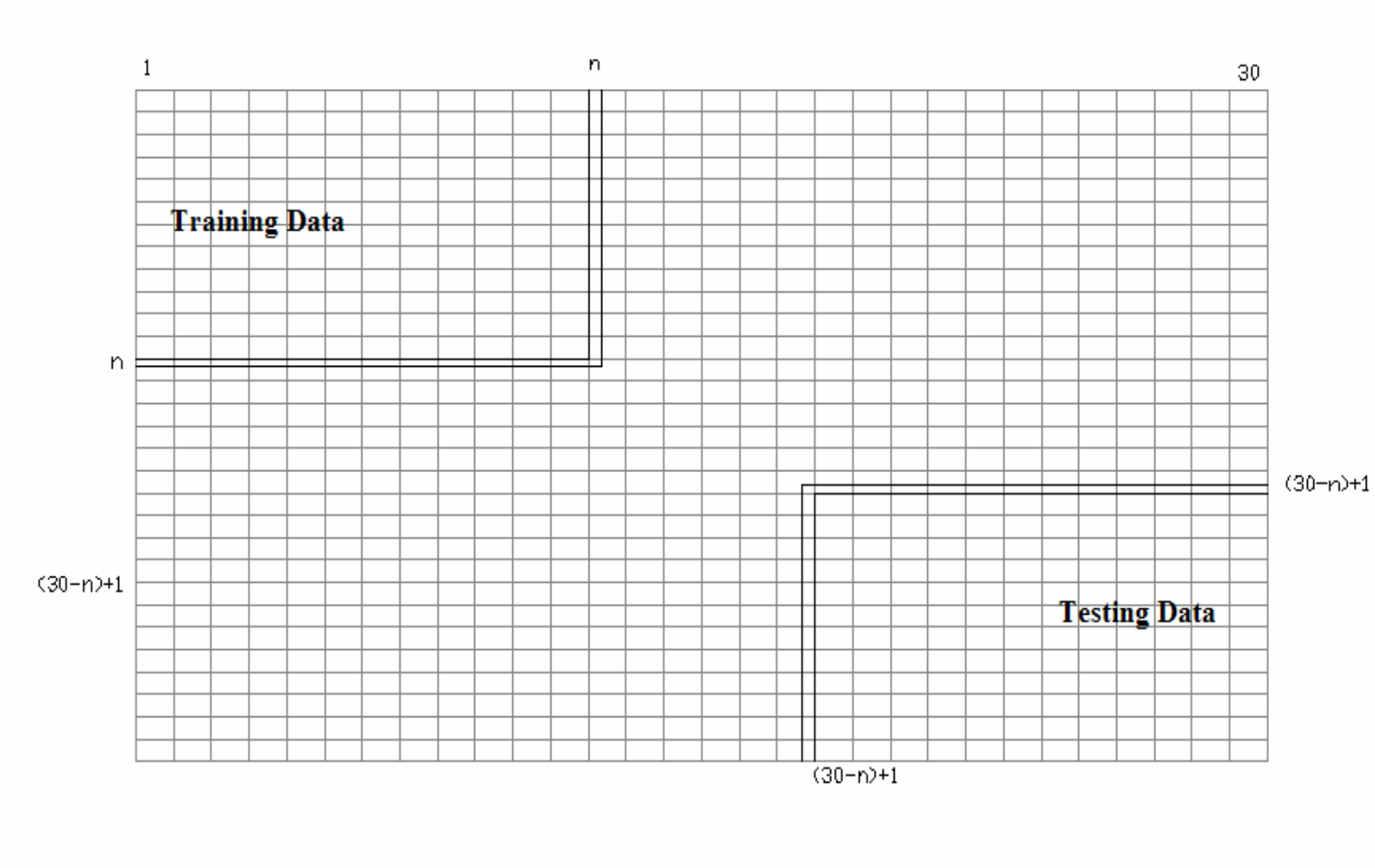

Figure 5.1 Training and Testing the Models

5.2 Training and Testing the Models

For the clarity of discussion, the term neural network is hereafter called a neural model to distinguish this type of network from a transport flow network.

In this section, the training and testing phases for back-propagation neural model approach, NETDIM, modular neural approach and gravity models are discussed.

5.2.1 Training and Testing of Back-Propagation Neural Models

For training the back-propagation neural models, the inputs and outputs are normalised by using the longest distance and the total flow as normalising factors. The training and testing processes are performed by using the Neural Network Toolbox of Matlab. The distances and flow values are normalised outside the Matlab environment.

Distances are normalised by dividing the elements of the distance matrix with the element having the maximum value in that matrix. Whereas, the zonal flows as well as the trip productions and trip attractions are normalised by dividing these values with the total flow value. For illustrative purposes, only the normalisation procedure for a three-node sized network is presented below.

Flow data of a three-node sized network corresponding to the upper most left-hand portion of the original full set of flow matrix to be used for training are given below in Table 5.1.

Table 5.1 Flow data of a three-node sized network used in training

	1	**2**	**3**	**Pj**
1	3275	1500	400	**5175**
2	1550	3800	1025	**6375**
3	525	975	2525	**4025**
Ai	**5350**	**6275**	**3950**	**15575**

The above flow values are all divided by the total flow value of 15575 resulting the following normalised flow values depicted in Table 5.2 such as,

Table 5.2 Normalised flow data of a three-node sized network used in training

	1	**2**	**3**	**Pj**
1	0.2103	0.0963	0.0257	**0.3323**
2	0.0995	0.2440	0.0658	**0.4093**
3	0.0337	0.0626	0.1621	**0.2584**
Ai	**0.3435**	**0.4029**	**0.2536**	**1.0000**

Distance data of the same three-node sized network corresponding to the upper most left-hand portion of the original full set of distance matrix which will also be used for training are depicted in Table 5.3 below.

Table 5.3 Distance data of a three-node sized network used in training

	1	**2**	**3**
1	0.26	0.85	0.85
2	0.85	0.19	0.85
3	0.85	0.85	0.20

The distances are divided by the maximum element which is 0.85 resulting the following normalised distance values as,

Table 5.4 Normalised distance data of a three-node sized network used in training

	1	**2**	**3**
1	0.3059	1.0000	1.0000
2	1.0000	0.2235	1.0000
3	1.0000	1.0000	0.2353

For testing the back-propagation model, the flow and distance data used corresponds to the lower most right hand portion of the original full set of flow and distance matrices. These data presented below as flow and distance matrices respectively are,

Table 5.5 Flow data of a three-node sized network used in testing

	28	**29**	**30**	**Pj**
28	18200	1725	900	**20825**
29	1625	625	125	**2375**
30	950	175	450	**1575**
Ai	**20775**	**2525**	**1475**	**24775**

Table 5.6 Distance data of a three-node sized network used in testing

	28	**29**	**30**
28	0.50	2.35	2.42
29	2.35	0.31	2.06
30	2.42	2.06	0.33

The normalised values are found as,

Table 5.7 Normalised flow data of a three-node sized network used in testing

	28	**29**	**30**	**Pj**
28	0.7346	0.0696	0.0363	**0.8406**
29	0.0656	0.0252	0.0050	**0.0959**
30	0.0383	0.0071	0.0182	**0.0636**
Ai	**0.8385**	**0.1019**	**0.0595**	**1.0000**

Table 5.8 Normalised distance data of a three-node sized network used in testing

	28	**29**	**30**
28	0.2066	0.9711	1.0000
29	0.9711	0.1281	0.8512
30	1.0000	0.8512	0.1364

Since, the training of the back-propagation network is performed by Matlab, a brief description about the software and the data input is given below.

First, the network weights and biases are initialised by the Matlab software internally. At this point, the network is ready for training. The network can be trained for function approximation [48]. The training process requires a set of examples of proper network behaviour namely network inputs and targets. These inputs and targets are named as p and t respectively in the input files of Matlab that are used in this dissertation. During training, the weights and biases of the network are iteratively adjusted to minimise the network performance function. The default performance function for networks is the mean square error (MSE). The mean square error is the average squared error between the network outputs and the target outputs [48].

The gradient descent algorithm is used in the training of the back-propagation model. There are two different ways in which the gradient descent algorithm can be implemented: incremental mode and batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode, all of the inputs are applied to the network before the weights are updated.

In the batch mode, the weights and biases of the network are updated only after the entire training set has been applied to the network. The gradients calculated at each training example are added together to determine the change in the weights and biases [48].

The user of the Toolbox has to be very careful when selecting the step size, in other words, the learning rate. This is often necessary to ensure smooth convergence.

It has been shown in some of researches that a large step size may cause the network to become paralysed [40]. When network paralysis occurs, further training does very little for convergence. On the other hand, if the size is too small, convergence can be very slow. Some researchers have suggested to vary the step size adaptively during training [40].

Some techniques have been used to accelerate the convergence of a gradient descent technique like back-propagation. Newton's method uses the information of the second order derivatives. Quasi-Newton methods approximate second-order information with first-order information. Conjugate gradient methods compute a linear combination of the current gradient vector and the previous search direction (momentum) for the current search direction [40].

In this work, the Levenberg-Marquardt training algorithm has been used. Like the quasi-Newton methods, the Levenberg-Marquardt training algorithm was designed to approach second-order-training speed. This algorithm appears to be the fastest method for training-moderate sized neural networks (up to several hundred weights) [48]. Therefore, it served very well for the purposes of this work.

At this stage, it is important to mention about the input data file format. This input format shows how the Toolbox processes the data.

First of all, basic information about the neural model should be supplied to the Toolbox by implementing "newff" command option so that the network selected is a feedforward one. Then, the number of hidden neurons (changing between 10 and 30 for the 8 cases analysed) is presented to the Toolbox. After this, number of hidden layers as 1 is fed into the system. Then the user of the Toolbox has to mention the connections between the input-hidden and hidden-output layers. For all of the cases,

the logarithmic-sigmoid function is used. Finally, the training algorithm is set as Levenberg-Marquardt.

In the table below, for the eight different sized networks, the number of hidden neurons, the number of hidden layers, type of thresholding functions, the training algorithm, the goal and the minimum gradient for training and testing the neural model are presented.

Table 5.9 Neural network training and testing parameters

Network Size	Hidden Neurons	Hidden Layers	Thresholding function	Training Algorithm	Goal	Minimum Gradient
3 node	10	1	logsig	lm	10^{-4}	10^{-10}
6 node	10	1	logsig	lm	10^{-4}	10^{-10}
9 node	10	1	logsig	lm	10^{-4}	10^{-10}
12 node	10	1	logsig	lm	10^{-6}	10^{-10}
15 node	25	1	logsig	lm	10^{-6}	10^{-10}
18 node	25	1	logsig	lm	10^{-6}	10^{-10}
21 node	25	1	logsig	lm	10^{-6}	10^{-11}
24 node	30	1	logsig	lm	10^{-6}	10^{-12}

where logsig represents logarithmic-sigmoid function and lm represents the Levenberg-Marquardt training algorithm.

There are four training parameters associated with the Levenberg-Marquardt algorithm. These are namely as: show, epochs, goal and minimum gradient. There is no learning rate parameter as in the gradient descent algorithm in the Levenberg-Marquardt algorithm. The learning rate is multiplied with the negative of the gradient to determine the changes to the weights and biases. The larger the learning rate, the bigger the step. If the learning rate is made too large, the algorithm becomes unstable. If the learning rate is set too small, the algorithm takes a long time to converge. The Matlab software does the necessary modifications for the learning parameter and momentum factor internally. Therefore, this eliminates the trial and error procedure so the user of the Toolbox gains more flexibility and time [48].

The training status is displayed for every show iteration of the algorithm. The training stops if the number of iterations exceeds epochs. If the performance function, which is mean square error (MSE), drops below goal, the training stops again. In addition, the same criteria will hold if the magnitude of the gradient is less than the minimum gradient [48].

Normalised flow and distance values are fed into the system as input data. Finally, normalised target flow values are also fed to the system. These input parameters are fed into the system with a standard format that is unique to the Toolbox [48].

After the training has finished, the testing phase started with the simulation command option of the Toolbox. Simulated normalised values which are the test results are multiplied with the associated total flow values in order to obtain the predicted flow values. Then the RMSE values are calculated.

5.2.2 Training and Testing of NETDIM and Modular Model Approaches

A similar procedure as discussed previously is performed to normalise all input and output data used to train and test the NETDIM and modular model approaches.

The training of the modular neural model and NETDIM is performed with two different computer programs developed. These programs are essentially ones based on minimising an error function with the gradient descent algorithm and adjusting the connection weights accordingly.

Training is performed incrementally in the sense that the inputs are presented to the neural model in an iterative manner. Each input set is consisting of trips produced, trips attracted and distance as impedance factor is presented to the model, error is minimised and weights are adjusted in that iteration. This iterative procedure continued until all input sets are exhausted. Once all input sets are used, one cycle is completed. The next cycle starts with the adjusted weights in the last cycle and this process continued until a level of accuracy is obtained in the RMSE values.

In the training and testing phases of NETDIM and modular neural models, a desired level of accuracy is reached by employing a momentum factor of 0.1 for all network sizes up to 18 node-sized network. However, a problem related with a local minimum is encountered after this sized network is trained. To avoid this problem, the momentum factor is changed as 0.6 for training and testing the rest of the networks.

As the number of neurons in hidden layer of both NETDIM and modular models are determined by the number of input variables, there is no need to change the number of neurons in the so-called hidden layer of modular model and Layer 2 of NETDIM. Since the number of input variables used in this work is three, the number of neurons is fixed which gives the user a flexibility in carrying out multiple runs of the programs.

Testing of input sets are performed with the trained model whose connection weights are those adjusted weights from the last iteration of the last cycle.

5.2.3 Calibrating and Testing the Gravity Model

The essence of the back-propagation algorithm is based on presenting the patterns successively one after another to the network. This results in zonal trips as output values, which provide any information about the row and column totals. In other words, equal row and column totals cannot be obtained within the environment of the algorithm. Since our aim is to compare the level of predictions of all the models, unconstrained gravity model is chosen rather than constrained gravity model.

The unconstrained gravity model in the form as discussed in Chapter 1 is calibrated for all those sized networks with the algorithm given in Dickey [11]. This algorithm is programmed in the source code of Matlab software. The basis of this algorithm is to calibrate F_{ij} factors using the trips observed and trips calculated for a predetermined time interval. This calibration is performed iteratively until ratio of trips observed to trips calculated are sufficiently close to one.

In this work, the time intervals are obtained by dividing the maximum distance value of a distance matrix associated with a network size with the size of the network. For example, the maximum distance value which is 0.85 in the distance matrix below is divided by 3 which is the size of the network resulting in time intervals as, 0.19-0.41, 0.41-0.63, 0.63-0.85.

Table 5.10 A sample distance matrix

	1	**2**	**3**
1	0.26	0.85	0.85
2	0.85	0.19	0.85
3	0.85	0.85	0.20

5.3 Testing Results for all Models

In the following discussion, the test results (i.e., the predicted flow values) only for a three-node sized network are presented below for the back-propagation neural model, NETDIM, modular and gravity model approaches. The other flow values for different sized networks can be found in Appendix C.

The predicted flow values obtained through the simulation of back-propagation neural networks are given below with the RMSE value as 2436.

Table 5.11 The predicted flow values obtained through the simulation of back-propagation neural networks

	28	**29**	**30**	**Pj**
28	10946	1332	1332	**13610**
29	991	723	139	**1853**
30	991	60	404	**1455**
Ai	**12928**	**2115**	**1875**	**16918**

The predicted flow values obtained through testing of NETDIM network are given below with the RMSE value 2865.

Table 5.12 The predicted flow values obtained through testing of NETDIM network

	28	**29**	**30**	**Pj**
28	25018	2182	1269	**28469**
29	4830	2426	188	**7444**
30	4029	200	2457	**6686**
Ai	**33877**	**4808**	**3914**	**42599**

The predicted flow values obtained through testing of modular network are given below with the RMSE value as 5827.

Table 5.13 The predicted flow values obtained through testing of modular network

	28	**29**	**30**	**Pj**
28	17961	9085	8590	**35636**
29	10388	3392	3145	**16925**
30	10051	3231	2994	**16276**
Ai	**38400**	**15708**	**14728**	**68836**

The predicted flow values obtained through testing of unconstrained gravity model are given below with the RMSE value as 5567.

Table 5.14 The predicted flow values obtained through testing of unconstrained gravity model

	28	**29**	**30**	**Pj**
28	1778	2085	1312	**5175**
29	2190	2568	1617	**6375**
30	1383	1622	1021	**4025**
Ai	**5350**	**6275**	**3950**	**15575**

5.4 Comparison of Results

The RMSE values of the trained neural networks which indicate the prediction performance of the tested models are given in Table 5.15. The network

sizes varying between three and twenty-four are shown in the first column, whereas RMSE values are presented in the other columns. These RMSE values versus network sizes are also depicted graphically in Figure 5.2.

Table 5.15 Test results of trained models

Network Size	**NETDIM**	**Modular**	**Gravity**	**Back-propagation**
3 node	2865	5827	5567	2436
6 node	2743	3050	4353	7185
9 node	1956	2506	2754	2641
12 node	1264	1287	2137	22672
15 node	1465	1561	2007	2206
18 node	1233	1330	1803	9721
21 node	1184	1225	1611	869
24 node	1004	1048	1672	1181

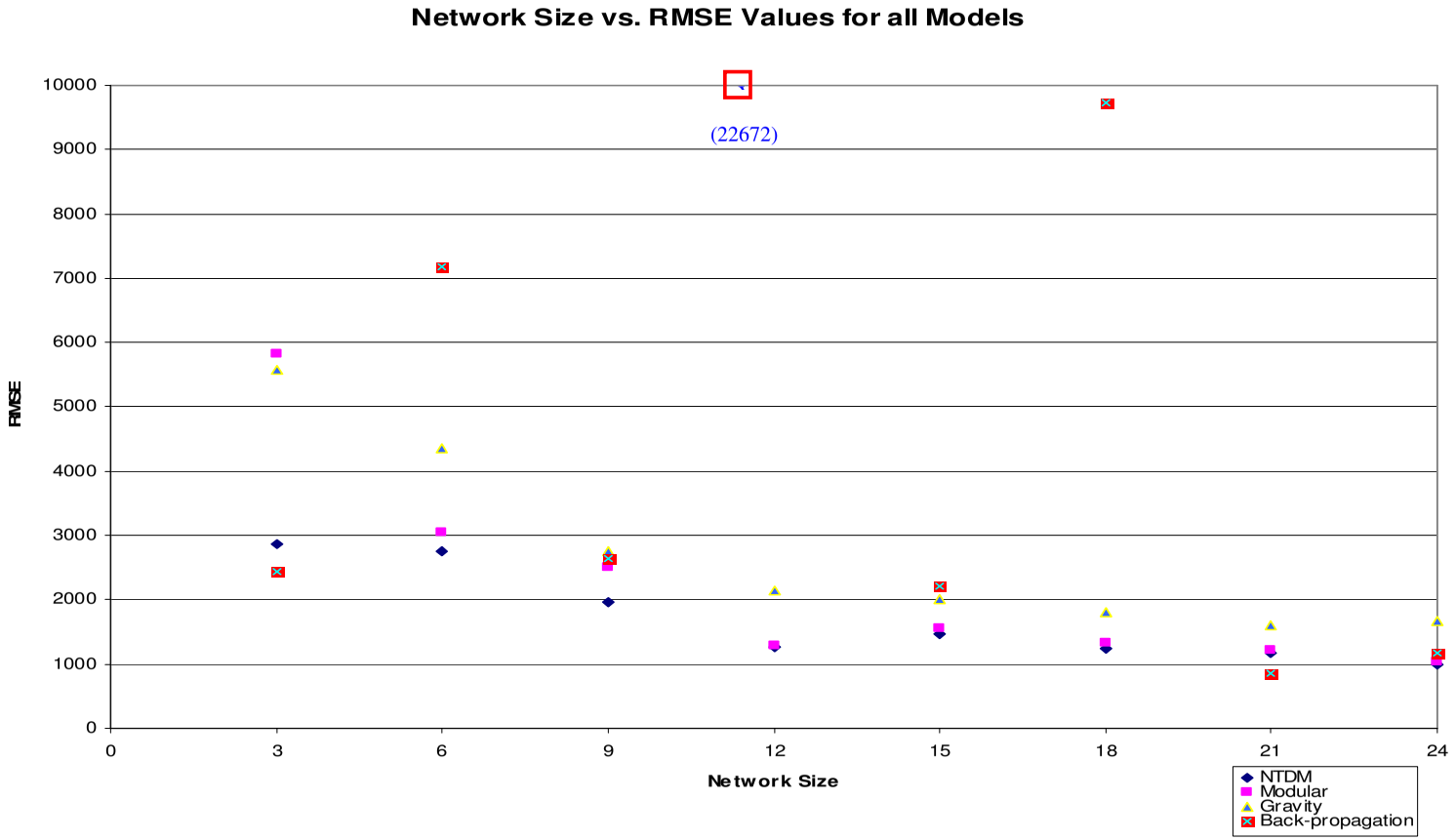

Figure 5.2 Network size vs. RMSE values for all models

As can be seen from Table 5.15 and as well as from Figure 5.2 NETDIM has the lowest RMSE values for different sized models. Although RMSE values are obtained with testing of network sizes of three nodes and twenty-one nodes, the back-propagation seem to be lesser than those of NETDIM. When the whole set of the values is examined, it can be seen that there are significant fluctuations in RMSE values. The fluctuation indicates that simulation results obtained through use of back-propagation model are highly unpredictable. In other words, one cannot depend the predictions on the simulation results. In addition, when RMSE values of NETDIM are compared with those of the modular neural network, it can be seen that NETDIM has the lowest values. Therefore, it can be stated that NETDIM performs better than back-propagation and modular neural models in predicting the zonal trips regardless the size of networks. It can be further stated that NETDIM has better predictive performance than the gravity model.

At this point, it seems necessary to state some major points about those values given in Table 5.15 in order to illustrate the prediction levels of each model. These are:

a) There is a continuous decrease in the RMSE values for the NETDIM approach except the fifteen-node network. For example, three-node network has RMSE value of 2865 whereas twelve-node network has RMSE value of 1264 with a drop of about 56 %. However, for fifteen-node network, there is an increase in RMSE values in the range of 16 %. This 16 % increase when compared to 56 % decrease can be regarded as a slight increase. After this slight increase, the decrease in RMSE values further continues. For twenty-four-node network, the RMSE value drops to 1004 where this decrease is about 32 %.

b) A similar trend in the RMSE values of the modular network with NETDIM can be observed. There is a continuous decrease in the RMSE values of the modular network approach except for the fifteen-node network. For example, the three-node network has RMSE value of 5827 whereas the twelve-node network has an RMSE value of 1287 with a drop of about 78 %. However, for fifteen-node network, there is an increase in RMSE values in the range of 21 %. This 21 % increase when compared to 78 % decrease can be interpreted as a slight increase. After this slight increase, the decrease in RMSE values further continues. For the twenty-four-node network, the RMSE value drops to 1048 where this decrease is about 33 %.

c) Just like the NETDIM and modular networks, there is a continuous decrease in the RMSE values of the gravity model except for the twenty-four-node network with a slight increase of about 4 % that can be considered a negligible increase.

d) When the RMSE results related with back-propagation neural networks are analysed, it can be easily seen that there is no increasing or decreasing trends in the values. These values are fluctuating as the network size varies. For example, the three-node network has the RMSE value of 2436 whereas the six-node network has the RMSE value of 7185 where the increase is 195 %. Then the nine-node network has the RMSE value of 2641 where the decrease is 63 %. Whereas for the twelve-node network, the RMSE value of 22672 is a drastic increase in the RMSE value nearly about 750 % increase. Then, for the fifteen-node network, the RMSE value of 2206 shows a sudden drop again. Similar fluctuations in the RMSE value can also be seen from the Table 5.15 for other sized networks.

From the experience gained by training and testing the neural models in this work, there seems that there are three main advantages of the NETDIM and modular approach over the back-propagation approach due its architecture. These can be stated briefly as:

1) The final optimised weights found by NETDIM and modular approaches do not vary from those found if the training phase is performed once more with the same input data. However, this is not true for the back-propagation neural models. In contrast, the final optimised weights found by back-propagation approach after a training phase is completed do vary for next training phases. This obviously introduces fluctuations in the RMSE values.

2) Since the number of neurons is kept constant as 3 (equal to the number of input parameters), optimum number of neurons in the second layer does not need to be determined by the trial and error procedure used in training phases of back-propagation model approach.

3) The momentum factor is the only factor that should be changed in the training phase in contrast to changing parameters as goal achievement and minimum gradient as in back-propagation neural models. This gives a flexibility in the use of the NETDIM and modular model approaches as it is not required to change any parameter that speeds up the process time except the momentum factor.

In summary, it can be said that NETDIM gives the best, the modular approach provides the second best, the unconstrained gravity model furnishes the third best results and back-propagation neural networks gives the least reliable estimates. Thus, it can be concluded that the NETDIM performs better than all the other three models for all network sizes.

CHAPTER 6

SUMMARY AND CONCLUSIONS

In this dissertation, a neural trip distribution model (NETDIM) as a recommended approach has been developed and the predictive performance of this model compared with back-propagation neural, modular neural and unconstrained gravity models. The objective of this work was to illustrate the predictive performances of these models by comparing their levels of prediction rather than to demonstrate how well they predict a given set of data. The comparison of the levels of prediction of the models were based on the Root Mean Squared Errors of the predicted and the observed zonal trips for various sized networks.

For the generation of different sized networks, a thirty-node sized network was chosen and the data sets regarding the zonal trips and distances were taken from the Bursa Transportation Master Plan. As the data sets are used solely for hypothetical investigation, any problem related with the acquired data set of a chosen network is neither questioned nor addressed.

The networks with various nodes each corresponding to a particular sized network are generated from the thirty-node sized largest network. This generation is simply done by assigning the upper most left hand and lower most right portions of the flow and distance matrices of that large network to smaller ones. The upper most left hand portions are used for training and calibrating, the lower most right hand portions are used for testing purposes.

Test results of trained neural models and calibrated gravity model for various sized networks show that NETDIM has the lowest, the modular model has the second lowest and gravity model has the third lowest values in terms of RMSE. The significant fluctuations of RMSE values obtained through the analysis of the back-propagation model indicate that this model has the least predictive capability. Thus, it can be said that NETDIM performs better than the other models in predicting zonal trips regardless the network sizes.

From the experiences gained in this work with the neural models, it may be concluded that there are three advantages of NETDIM over other neural models which deserve mention. These are:

- The final optimised weights found by NETDIM and modular approaches do not vary than those found if training phase performed once more with the same input data in contrast to the back-propagation neural models
- The optimum number of neurons in the second layer do not need to be determined by a trial and error procedure used in training phases of back-propagation model approach.

- The momentum factor is the only factor that should be changed in the training phase in contrast to changing parameters as goal achievement and minimum gradient as in back-propagation neural models.

As a summary, it can be said that NETDIM provide better level of predictions than modular approach, unconstrained gravity model and back-propagation neural network. Thus, it can be concluded that NETDIM has better predictive performance than all the other three models for all network sizes.

Despite the efforts devoted to the analyses of all of the approaches discussed in this dissertation, there are major areas that still need to be researched. First, NETDIM should be compared with the doubly-constrained gravity model rather than the unconstrained one using different sized networks than those used in this research. Second, as back-propagation has the least predictive capabilities than the other neural models due to the fluctuations in predicted values, other neural models should also be used in the analyses, such as radial basis models. Third, NETDIM trained with a small sized network should also be tested with a large sized network to decide upon its potential applicability.

REFERENCES

[1] Fotheringham A.S., Dignan T., 1984. "Further contributions to a general theory of movement". Annals of the Association of American Geographers 74, p. 620-633.

[2] Fotheringham A.S., 2001. "Spatial Interaction Models". International Encyclopedia of the Social & Behavioural Sciences.

[3] Southworth F., 1983. "Temporal versus other impacts upon trip distribution model parameter values". Regional Studies 17, p. 41-70.

[4] Meyer R.J., Eagle T.C., 1982. "Context-induced parameter instability in a disaggregate-stochastic model of store choice". Journal of Marketing Research 19, p. 62-71.

[5] Fotheringham, A.S., 1981. "Spatial Structure and the Parameters of Spatial Interaction Models", Annals of the Association of American Geographers, 71, p.426-436.

[6] Fotheringham, A.S., Brunsdon C., Charlton M., 2000. "Quantitative Geography: Perspectives on Spatial Data Analysis". Sage, London.

[7] Wilson A.G., 1967. "Statistical theory of spatial trip distribution models". Transportation Research 1, p. 253-269.

[8] O'Kelly M.E., 1986. "Activity levels at hub facilities in interaction networks". Geographical Analysis 18, p. 343-356.

[9] Papacostas C.S., Prevedouros P.D. 2001, "Transportation Engineering & Planning". Prentice Hall.

[10] Ortuzar J. de Dios, Willumsen L.G., 1994. "Modelling Transport". Wiley, Chichester.

[11] John W. Dickey, 1976. "Metropolitan Transportation Planning"

[12] Ji-Rong Xie, "A neural network approach to modelling and predicting intercity passenger flows" PhD dissertation, Indiana University.

[13] Hanson S., 1986. "The Geography of Urban Transportation", The Guilford Press.

[14] Black W.R.,1995, "Spatial interaction modelling using artificial neural networks". J. Transport Geography 3 (3), p. 159-166

[15] Fotheringham, A.S., 1983. "A New Set of Spatial Interaction Models", Environment and Planning A 15, p.15-36.

[16] Wilson A.G., 1975, "Urban & regional models in geography & planning", John Wiley & Sons, Great Britain.

[17] Ramon y Cajal, S., 1911. "Histologie du systeme nerveux de l'homme et des vertebras. Paris: Maloine; Edition Francaise Revue: Tome I, 1952; Tome II, 1955; Madrid: Consejo Superior de Investigaciones Cientificas.

[18] Shepherd, G.M., and Koch C., 1990. "Introduction to synaptic circuits." . In The Synaptic Organisation of the Brain (G.M.Shepherd ed.), p.3-31. New York: Oxford University Press.

[19] McCulloch, W.S., and Pitts, W., 1943. "A logical calculus of ideas imminent in nervous activity." Bull. Math. Biophysics, 5, p. 115-133.

[20] Hebb, D.O., 1949. “The Organisation of Behaviour”. Wiley, New York.

[21] Minsky, M., 1954. “Neural Nets and the Brain-Model Problem”, Ph.D. thesis, Princeton University.

[22] Rosenblatt, F., 1958. “The Perceptron: A probabilistic model for information storage and organisation in the brain.”, Psychological Review, 65, p. 386-407.

[23] Widrow, B., and M.A. Lehr, 1990. “30 years of adaptive neural networks: Perceptron, madaline and back-propagation.” Proceedings of the IEEE 78, p. 1415-1442.

[24] Minsky, M., and Papert, S., 1969. “Perceptrons”. MIT Press, Cambridge, MA.

[25] Anderson, J.A., Silverstein, J.W., Ritz, S.A., and Jones R.S. 1977. “Distinctive features, categorical perception, and probability learning: Some applications of a neural model”. Psychological Review, 84, p. 413-451.

[26] Grossberg, S., 1980. “How does a brain build a cognitive code?”. Psychological Review, 87, p. 1-51.

[27] Kohonen, T., 1977. “Associative Memory: A System-Theoretical Approach.”. Springer-Verlag, New York.

[28] Hopfield, J.J., 1982. “Neural networks and physical systems with emergent collective computational abilities”. In Proceedings of the National Academy of Science”, 79, p 2554-2558.

[29] Feldman, J.A., and Ballard, D.H., 1982. “Connectionist models and their properties”. Cognitive Science, 6(3), p. 205-254.

[30] Smolensky, P., 1988. “On the proper treatment of connectionism”. Behavioural and Brain Sciences, 11, p. 1-23.

[31] Rumelhart, D.E., McClelland, J.L. and the PDP Research Group (eds.), 1986. "Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Volume 1 and Volume 2. MIT Press, Cambridge, MA.

[32] Rumelhart, D.E., McClelland, J.L., and Williams, R.J., 1986. "Learning internal representation by error propagation.". In Parallel Distributed Processing: Explorations in the Microstructures of Cognition, Volume 1 and Volume 2. MIT Press, Cambridge, MA.

[33] Sejnowski, T.J., and Rosenberg, C.R., 1987. "Parallel networks that learn to pronounce English text.". Complex Systems, 1, p. 145-168.

[34] Werbos, P.J., 1974. "Beyond Regression: New Tools for Prediction and Analysis in the Behavioural Sciences". Ph.D. thesis, Harvard University.

[35] Parker, D.B., 1982. "Learning Logic", Invention Report S81-64, File 1. Office of Technology Licensing, Stanford University.

[36] Beale R., Jackson T., 1988. "Neural Computing: An Introduction", Department of Computer Science, University of York.

[37] Haykin Simon, 1994, "Neural Networks, A Comprehensive Foundation"

[38] Hornik K., Stinchombe M. and White H., 1989. "Multilayer Feedforward Networks are Universal Approximators", Neural Network, 2, p.359-366.

[39] Cybenko G., 1989. "Approximation by Superpositions of Sigmoidal Function", Mathematics of Control, Signals, and Systems, 2, p.303-314.

[40] Fu, Li Min, 1994. "Neural Networks In Computer Intelligence". McGraw Hill.

[41] M.Mozolin, J.-C. Thill, E. Lynn Usery, 1999, "Trip distribution forecasting with multilayer perceptron neural networks: A critical evaluation", Transportation Research Part B, p. 53-73.

[42] Fischer M. and Gopal S., 1994. "Artificial Neural Networks: A New Approach to Modelling Interregional Telecommunication Flows", Journal of Regional Science, 34, p.503-527.

[43] Openshaw S., 1993. "Modelling Spatial Interaction Using a Neural Net", in Fischer M. and Nijkamp P. (eds.), Geographic Information Systems, Spatial Modelling and Policy Evaluation, p.147-164, Springer Verlag, New York.

[44] Fotheringham A.S., O'Kelly M.E. 1989, "Spatial Interaction Models: Formulations and Applications". Kluwer Academic Publishers, Dordrecht, The Netherlands.

[45] Slater P.B., 1976. "Hierarchical internal migration regions of France", IEEE Transactions on Systems, Man and Cybernetics 6 (4), p.321-324.

[46] Jacobs, R.A., Jordan, M.I., Nowlan, S.J. and Hinton, G.E., 1991. "Adaptive mixtures of local experts". Neural Computation, 3, p. 79-87.

[47] Bursa Transportation Master Plan, July 1987. Middle East Technical University, Transportation Research Centre, Ankara, Turkey.

[48] Demuth Howard, Beale Mark, 2000. "Neural Network Toolbox For Use with MATLAB", User's Guide, Version 4, Copyright 1992-2000 by The MathWorks, Inc.

APPENDIX–A

Table A.1 Literature Survey

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
General Overview of Neural Network Applications to Transportation	Dougherty Mark 1995	This paper attempts to summarise the findings of a large number of research papers concerning the application of neural networks to transportation.			It is postulated that a more rigorous approach to matters such as comparison with other techniques and also the methodology used to design the neural networks would help a clearer picture to emerge as to best practice and future research directions.
Spatial interaction modelling	Black,W.R 1995	Neural network models were used in order to model commodity flows between nine census regions and migration flows between nine census regions. Model structure is composed of 3 input neurons, 3 hidden neurons, 1 output neuron (3/3/1).	Back-Propagation feed forward	An unconstrained, a fully constrained gravity model and a regression model	50% better than all of the reference models

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Openshaw, S. 1993	Neural network models were used as systems which are able to model evolutionary processes and spatial interaction in urban structure.	Back-Propagation feed forward	Unconstrained spatial interaction model	Better than the reference models when time parameter is involved
	Fischer, M. and S. Gopal 1994	Artificial neural networks were used as a new approach to modelling interregional telecommunication flows between 32 regions in Austria. Model structure was 3/3/1	Back-Propagation feed forward	Conventional regression of gravity type	Good, emphasise on parameters, structure and sensitivity of the model.
	Boritz, J.E. and Kennedy, D.B. 1994	Neural networks were analysed as a possible tool of prediction and were compared to the conventional model of spatial interaction of interregional telecommuting flows.	Back-Propagation feed forward	Unconstrained spatial interaction model	Better than the reference models.
	Chin et al 1994	A neural network was used in order to synthesise Origin-Destination flows in a traffic circle (12/5/36)	Back-Propagation feed forward		Good, but inability to generalise.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Traffic flows freeway flows	Nijkamp Peter, Reggiani Aura, Tritapepe Tommaso 1996	In this paper, a modal split problem is analysed by means of two competing statistical models, the traditional logit model and the new technique for information processing, the feedforward neural network model	Feed-forward neural network model	Traditional logit model	The neural network model turns out to have a slightly better performance, although there are still critical problems inherent its application.
	Dia Hussein, Rose Geoff 1997	Discusses a multi-layer feedforward neural network incident detection model that was developed and evaluated using field data.	Multi-layer feedforward neural	Incident detection model in operation on Melbourne's freeways.	Results presented in this paper provide a comprehensive evaluation of the developed model and confirm that neural network models can provide fast and reliable incident detection on freeways.
	Pursula, M. 1998	Two-dimensional estimation of speed and flow density relationships with back-propagation neural networks. Neural networks were applied to different dimensional estimation on May model and Van Aerde Model.	Back-Propagation feed forward		The neural network is a useful tool in estimation of the parameters.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Park, D. and L. Rilett 1998	Forecasting multiple-period freeway link travel times using modular neural networks. Two unsupervised clustering techniques: Kohonen SOFM and fuzzy c-means model and a back-propagation neural network were used.	Unsupervised Kohonen self-organising feature maps and a back-propagation	Kalman filtering model	The best neural network of the three neural network models that were compared with the reference model
	Dougherty, M. 1998	Exploring traffic systems by elasticity analysis of neural networks to route choice considering congested conditions (24/8/3)	Back-Propagation feed forward	Logit model	Better. Elasticity testing of neural networks aid understanding of transport system.
	Nakatsuji T. and S. Shibuya 1998	Neural network models (3 and 4 layers) were applied to traffic flow problems by simulations.	Back-Propagation feed forward and Kohonen feature map model	Nonlinear equations on occupancy-speed curve	Both neural network models are useful in optimum traffic control problem and traffic situations classification with improved estimation precision.
	Mussone, L. 1994	Application of neural networks in transportation an uninterrupted flow control study, (4/2/1)	Back-Propagation feed forward	Mode Choice Model	Good. Precision of prediction depends on choices of training set.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Cheu Ruey L., Ritchie Stephen G. 1995	Three types of artificial neural network models, namely the multi-layer feedforward (MLF), the self-organising feature map (SOFM) and adaptive resonance theory 2 (ART2), were developed to classify traffic surveillance data obtained from loop detectors, with the objective of using the classified output to detect lane-blocking freeway incidents.	Multi-layer feedforward (MLF), the self-organising feature map (SOFM) and adaptive resonance theory 2 (ART2)	California, McMaster and Minnesota	Results and analyses with data from the study site as well as the three test sites have shown that the multi-layer feedforward consistently detected most of the lane blocking incidents and typically gave a false alarm rate lower than the California, Mcmaster and Minnesota algorithms currently in use.
Traffic management	Spall James C., Chin Daniel C. 1997	This paper presents a fundamentally different approach for optimal signal timing that eliminates the need for complex models.			S-TRAC has the potential to deliver real-time-system-wide signal timings in a practically feasible manner.
	Chang, Y. and C.C. Shen 1998	The application of fuzzy multiobjective and artificial neural networks on urban public transport equilibrium and on constructs the modal choice model. Experiment with different number of units in the hidden layers was conducted. The focus was on urban transit equilibrium system.	Back-Propagation	Mode choice Model	No significant effects on accuracy.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Kontratos et al 1998	Predicting parking characteristics: the use of 3-layer neural network to support parking management.	Back-Propagation		Used successfully for clustering the patterns of the various days and time of the day travel and parking behaviour.
	Matsui, H. and M. Fujita, 1998	Travel time prediction for freeway traffic information by neural network driven fuzzy reasoning.	Back-Propagation	Conventional present travel time estimation	Better than the reference model.
Traffic control	Huang, X.P. and Prahlad Pant, D. 1994	A new simulation model interfaced with a neural model, model structure: 27/27/27/1	Back-Propagation feed forward		Good.
	Henry J.J., Farges J.L., Gallego J.L. 1998	Neuro-fuzzy techniques are proposed here to control each light of an intersection, at one-second intervals.	Neuro-fuzzy acceleration of Forward Dynamic Programming	No rules to apply	Simulations on different intersections show decreases in delays with respect to fixed timing from 0% to 30% for neuro-fuzzy control, and from 15% to 35% for neuro-fuzzy acceleration of Forward Dynamic Programming.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Bullock et al	New image based vehicle detection system – neural network vehicle tracking model.	Back-Propagation feed forward	Pattern matching algorithms	Can reliably detect vehicles and the system is more adaptable.
	Masaki et al 1992	Quasi real time: recognition of vertical road sign in a running vehicle, structured programming, expert system approach.			
	Nkatsuji, T. and Kaku, T. 1991	A multi layer neural network model was introduced in order to realise a self-organising traffic control system. The model was trained by back-propagation methods and optimisation by stepwise method combining Monte Carlo with feedback method.	Back-propagation		Good accordance with reference ones.
	Huang, X.P. and Prahlad Pant, D. 1994	A new simulation model interfaced with a neural model for evaluating dilemma zone problems.	Back-Propagation feed forward		Good.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Povtin, J.Y. 1992	A survey on neural network models for solving Travelling Salesman Problem.			Hopfield-Tank model, elastic net and the self-organising maps are introduced, as well as many variants.
	Rourke, A. and Bell, M.G.H. 1992	Application of a neural network based system. Vision process applied to traffic monitoring.			
	Wan, C.L. and Dickinson, K.W. 1992	Application of a neural network based system. Vision process applied to traffic monitoring.	Back-Propagation	Conventional systems	Neural network based system can perform as well as conventional systems with more capabilities. Troubles with 'over training'
Driver Behaviour	Yang et al	A neural network was used to model the behaviour of driver when they are to choose between freeway or side road.			The acceptance rate of advice and the quality of advice are closely correlated.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Onken R., Feraric J.P. 1997	A driver monitoring and warning system is presented which adapts its warning messages to warning thresholds acceptable to the driver.	Adaptive resonance technique networks		The network parameters were adjusted through genetic algorithm optimisation.
	Mussone et al 1998	A methodology for modelling driver behaviour in signalised urban intersections	Back-Propagation feed forward		Models are interesting both in the designing of a fixed plan for traffic or in dynamic control of intersections.
	Tabuko, N. 1991	A neural network was applied to a driving simulator in a study of the driver's behaviour.	Back-Propagation		Investigate driving behaviour from neural network system model.
Accidents and the human factor	Hussein, D. and G. Rose 1998	The impact of data quantity on the performance of neural network freeway incident detection models was analysed. Real time and data were used, model structure: 6/3/1	Logic on Projection network		The impacts of the sample size were analysed.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Peeta,S. and D. Das	A framework in corporating a neural network based continuous learning capability to detect freeway incidents.	Back-Propagation feed forward/ least squares technique	California algorithm decision tree technique	Neural network implementations of the California McMaster algorithms perform better than without implementations and off-line calibration.
	Award, W. and B. Janson 1998	Prediction and detection for truck accidents at freeway ramps in Washington State using a regression and a neural network – the hybrid neural system called the Adaptive Networks Based Fuzzy Inference System (ANFIS).	Hybrid neural system called ANFIS.	Regression	ANFIS produce better results, over fitting problem exists.
Travel Behaviour	Faghri, A. and A. Sandeep 1998	Analysis of performance of back-propagation Neural Networks with different training parameters 5/1-9/1.	Back-Propagation feed forward		Emphasise on back-propagation architectures such as one to nine hidden layers and sensitivity of the models.
	Himanen, et al 1998	Daily travelling behaviour in terms of mode split, average daily travel distance, speed, time etc. are viewed by self-organising maps.	Unsupervi-sed neural networks		Ability to capture in an intuitive manner some of the complex structure of daily travelling behaviour.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Reggiani, A. and T. Tritapepe 1998	A neural network was applied to commuter's mobility between 224 municipalities in the metropolitan area of Milan. A modal split case.	Back-Propagation feed forward	Logit model	Better than the reference model.
	Schintler, L.A. and O. Olurotimi 1998	Neural networks as adaptive logit models were applied to commute flows to, from and within 250 districts in Washington DC area, model structure : 3/3/3	Back-Propagation feed forward	Binary logit model	Good. Illustrates how to derive the adaptive regression coefficients and elasticity of demand.
	Shmueli et al 1998	Neural network analysis is conducted to travel behaviour using household and individual travel data, model structure: 30/8/4	Back-Propagation feed forward	Classification and Regression trees	A useful method for relatively assumption-free reconnaissance of large socioeconomic data sets.
	Nykanen et al 1998	Research on daily travelling of the Finnish people. Self-organisation neural network was used with socioeconomic data for viewing daily travel behaviour.	Unsupervised		

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Freight flows	Nijkamp et al 1997	Focuses on the Trans Alpine freight transport systems in the light of the future integration of single national transport system into the European transport network of 108 zones.	Back-Propagation feed forward	Binary logit model	Better than the reference model.
Short-term forecasting	Park et al 1998	Short-term freeway traffic volume forecasting using radial basis function neural networks. The two neural networks are tested with different combinations of number of input and hidden neurons.	Back-Propagation feed forward/ Radial basis function neural network.	Taylor series/ Exponential smoothing models.	Results vary. The potential of forecasting short-term volume in real time is demonstrated.
	Perrin H.J., and P.T. Martin 1997	On-line comprehension traffic flow estimation from link flow detectors. Solved a minimal cost problem in which the objective function served to provide solutions to unknown link and turning movements network-wide.	Weighted objective function.	Origin-Destination estimating techniques.	Estimated real time link flows and turning flows. a**yer feedforward neural**

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Guoqiang Zhang, B. Eddy Patuwo, Michael Y.Hu 1997	Presents a state-of-the-art survey of artificial neural network applications in forecasting.			Provides a synthesis of published research in this area, gives insights on artificial neural network modelling issues and supplies the future research directions.
Application Field	**Authors**	**Description**	**Type of Network**	**Comparison Model**	**Result/peculiarity**
	Dia Hussein 2001	This paper discusses an object-oriented neural network model that was developed for predicting short-term traffic conditions on a section of the Pacific Highway between Brisbane and the Gold Coast in Queensland, Australia.			Results represent substantial improvements on conventional model performance and clearly demonstrate the feasibility of using the object-oriented approach for short-term traffic prediction.
Intersection signal analysis	Fan J. and M.Saito 1998	Application of artificial neural networks for level of service analysis of signalised intersections.	Back-Propagation feed forward		The neural network can be used to analyse a variety of signalised intersections. Potential for development of an optimal signal time algorithm for the realisation of dynamic signal control.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Modelling vehicle discharge headway	Tong H.Y., W.T. Hung, 2000	Application of artificial neural networks to simulate the discharge headway of individual queued vehicles.	Back-Propagation feed forward	Other headway models	Artificial neural networks could produce reasonable discharge headway estimates for individual vehicles.
Congestion detection	Kilby et al 1993	Experiments were conducted to a) detect congestion, b) make short term forecast, c) investigate multivariate relationship.			Neural networks can be used effectively for such work. Advantage and disadvantages of using neural networks are described.
	Chen et al 1991	Neural network models were applied to automated detection of non-recurring congestion on urban freeways.			Encouraging results.
Application Field	**Authors**	**Description**	**Type of Network**	**Comparison Model**	**Result/peculiarity**
	Burattini et al 1998	A new traffic light single (congested) junction control system was implemented by a symbolic neural network.	Neural rule-based model.	Classical fixed time approaches.	Better. The neural weightless approach as congestion detector obtained better traffic control performance.
	Goodwin, J. 1991	Neural networks were used to predict congestion in heavy-duty condition.			

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Traffic pattern analysis	Lyons et al 1998	Factors influencing the performance of a neural network decision model: a case study using simulated data	Learning Vector Quantization (LVQ)		LVQ is an alternative of propagation feedforward approach specifically for classification tasks.
	Belagoui, B and Blosseville, J.M 1993	Neural networks were used to recognise the patterns of a road traffic scene, model structure: 18/10/7/4	Back-Propagation feed forward	Discriminate analysis	Similar results as the reference model.
	Ritchie, S.G. and Cheu, R.L. 1993	Spatial and temporal traffic pattern are recognised and classified by a neural network trained with a simulation of freeway incident detection.			Neural network models have the potential to achieve significant improvements in incident detection performances.
	Dickinson, K.W. and Wan, C.L.	Neural network pattern recognition techniques for traffic monitoring are described.			

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Telecommunication	Fischer, M. and S. Gopal 1994	Neural networks were used as a new approach to modelling interregional telecommunication flows between 32 regions in Austria, model structure: 3/3/1.	Back-Propagation feed forward.	Conventional regression of gravity type.	Good, emphasise on parameters, structure and sensitivity of the model.
	Junius, M. and O. Kennmann 1995	The (GSM) Global System for Mobile Communications provides measurement data about the radio propagation situation that was used for power control and handover decisions. Fuzzy control for the handover process and pattern recognition was used for locating mobile stations.	Node Plitting Distance Classifying Network – Neural Network specifically designed for the level trace pattern recognition problem.	GSM	The performance of the handover procedure is improved.
	Comellas, F. and J. Ozon 1995	A neural network of genetic algorithm was used as a graph colouring algorithms for the channel assignment in radio networks.	Genetic algorithm	Simulated annealing and heuristics	Both reference models outperformed the genetic algorithm
	Berger, M. O. 1995	A non-positive neural network model was used to solve channel assignment problems in arbitrary cellular mobile radio system.	Modified version of the 2-dimensional Hopfield-Tank neural network.	Traditional neural nets.	Convergence is guaranteed and much faster.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Zeremba et al 1995	Application of neural networks to the optimisation of link bandwidth allocation in integrated Services Digital Networks. Neural networks were shown to perform favourably in the process of the acceptance or rejection of a call in multi service communication network.	Feedforward, Multi-layer neural network	Dynamic Trunk Reservation Policy Model	Perform at 95-99% of the desired result obtained by the reference model.
	Engelbrecht, A. and I. Cloete 1995	A feedforward neural network was used to approximate the distribution of both primary (first-offered) traffic and overflow traffic in a telephone network	Feedforward, Multi-layer neural network	Traffic model and dimensioning model	The neural network accurately approximates both primary and overflow traffic distributions.
	Chiueh T. and L. Bu 1995	A neural network was used to solve the shortest path problem between any two nodes in a given graph.	An analog circuit network		Also has potential to solve fastest path routing problems in a communication network with node delays as well as link delays.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Estrella et al 1995	The asynchronous transfer mode (ATM) is an essential technology in integrating multimedia communications services. An ATM multiservice call admission control using back-propagation learning was presented.	Back-propagation	Standard ATM	Simulation results show that complicated admission boundaries can be learned.
	Felix Lor, W.K. and Y.M. Wong 1995	A decentralised neural approach dynamically distributes telephonic traffic among alternate routes in circuit-switched networks according to the fluctuating number of free circuits and the evolving call attempts.	A simplex centralised neural network algorithm	Maximum free circuit	The neural network can learn the controlling functions prescribed by the centralised controller and have a comparable performance.
	Dixon et al 1995	A Hopfield neural network was used to solve the routing problem in a communication network..	A Hopfield neural network which does not use constraint terms.	Traditional methods with constraints.	The Ability to find optimal or near-optimal valid routes and help to overcome the scaling problem.
Traffic theory	Davies, H. and Winnet, M.A. 1993	Neural network technology was used to identify association in the collected data for a study of pedestrian accidents.			

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Kikuchi et al 1993	A method which identify the trip origin-Destination (O-D) matrix when many pair of values for the right hand side (B) column and the bottom row (A) of the matrix are given.	Back-propagation		Useful to problems that requires identification of the cause-effects relationship when many sets of data are available.
	Clark et al 1993	Neural networks were applied to short term forecasts of traffic flow.		Linear regression and ARIMA	
	Vythoulkas, P.C. 1992	System identification (SI) and artificial neural network models for short term forecasting of traffic of traffic condition in urban network were investigated.	Back-propagation Dew learning rule based on Kalman filtering	Linear models	Both methods are promising.
	Dougherty et al 1993	Application to recognition problems: a) state of congestion of road, b) short term forecasting			Provide a powerful method of analysing, interpreting complex data sets.
	Dougherty et al 1997	Discusses the relative merits of artificial neural networks and time series methods for traffic forecasting and summarises the findings from a comparative study of their performance for motorway traffic in France.	A simplex centralised neural network algorithm	ARIMA and ATHENA	Provide a powerful method of analysing, interpreting complex data sets.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Hai Yang, T. Akiyama and T.Sasaki 1998	The origin-destination flows in each short time interval are estimated through minimisation of the squared errors between the predicted and observed exiting counts which are normalised using a logistic function.	Back-propagation		Numerical results show that the back-propagation-based model is capable of tracking the time variations of the origin-destination flows with a high stability.
Dynamic travel time estimation	Alpaydın Ethem, Altınel İ.Kuban, Aras Necati 1996	In this paper, it is proposed to use nonparametric approaches using neural networks for estimating actual distances.	Back-propagation algorithm	Parametric distance functions	Estimating actual distances has many applications in location and distribution theory.
	Gong Zhejun, 1998	In this paper, the Hopfield neural network model is used to estimate urban Orientation-Destination (O-D) distribution matrix from the link volumes of the transportation network, so as to promote the solving speed and precision.	Hopfield neural network		The Hopfield neural network model has the ability of quick computation and finding the global optimal solution to the problems to be solved, and the simulation experiment of using it to estimate the O-D distribution is satisfactory.
	Palacharla Prasad V., Nelson Peter C. 1999	The paper represents a fuzzy reasoning model to convert lop detector data into link travel times obtained from empirical studies.		Available statistical models.	Incorporates flexible reasoning and captures non-linear relationship between link specific detector data and travel times.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Prediction	Shmueli Deborah, Salomon Ilan, Shefer Daniel 1996	This article explores the application of neural networks to a behavioural transportation-planning problem. It concentrates on the extent to which neural networks can combine the relative simplicity of aggregated transportation models, without the latter's complexity.	Feedfor-ward back-propagation		There are recent developments which might be applied to more general network topologies.
Road surface / pavement	Keseko, M.S. 1993	Neural network models were used in image processing and pavement crack detection.			Demonstrates the feasibility of the neural network for detection, classification and quantification of highway pavement surface cracking.
	Ritchie et al 1991	A system that integrates three artificial intelligence technologies: computer version, neural networks and knowledge-based system in addition to conventional algorithmic and modelling techniques were presented.	Back-propagation		Results demonstrate the feasibility of a neural network approach.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Eldin Neil N., Senouci Ahmed B. 1995	This paper discusses the use of back-propagation neural networks as a management tool for the maintenance of jointed concrete pavements.	Back-propagation feed forward		A statistical hypothesis test was conducted to demonstrate the system's fault-tolerance and generalisation properties.
	Tutumluer and Seyhan 1998	Neural network modelling of anisotropic aggregate behaviour from repeated load triaxial tests.	Back-propagation feed forward	Standard AASHTO test for pavement	Better. The applied stress state and the aggregate properties were found to affect the generalisation and prediction ability of artificial neural networks.
	Kim et al 1998	Prediction of layer module from FWD and surface wave measurements using artificial neural networks was presented. The prediction algorithm made use of surface wave measurements from the FWD and SASW test was developed.	Supervised back-propagation		The new algorithm is more sensitive.
	Shekharan, A.R. 1998	Effect of noisy data on pavement performance prediction by an artificial neural network with genetic algorithm.	Genetic algorithm based on multi-layer feedforward	Regression estimation	Better than the reference model when more errors introduced into the data, inferior than the reference model.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Vehicle design and safety	Bacon et al 1992	Viability of replacing elements of a modern electronic engine control system with trained neural networks.			
	Pomerlau et al 1994	Networks were trained to develop single capabilities and a rule-based technique, combining multiple artificial neural networks with map-based symbolic reasoning to achieve hi-level, goal directed behaviour.			
	Nusser et al 1991	A neural network was applied in an intelligent system for automatic car control.	Feed-forward.		
	Ramli, M.M. and Morris, A.S. 1993	To represent system under uncertainty conditions, a controller that incorporates a neural network self-tuning model was presented.			
Vehicle Assignment Problem	Vukodino-vic Katarina, Teodorovic Dusan, Pavkovic Goran, 1999	In this paper, a neural network is used to refine and adapt the fuzzy system to achieve better performance to control complex processes like the dispatcher's work.	A neuro-fuzzy model.		As a result of the study, on a real set of numerical data, it was shown that the proposed feedforward adaptive neural networks with supervised learning capabilities can be used to tune the initial fuzzy systems.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Transportation / Land Use Systems	Rodrigue Jean-Paul, 1997	In this conceptual paper, an overview of a parallel transportation/land use-modelling environment is provided.	Parallel distributed processing		A set of advantages, disadvantages, drawbacks and some research directions about the usage of neural networks for spatial analysis and modelling is also provided.
Transportation Safety Modelling	Hashemi Ray R., Blanc Louis A. Le, Rucks Canway T., Shearry Angela 1995	In this paper, three models were developed to predict vessel accidents on the lower Mississippi River.		Multiple discriminant analysis and logistic regression	Discriminant analysis and logistic regression were able to correctly classify only 53% and 56% respectively of accident cases into three casualty groups: collisions, rammings or groundings.
Expert Systems	Potvin Jean-Yves, Shen Yu, Dufour Gina, Rousseau Jean-Marc 1995	This paper proposes learning by examples as a means to uncover the human expertise in the domain of expert vehicle dispatching system.		Linear programming	Experimental results are reported on real data from a courier service company.
Fuzzy Logic Systems	Teodorovic Dusan 1999	The paper represents a classification and analysis of the results achieved using fuzzy logic to model complex traffic and transportation processes.			Possibilities are shown regarding the further application of fuzzy logic in the field.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Niitymaki Jarkko, Turunen Esko 2002	The main intention in this study is to tie fuzzy reasoning to many-valued logic framework; a Lukasiewicz many-valued logic similarity based fuzzy control algorithm is introduced, and tested in three realistic traffic control systems.	Matlab Fuzzy Logic toolbox's Mamdani-style system	The compared traffic signal control modes are signalised pedestrian crossing and multi-phase signal control with phase selection.	Similarity-based inference systems offer a competitive method for control in traffic signal design.
	Teodorovic Dusan, Radivojevic Gordana 2000	Two approximate reasoning algorithms are developed in this paper. Using the first one, the decision is given about which vehicle will accept the new request. The second one was used to design the new route and schedule for the vehicle chosen to serve the new request.			The results obtained are very promising.
	Wu Jianping, Brackstone Mark, McDonald Mike 2000	This paper describes the development of the fuzzy logic motorway simulation model (FLOWSIM)			Model validation results have shown that the fuzzy model can closely replicate real systems and in test cases have performed better than some deterministic models such as the 'GHR', 'Gipps' and 'MISSION' models.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
	Lingras Pawan 1998	This paper compares rough and neurofuzzy neural networks.			It is shown that the introduction of rough and fuzzy semantic structures in neural networks can increase the accuracy of predictions.
Railway Systems	Malavasi Gabriele, Ricci Stefano 2001	Research deals with an analysis of the critical behaviour parameters, difficult to be effectively modelled by means of analytical simulation tools. Selection of the self-learning process for the application of the reliability of a railway network capable to work as a part of a wider simulation model of railway traffic; the development of a preliminary version of the model simulating the stochastic failure events and its application to a case study.	Self-learning processes		It will start from the results of the global application and from the determination of the essential parameters capable to provide with the maximum added value the training process of the neural network.
Airline Systems	Lagerholm Martin, Peterson Carsten, Söderberg Bo 2000	A novel method is presented and explored within the framework of Potts neural networks for solving optimisation problems with a non-trivial topology, with the airline crew-scheduling problem as a target application.		Two real world problems.	Very good results are obtained for a variety of problem sizes.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Environmental	Moseholm et al 1996	A neural network was used with lane specific traffic information and on-site wind parameters to investigate the relationships between traffic and carbon monoxide concentrations near a sheltered intersection, model structure: 32/17/1.	Back-propagation feed forward	Traditional statistical techniques for correlation	Good. However, over fitting problem exists due to too few observations and too many predictor variables and hidden neurons.
OTHER					
Prediction of business failures	Fischer, M.M. and Gopal, S 1994	Neural networks were compared against traditional prediction techniques.	Back-propagation feed forward	Discriminate Analysis, Logit-probit	Good, with better configuration.
Regional economics	Roson,R. 1992	Neural networks applied in linear and non-linear regression in a study of economics and regional science.			
	Lee, C. and Sterling, R. 1992	A neural network experiment was proposed to identify probable modes for underground openings from prior case history information, problem of design of tunnels were studied too.		Rule-based expert system.	Neural network system allows input information to be incomplete and vague.

Application Field	Authors	Description	Type of Network	Comparison Model	Result/peculiarity
Market prices	Malliaris, M and Salchenberger, L. 1992	Neural networks were compared against Black-Scholes model in estimating the choices of the market price.	Back-propagation feed forward	Black Scholes model	Good, achieve the configuration.
Inference with spatial knowledge	Leung, L. 1994	Neural networks were proposed as a new tool for analysing spatial inference	Feed-forward, Hopfield, IBAM, ABAM, BAM	Comparison among the different architecture of neural networks	ABAM model is better than other models.
Evolution of urban structure	White, R.W. 1989	Showed that the architecture of neural network was a self-organising system evolving towards a desired result.	Back-propagation feed forward		Better than the reference models.
Target differentiation	Ayrulu Birsel, Barshan Billur 2001	This study investigates the processing of sonar signals using neural networks for robust differentiation of commonly encountered features in indoor robot environments.	Modular and non-modular neural network structures trained with the back-propagation		This work can find application in areas where recognition of patterns hidden in sonar signals is required.

APPENDIX – B

Table B.1 Zonal trips taken from Bursa Transportation Master Plan

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	3275	1500	400	1250	350	4725	1225	625	1725	1250	2275	2125	1650	1275	1250	2575	625	300	875	800	50	575	550	975	1675	625	2150	1375	625	50
2	1550	3800	1025	25	200	850	750	350	350	300	675	275	250	450	350	1525	50	300	625	75	0	75	350	200	175	25	25	400	200	200
3	525	975	2525	300	0	550	275	25	25	75	275	50	100	125	50	300	0	25	200	0	25	125	75	125	100	50	100	0	125	25
4	1350	75	250	1925	350	600	50	100	175	50	325	50	300	125	50	125	50	50	0	50	25	25	50	75	25	0	50	50	175	25
5	450	150	25	300	650	1000	75	50	0	75	200	125	25	100	0	50	50	0	50	0	0	0	0	0	175	25	100	100	75	50
6	4750	875	575	650	975	10175	450	325	800	600	1050	450	425	250	350	600	250	375	475	75	100	300	550	275	350	375	450	1050	1750	700
7	1175	775	250	50	50	450	1725	100	150	100	275	50	75	75	0	125	75	50	50	25	0	200	200	100	50	150	125	175	350	50
8	650	375	0	125	50	325	125	4350	700	125	275	50	200	150	50	175	100	0	100	0	0	0	125	225	575	200	450	450	200	50
9	1550	375	0	150	0	750	175	650	3525	700	825	200	25	150	200	525	250	25	225	75	50	25	525	1375	350	250	150	300	125	100
10	1100	325	75	50	50	625	100	100	775	5925	575	100	150	50	0	2150	600	250	75	325	0	425	775	375	275	25	125	150	250	0
11	2400	575	325	275	175	1000	275	225	800	625	9350	600	500	950	425	1225	300	250	400	125	50	0	550	450	250	50	175	300	575	75
12	2200	325	75	50	125	575	25	100	150	125	525	4225	325	600	600	50	0	150	75	0	0	0	275	225	25	0	100	25	225	50
13	1550	250	100	375	50	450	75	150	25	175	475	450	3375	275	100	100	75	200	100	0	0	50	125	75	100	100	25	425	175	75
14	1300	400	125	125	50	250	100	150	150	25	1000	575	250	1550	125	125	100	100	75	0	0	50	150	25	25	75	50	25	425	0
15	1275	350	50	50	0	300	0	50	175	0	525	650	100	150	3525	225	50	75	125	0	0	25	0	100	150	50	50	50	250	0
16	2550	1525	300	100	50	575	125	200	475	2200	1250	50	150	125	275	17500	625	400	1050	125	0	275	2175	375	100	75	150	325	825	75
17	675	100	0	50	50	250	50	100	250	625	275	25	100	125	50	650	3800	350	100	375	0	250	1475	275	75	125	125	50	225	75
18	300	300	25	50	0	400	50	0	25	250	250	125	175	100	100	425	375	275	375	900	0	75	400	125	100	50	25	200	50	100
19	825	575	175	25	50	450	50	125	175	100	400	75	75	100	125	1025	100	350	5600	750	25	200	300	150	100	25	75	100	275	0
20	775	50	0	50	0	100	25	25	75	325	125	0	0	0	0	125	400	925	750	5325	75	350	850	125	100	25	75	100	325	0
21	75	0	25	25	0	75	0	0	75	0	50	0	0	0	0	0	0	0	0	75	200	25	75	25	0	0	0	75	25	0
22	575	50	75	25	0	300	200	0	25	425	0	0	50	50	25	275	300	75	200	325	50	5425	300	200	50	50	75	25	200	0
23	600	325	75	25	0	425	200	125	525	800	525	250	125	125	0	2075	1400	400	300	825	100	300	3000	925	150	125	150	500	150	0
24	1025	250	125	50	0	300	75	225	1450	375	450	200	75	0	100	350	275	150	150	200	25	175	875	11750	1825	875	550	475	325	100
25	1725	200	75	25	200	375	50	525	350	275	225	50	175	0	100	100	75	75	100	50	0	50	250	1925	11650	850	1675	975	575	50
26	650	0	50	0	25	400	150	200	250	25	100	50	75	75	50	50	125	25	25	25	0	50	150	875	850	4125	600	400	450	175
27	2025	25	100	75	150	425	75	525	150	125	200	50	25	50	50	200	125	25	75	50	0	50	200	575	1675	650	16050	1075	1075	250
28	1425	450	25	25	175	1025	175	375	275	125	300	25	400	50	50	200	50	225	125	100	100	25	550	475	1025	400	1125	18200	1725	900
29	475	300	100	125	100	1700	275	175	150	200	650	225	150	325	350	775	225	50	225	325	50	200	150	300	575	450	1000	1625	625	125
30	75	200	25	25	50	675	0	50	75	0	75	100	75	0	0	50	75	100	25	0	0	0	25	100	25	175	250	950	175	450

Table B.2 Zonal distances taken from Bursa Transportation Master Plan

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0.26	0.85	0.85	1.45	2.45	2.89	1.36	1.94	1.22	1.70	0.65	0.73	2.17	1.70	1.95	2.96	2.81	5.07	4.85	5.72	7.32	5.59	3.47	3.32	2.00	4.71	2.94	4.09	2.99	3.41
2	0.85	0.19	0.85	1.45	2.45	2.89	1.36	1.94	1.22	1.70	0.65	0.73	2.17	1.70	1.95	2.96	2.81	5.07	4.85	5.72	7.32	5.59	3.47	3.32	2.00	4.71	2.94	4.09	2.99	3.41
3	0.85	0.85	0.20	1.20	2.20	2.74	1.68	2.79	2.07	2.55	1.50	1.58	1.92	2.55	2.80	3.81	3.66	5.92	5.70	6.57	8.97	6.44	4.32	4.17	2.85	5.14	2.69	4.84	2.74	3.16
4	1.45	1.45	1.20	0.19	1.05	2.64	1.58	2.69	2.77	3.15	2.10	2.18	2.52	3.15	3.40	4.41	4.26	6.55	6.30	7.17	8.77	7.04	4.92	4.77	3.45	5.04	2.59	4.74	2.64	3.06
5	2.45	2.45	2.20	1.05	0.29	1.47	1.81	2.91	3.77	3.22	2.57	2.65	3.00	3.53	3.87	4.20	4.05	6.34	6.08	6.95	8.55	6.82	4.70	4.55	3.23	4.97	2.53	4.37	2.27	1.89
6	2.89	2.89	2.74	2.64	1.47	0.29	1.25	1.20	2.30	3.11	3.64	3.72	4.60	4.89	5.14	4.37	4.22	6.48	6.23	7.10	8.70	6.97	4.85	4.70	3.38	3.55	1.10	2.95	8.50	1.42
7	1.36	1.36	1.68	1.58	1.81	1.25	0.23	0.40	1.35	1.83	2.01	2.09	2.99	3.06	3.31	3.09	2.94	5.20	4.95	5.82	7.42	5.69	3.57	3.15	1.83	4.20	1.65	3.50	1.40	2.49
8	1.94	1.94	2.79	2.69	2.91	1.20	0.40	0.22	1.45	1.93	2.02	2.10	3.55	3.07	3.32	3.19	3.04	5.30	5.05	5.92	7.52	5.79	3.67	2.57	1.25	4.05	1.60	3.45	1.35	2.45
9	1.22	1.22	2.07	2.77	3.77	2.30	1.35	1.45	0.32	0.68	1.87	1.99	3.39	3.92	5.17	1.94	1.79	3.05	2.80	3.67	5.20	4.57	2.45	3.40	2.08	4.79	2.21	4.81	2.71	3.54
10	1.70	1.70	2.55	3.15	3.22	3.11	1.83	1.93	0.68	0.27	2.35	2.43	3.87	4.40	5.65	2.20	0.90	3.33	3.08	3.95	5.50	3.82	1.70	2.83	1.51	4.06	1.71	4.31	2.21	3.67
11	0.65	0.65	1.50	2.10	2.57	3.64	2.01	2.02	1.87	2.35	0.31	0.68	2.82	1.85	2.10	3.61	3.46	5.72	5.51	6.37	7.97	6.24	4.12	3.97	2.65	5.36	3.25	5.85	3.75	4.06
12	0.73	0.73	1.58	2.18	2.65	3.72	2.09	2.10	1.99	2.43	0.68	0.23	2.90	1.17	2.42	3.69	3.54	5.80	4.83	6.45	8.05	6.32	4.20	3.97	2.73	5.44	3.33	5.93	3.83	4.14
13	2.17	2.17	1.92	2.52	3.00	4.06	2.99	3.55	3.39	3.87	2.82	2.90	0.20	1.98	4.10	5.13	4.98	7.24	6.99	7.86	9.46	7.73	5.61	5.45	4.13	6.88	4.01	5.86	3.76	4.48
14	1.70	1.70	2.55	3.15	3.53	4.89	3.06	3.07	3.92	4.40	1.85	1.17	1.98	0.25	2.20	4.86	4.71	6.97	4.41	6.34	7.32	7.29	5.17	5.02	3.70	6.61	4.50	7.10	5.00	5.31
15	1.95	1.95	2.80	3.40	3.87	5.14	3.31	3.32	5.17	5.65	2.10	2.42	4.10	2.20	0.63	5.11	4.96	7.22	3.41	5.34	6.32	7.54	5.42	5.27	3.95	6.88	4.75	7.35	5.25	5.56
16	2.96	2.96	3.81	4.41	4.20	4.37	3.09	3.19	1.94	2.20	3.61	3.69	5.13	4.86	5.11	0.55	1.36	2.00	1.89	3.72	4.61	3.18	2.38	3.42	3.84	6.55	3.95	6.05	4.45	5.28
17	2.81	2.81	3.66	4.26	4.05	4.22	2.94	3.04	1.79	0.90	3.46	3.54	4.98	4.71	4.96	1.36	0.44	2.62	3.68	4.00	4.89	3.14	1.20	2.06	2.48	5.19	2.68	5.28	3.18	4.64
18	5.07	5.07	5.92	6.55	6.34	6.48	5.20	5.30	3.05	3.33	5.72	5.80	7.24	6.97	7.22	2.00	2.62	0.33	1.11	1.63	3.22	2.18	2.62	3.66	4.74	7.45	4.94	7.54	5.44	6.90
19	4.85	4.85	5.70	6.30	6.08	6.23	4.95	5.05	2.80	3.08	5.51	4.83	6.99	4.41	3.41	1.89	3.68	1.11	0.40	2.32	3.21	3.19	3.63	4.67	5.75	8.46	5.95	8.55	6.45	7.91
20	5.72	5.72	6.57	7.17	6.95	7.10	5.82	5.92	3.67	3.95	6.37	6.45	7.86	6.34	5.34	3.72	4.00	1.63	2.32	0.35	1.59	1.65	3.27	4.31	5.39	8.10	5.59	8.19	6.09	7.55
21	7.32	7.32	8.97	8.77	8.55	8.70	7.42	7.52	5.20	5.50	7.97	8.05	9.46	7.32	6.32	4.61	4.89	3.22	3.21	1.59	0.59	2.54	4.89	5.93	7.14	9.72	7.31	9.91	7.81	9.17
22	5.59	5.59	6.44	7.04	6.82	6.97	5.69	5.79	4.57	3.82	6.24	6.32	7.73	7.29	7.54	3.18	3.14	2.18	3.19	1.65	2.54	0.70	2.44	3.48	5.26	7.97	5.41	8.06	5.96	7.42
23	3.47	3.47	4.32	4.92	4.70	4.85	3.57	3.67	2.45	1.70	4.12	4.20	5.61	5.17	5.42	2.38	1.20	2.62	3.63	3.27	4.89	2.44	0.63	1.04	3.14	3.67	3.34	4.18	3.84	5.30
24	3.32	3.32	4.17	4.77	4.55	4.70	3.15	2.57	3.40	2.83	3.97	3.97	5.45	5.02	5.27	3.42	2.06	3.66	4.67	4.31	5.93	3.48	1.04	0.38	2.72	2.63	2.92	3.10	3.42	4.88
25	2.00	2.00	2.85	3.45	3.23	3.38	1.83	1.25	2.08	1.51	2.65	2.73	4.13	3.70	3.95	3.84	2.48	4.74	5.75	5.39	7.14	5.26	3.14	2.72	0.35	4.11	1.20	2.40	1.70	3.51
26	4.71	4.71	5.14	5.04	4.97	3.55	4.20	4.05	4.79	4.06	5.36	5.44	6.88	6.61	6.88	6.55	5.19	7.45	8.46	8.10	9.72	7.97	3.67	2.63	4.11	0.90	3.20	2.45	3.20	5.01
27	2.94	2.94	2.69	2.59	2.53	1.10	1.65	1.60	2.21	1.71	3.25	3.33	4.01	4.50	4.75	3.95	2.68	4.94	5.95	5.59	7.31	5.41	3.34	2.92	1.20	3.20	0.41	2.35	0.50	2.31
28	4.09	4.09	4.84	4.74	4.37	2.95	3.50	3.45	4.81	4.31	5.85	5.93	5.86	7.10	7.35	6.05	5.28	7.54	8.55	8.19	9.91	8.06	4.18	3.10	2.40	2.45	2.35	0.50	2.35	2.42
29	2.99	2.99	2.74	2.64	2.27	8.50	1.40	1.35	2.71	2.21	3.75	3.83	3.76	5.00	5.25	4.45	3.18	5.44	6.45	6.09	7.81	5.96	3.84	3.42	1.70	3.20	0.50	2.35	0.31	2.06
30	3.41	3.41	3.16	3.06	1.89	1.42	2.49	2.45	3.54	3.67	4.06	4.14	4.48	5.31	5.56	5.28	4.64	6.90	7.91	7.55	9.17	7.42	5.30	4.88	3.51	5.01	2.31	2.42	2.06	0.33

APPENDIX – C

PREDICTED FLOW VALUES FOR ALL SIZED NETWORKS

The test results for the 3, 6, 9, 12, 15, 18, 21 and 24 sized networks will be presented for back-propagation neural networks, NETDIM, modular and gravity model approaches.

a) Three-node sized network

The test results for three-node sized network are presented below for back-propagation neural networks, NETDIM, modular and gravity model approaches.

The predicted flow values obtained through the simulation of back-propagation neural networks are given below with the RMSE value as 2436.

Table C.1 The predicted flow values for three-node sized network obtained through the simulation of back-propagation neural networks

	28	**29**	**30**	**Pj**
28	10946	1332	1332	**13610**
29	991	723	139	**1853**
30	991	60	404	**1455**
Ai	**12928**	**2115**	**1875**	**16918**

The predicted flow values obtained through testing of the NETDIM network are given below with the RMSE value as 2865.

Table C.2 The predicted flow values for three-node sized network obtained through testing of the NETDIM network

	28	**29**	**30**	**Pj**
28	25018	2182	1269	**28469**
29	4830	2426	188	**7444**
30	4029	200	2457	**6686**
Ai	**33877**	**4808**	**3914**	**42599**

The predicted flow values obtained through testing of the modular network are given below with the RMSE value as 5827.

Table C.3 The predicted flow values obtained for three-node sized network through testing of the modular network

	28	**29**	**30**	**Pj**
28	17961	9085	8590	**35636**
29	10388	3392	3145	**16925**
30	10051	3231	2994	**16276**
Ai	**38400**	**15708**	**14728**	**68836**

The predicted flow values obtained through testing of the unconstrained gravity model are given below with the RMSE value as 5567.

Table C.4 The predicted flow values for three-node sized network obtained through testing of the unconstrained gravity model

	28	**29**	**30**	**Pj**
28	1778	2085	1312	**5175**
29	2190	2568	1617	**6375**
30	1383	1622	1021	**4025**
Ai	**5350**	**6275**	**3950**	**15575**

b) Six-node sized network

The test results for six-node sized network are presented below for back-propagation neural networks, NETDIM, modular and gravity model approaches.

The predicted flow values obtained through the simulation of back-propagation neural networks is given below with the RMSE value as 7185.

Table C.5 The predicted flow values obtained for six-node sized network through the simulation of back-propagation neural networks

	25	**26**	**27**	**28**	**29**	**30**	**P_j**
25	4575	337	6209	7788	1340	186	**20435**
26	583	1696	1108	1421	316	187	**5311**
27	6541	955	8998	23795	772	564	**41625**
28	19090	2061	29577	18603	1174	551	**71056**
29	637	346	1717	902	1799	817	**6218**
30	624	285	577	667	578	1285	**4016**
A_i	**32050**	**5680**	**48186**	**53176**	**5979**	**3590**	**148661**

The predicted flow values obtained through testing of NETDIM network are given below with the RMSE value as 2743.

Table C.6 The predicted flow values obtained for six-node sized network through testing of the NETDIM network

	25	**26**	**27**	**28**	**29**	**30**	**P_j**
25	5416	167	3257	2358	1275	170	**12643**
26	191	2090	450	817	200	31	**3779**
27	3060	386	6795	4629	4177	586	**19633**
28	2317	704	4658	8733	637	500	**17549**
29	866	166	2292	643	2292	518	**6778**
30	113	25	335	359	298	1127	**2257**
A_i	**11962**	**3538**	**17787**	**17540**	**8879**	**2932**	**62639**

The predicted flow values obtained through testing of the modular network are given below with the RMSE value as 3050.

Table C.7 The predicted flow values obtained for six-node sized network through testing of the modular network

	25	**26**	**27**	**28**	**29**	**30**	**P_j**
25	3707	1024	4594	4547	1313	797	**15982**
26	1027	884	1717	2337	518	313	**6797**
27	4616	1721	7332	6486	2306	1407	**23867**
28	4587	2352	6511	10166	2061	1673	**27350**
29	1301	512	2275	2024	712	435	**7258**
30	808	317	1421	1682	445	486	**5159**
A_i	**16046**	**6809**	**23850**	**27243**	**7354**	**5110**	**86413**

The predicted flow values obtained through testing of the unconstrained gravity model are given below with the RMSE value as 4353.

Table C.8 The predicted flow values obtained for six-node sized network through testing of the unconstrained gravity model

	25	**26**	**27**	**28**	**29**	**30**	**P_j**
25	5397	1371	1000	850	508	2375	**11501**
26	1465	2594	716	609	364	1702	**7450**
27	1071	718	1382	445	16	1244	**4876**
28	922	618	451	1135	291	1134	**4551**
29	542	364	16	286	511	856	**2575**
30	2449	1642	1198	1078	826	10807	**18000**
A_i	**11846**	**7307**	**4763**	**4403**	**2516**	**18118**	**48953**

c) Nine-node sized network

The test results for nine-node sized network are presented below for back-propagation neural networks, NETDIM, modular and gravity model approaches.

The predicted flow values obtained through the simulation of back-propagation neural networks are given below with the RMSE value as 2641.

Table C.9 The predicted flow values obtained for nine-node sized network through the simulation of back-propagation neural networks

	22	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
22	1041	244	390	271	147	314	264	142	93	**2906**
23	250	1204	562	409	201	400	405	166	121	**3718**
24	925	2810	2494	1540	1344	2564	4567	954	445	**17643**
25	491	1207	1603	2440	833	3595	5869	2063	1049	**19150**
26	155	220	524	324	887	467	779	229	145	**3730**
27	903	1415	1960	2238	1442	4522	7562	2241	1643	**23926**
28	498	1443	2212	2457	1551	5165	8921	1554	1566	**25367**
29	144	167	386	445	209	683	447	1613	424	**4518**
30	134	187	359	326	212	202	219	840	2646	**5125**
A_i	**4541**	**8897**	**10490**	**10450**	**6826**	**17912**	**29033**	**9802**	**8132**	**106083**

The predicted flow values obtained through testing of NETDIM network are given below with the RMSE value as 1956.

Table C.10 The predicted flow values obtained for nine-node sized network through testing of the NETDIM network

	22	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
22	2206	419	383	333	87	500	714	72	38	**4751**
23	381	2026	1597	403	158	514	606	126	56	**5867**
24	344	2773	5839	1436	661	1951	2507	321	103	**15933**
25	245	409	1480	6277	359	3960	3059	1429	224	**17443**
26	79	168	630	418	2155	697	1096	235	67	**5546**
27	356	447	2005	3964	630	7670	4370	4913	706	**25060**
28	501	446	2619	3072	1068	4419	9453	903	642	**23123**
29	74	124	337	925	206	2413	842	2435	511	**7867**
30	53	62	194	237	78	445	518	286	1117	**2991**
A_i	**4239**	**6875**	**15084**	**17065**	**5402**	**22568**	**23165**	**10718**	**3464**	**108581**

The predicted flow values obtained through testing of the modular network are given below with the RMSE value as 2506.

Table C.11 The predicted flow values obtained for nine-node sized network through testing of the modular network

	22	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
22	463	345	785	658	199	871	770	213	134	**4436**
23	338	403	996	800	323	1053	1182	259	163	**5518**
24	759	982	2767	2161	954	2832	3454	706	445	**15059**
25	653	810	2221	3194	855	3840	4104	964	581	**17221**
26	197	327	979	854	565	1299	1795	342	206	**6564**
27	857	1056	2882	3805	1289	5622	5468	1508	911	**23400**
28	760	1188	3522	4075	1785	5479	8539	1486	1128	**27962**
29	208	258	713	948	337	1498	1473	405	246	**6086**
30	135	168	465	590	209	935	1155	254	244	**4155**
A_i	**4370**	**5537**	**15331**	**17086**	**6517**	**23429**	**27940**	**6136**	**4056**	**110401**

The predicted flow values obtained through testing of the unconstrained gravity model are given below with the RMSE value as 2754.

Table C.12 The predicted flow values obtained for nine-node sized network through testing of the unconstrained gravity model

	22	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
22	7197	1443	902	813	304	2251	592	452	1121	**15075**
23	1491	3109	603	543	203	1504	395	302	749	**8899**
24	966	625	1259	283	131	974	318	391	252	**5199**
25	864	559	281	1015	198	872	285	350	451	**4875**
26	327	212	132	201	629	826	94	277	1	**2699**
27	2290	1481	926	834	781	10652	726	1068	816	**19574**
28	603	390	303	273	89	727	795	1169	376	**4725**
29	484	313	391	353	276	1124	1230	1808	722	**6701**
30	1124	727	236	426	1	805	370	677	2811	**7177**
A_i	**15346**	**8859**	**5033**	**4741**	**2612**	**19735**	**4805**	**6494**	**7299**	**74924**

d) Twelve-node sized network

The test results for twelve-node sized network is presented below for back-propagation neural networks, NETDIM, modular and gravity model approaches.

The predicted flow values obtained through the simulation of back-propagation neural networks are given below with the RMSE value as 22672.

Table C.13 The predicted flow values obtained for twelve-node sized network through the simulation of back-propagation neural networks

	19	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
19	73288	2444	1	314	233	483	391	92	724	679	114	72	**78835**
20	2309	113870	87739	31	306	586	455	101	808	789	127	79	**207200**
21	152	1	616	370	84	238	223	50	305	233	55	41	**2368**
22	313	77	1	661	1054	766	406	94	772	703	116	73	**5036**
23	224	293	95	1044	898	1	1148	223	1663	2014	178	98	**7879**
24	489	492	216	278	127140	233	147	321	107	183	223	252	**130081**
25	502	551	195	475	188	520	169	410	1	1194	127140	26	**131371**
26	92	100	54	95	232	50	650	1	737	668	283	114	**3076**
27	672	709	273	571	1100	11380	127140	1085	220	6608	23988	12608	**186354**
28	684	793	236	699	1176	836	22	30	150	1393	265	24215	**30499**
29	106	117	60	110	173	670	2	273	1	751	4404	1	**6668**
30	65	70	42	67	90	282	477	101	187	245	1183	736	**3545**
A_i	**78896**	**119517**	**89528**	**4715**	**132674**	**16045**	**131230**	**2781**	**5675**	**15460**	**158076**	**38315**	**792912**

The predicted flow values obtained through testing of NETDIM network are given below with the RMSE value as 1264.

Table C.14 The predicted flow values obtained for twelve-node sized network through testing of the NETDIM network

	19	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
19	3824	662	255	386	346	571	644	481	784	1110	377	350	**9791**
20	664	4248	728	1135	385	577	638	482	776	1111	374	349	**11466**
21	251	650	266	297	238	451	517	365	626	845	298	267	**5070**
22	387	1126	312	2635	537	578	595	451	720	1039	349	326	**9055**
23	342	379	235	528	2711	3037	637	343	702	744	305	264	**10226**
24	562	568	443	572	3243	8667	1203	838	1260	1331	554	433	**19674**
25	640	634	514	590	648	1213	9253	582	4514	1826	1679	462	**22554**
26	480	480	364	449	346	830	583	2435	769	1190	368	273	**8566**
27	775	767	620	710	709	1270	4556	772	10420	2113	6163	933	**29808**
28	1103	1103	840	1031	747	1347	1841	1226	2127	10852	1214	962	**24393**
29	373	369	294	343	304	550	1567	362	5553	1152	3170	441	**14479**
30	354	353	270	329	270	445	465	277	891	921	444	1164	**6182**
A_i	**9755**	**11338**	**5141**	**9006**	**10483**	**19536**	**22499**	**8615**	**29141**	**24234**	**15295**	**6223**	**171265**

The predicted flow values obtained through testing of modular network are given below with the RMSE value as 1287.

Table C.15 The predicted flow values obtained for twelve-node sized network through testing of the modular network

	19	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
19	3344	801	307	395	281	191	86	7	84	13	32	9	**5549**
20	803	3606	1074	1299	377	257	115	10	113	17	42	12	**7725**
21	297	1037	1669	481	80	54	22	2	22	3	8	3	**3679**
22	394	1293	496	2528	675	460	121	10	124	18	45	13	**6176**
23	277	371	81	668	2611	2867	600	271	591	347	222	63	**8969**
24	197	264	58	475	2992	7207	1293	938	1274	1239	480	136	**16552**
25	89	120	24	126	634	1308	7774	315	4775	2174	1831	400	**19571**
26	7	10	2	10	274	908	301	2328	691	1361	380	82	**6354**
27	89	119	24	131	632	1304	4832	731	9737	2615	5181	1157	**26552**
28	14	19	4	20	378	1290	2238	1463	2659	11357	1473	1207	**22121**
29	31	42	8	44	220	457	1723	373	4826	1346	3105	716	**12893**
30	9	12	3	12	62	130	377	81	1078	1106	718	2295	**5882**
A_i	**5552**	**7693**	**3749**	**6189**	**9216**	**16432**	**19482**	**6530**	**25973**	**21597**	**13517**	**6092**	**142021**

The predicted flow values obtained through testing of unconstrained gravity model are given below with the RMSE value as 2137.

Table C.16 The predicted flow values obtained for twelve-node sized network through testing of the unconstrained gravity model

	19	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
19	9052	1402	947	712	236	2368	584	541	1171	779	1836	1098	**20726**
20	1465	2823	541	406	135	1352	334	309	668	445	1048	627	**10153**
21	1013	554	1319	285	51	465	227	185	120	231	720	430	**5600**
22	795	434	298	1415	276	492	240	195	268	565	202	120	**5300**
23	270	147	55	282	685	623	109	276	140	78	259	178	**3102**
24	2400	1311	441	447	553	11385	660	964	628	1527	851	509	**21676**
25	609	333	221	224	99	679	864	1262	369	246	152	91	**5149**
26	591	323	188	191	264	1038	1322	1930	556	376	233	139	**7151**
27	1195	653	115	245	125	633	362	520	3395	754	726	178	**8901**
28	814	445	225	529	72	1574	246	360	772	3829	585	350	**9801**
29	1937	1058	710	190	239	885	154	225	750	591	8776	1110	**16625**
30	1225	669	449	120	174	560	97	142	195	374	1175	3319	**8499**
A_i	**21366**	**10152**	**5509**	**5046**	**2909**	**22054**	**5199**	**6909**	**9032**	**9795**	**16563**	**8149**	**122683**

e) Fifteen-node, Eighteen-node, Twenty-one-node and Twenty-four-node sized networks

The test results for fifteen-node, eighteen-node, twenty-one node and twenty-four node sized networks are presented below for back-propagation neural networks, NETDIM, modular and gravity model approaches respectively.

The corresponding RMSE values for the relevant sized networks can be found below the estimated flow matrices.

Table C.17 The predicted flow values obtained for fifteen-node sized network through the simulation of back-propagation neural networks

	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
16	14874	504	896	445	650	14	541	602	3835	3727	490	5854	1466	635	42	**34575**
17	358	3285	20	146	145	1	197	6700	251	429	86	656	778	183	3	**13238**
18	470	9	835	76	59	1	14	6	191	976	2	120	112	61	1	**2933**
19	478	154	1	1863	1260	3	181	408	202	149	1	882	495	6	1	**6084**
20	815	162	16	1202	1272	1	3308	338	293	200	1	864	508	9	1	**8990**
21	133	10	1	1	577	2	1	32	78	124	1	1071	18745	1	1	**20778**
22	751	187	37	182	2567	1	472	635	287	130	1	528	397	32	1	**6208**
23	542	6098	34	432	409	1	612	6485	81	626	272	689	961	114	1	**17357**
24	1040	673	17	526	747	1	423	796	2021	4415	730	1087	1109	345	4	**13934**
25	1170	690	4	172	448	2	39	1519	6178	2699	373	723	1084	426	12	**15539**
26	777	86	1	1	1	1	1	294	453	198	1374	800	695	182	1	**4865**
27	7796	350	15	576	697	7	342	586	7889	6558	476	15695	10095	491	639	**52212**
28	1763	603	50	405	432	2	369	837	2966	2941	386	5572	11115	395	1311	**29147**
29	601	145	3	42	66	1	51	109	191	78	142	2844	692	3588	11	**8564**
30	84	75	1	1	5	1	1	79	587	122	79	167	923	9	111	**2245**
A_i	**31652**	**13031**	**1931**	**6070**	**9335**	**39**	**6552**	**19426**	**25503**	**23372**	**4414**	**37552**	**49175**	**6477**	**2140**	**236669**

RMSE = 2206

Table C.18 The predicted flow values obtained for fifteen-node sized network through testing of the NETDIM network

	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
16	6470	1931	897	1208	464	387	508	848	575	524	486	543	581	426	414	**16261**
17	1911	2751	374	355	352	318	371	1513	887	656	362	626	454	363	319	**11613**
18	892	374	1387	1139	735	252	448	417	383	380	400	403	496	344	361	**8411**
19	1203	355	1155	3232	587	304	377	377	410	437	448	463	554	398	409	**10709**
20	466	352	746	589	3530	621	923	407	411	429	444	455	549	390	404	**10717**
21	389	318	250	303	591	225	318	331	397	431	421	455	520	389	387	**5723**
22	511	372	451	379	920	324	2033	539	430	414	431	438	533	377	391	**8545**
23	840	1495	411	374	402	329	531	2972	2520	503	365	501	459	353	349	**12404**
24	573	886	379	408	409	395	426	2559	6095	717	595	683	657	427	374	**15583**
25	524	655	378	436	428	430	413	506	720	6335	403	2892	1006	1180	378	**16683**
26	486	362	399	448	444	420	430	367	597	404	1767	497	764	361	329	**8076**
27	542	624	401	462	453	454	436	502	684	2893	495	6950	1112	4269	649	**20928**
28	580	453	493	553	547	519	531	460	657	1003	763	1110	7076	795	643	**16182**
29	425	360	342	397	389	387	375	354	427	1156	359	4124	790	2707	426	**13017**
30	416	321	362	410	404	388	392	352	377	381	331	650	647	434	963	**6827**
A_i	**16231**	**11607**	**8425**	**10694**	**10654**	**5751**	**8511**	**12505**	**15571**	**16662**	**8070**	**20788**	**16199**	**13213**	**6795**	**181676**

RMSE = 1465

Table C.19 The predicted flow values obtained for fifteen-node sized network through testing of the modular network

	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
16	5934	2044	1075	1365	305	116	452	954	493	349	28	351	67	157	70	**13761**
17	2007	2812	417	202	157	59	302	1627	976	693	56	646	82	289	76	**10402**
18	1053	416	2425	1487	979	209	591	449	231	95	8	89	11	40	10	**8092**
19	1340	202	1489	3084	645	245	299	227	117	48	4	45	6	20	5	**7776**
20	301	157	984	648	3282	945	1076	309	159	65	5	61	8	27	7	**8036**
21	113	59	208	243	934	1684	405	63	33	12	1	12	1	5	1	**3774**
22	445	303	593	299	1075	409	2221	581	300	69	6	67	8	29	8	**6413**
23	929	1614	446	225	306	64	575	2768	2405	430	211	401	218	179	47	**10816**
24	486	981	233	117	160	33	300	2435	5035	748	614	698	654	312	82	**12888**
25	346	700	96	48	66	12	70	438	752	5213	183	2880	1172	1297	258	**13530**
26	28	56	8	4	5	1	6	213	610	180	1933	421	848	284	56	**4653**
27	349	655	90	45	62	12	68	410	704	2891	428	6003	1348	3808	766	**17640**
28	67	83	11	6	8	1	8	223	661	1179	864	1351	6574	914	762	**12714**
29	152	286	39	20	27	5	28	179	308	1272	282	3722	891	2951	621	**10782**
30	68	76	10	5	7	1	8	48	82	255	56	755	749	626	2289	**5035**
A_i	**13618**	**10445**	**8125**	**7799**	**8017**	**3798**	**6409**	**10922**	**12866**	**13500**	**4679**	**17500**	**12637**	**10937**	**5060**	**146312**

RMSE = 1561

Table C.20 The predicted flow values obtained for fifteen-node sized network through testing of the unconstrained gravity model

	16	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**Pj**
16	10131	1315	763	946	273	2132	663	610	1058	1031	2239	1312	540	666	1222	**24901**
17	1378	2705	403	501	144	1128	351	322	560	545	1185	694	286	352	646	**11200**
18	810	409	922	222	72	663	124	230	179	258	702	411	245	167	462	**5876**
19	999	504	221	1389	163	537	252	187	388	118	353	207	241	77	140	**5776**
20	305	154	76	172	524	703	96	178	115	75	262	153	230	92	93	**3228**
21	2156	1088	631	514	637	12716	565	866	649	351	871	510	386	327	433	**22700**
22	681	344	120	246	88	574	664	1018	363	268	241	141	248	210	96	**5302**
23	660	333	235	191	173	926	1072	1643	587	432	389	228	188	339	155	**7551**
24	1073	542	171	373	105	651	359	550	2922	923	732	429	128	114	208	**9280**
25	1089	550	256	119	71	366	275	422	961	4589	642	441	172	45	1	**9999**
26	2287	1154	675	342	239	880	240	367	737	621	7645	1152	839	585	736	**18499**
27	1387	700	409	207	145	533	145	223	447	441	1192	2716	509	447	523	**10024**
28	624	315	266	264	238	320	279	201	146	188	949	556	2922	385	223	**7876**
29	708	358	167	77	87	344	218	333	119	45	609	450	354	1929	377	**6175**
30	1315	664	468	143	89	461	100	154	220	1	776	533	208	382	1689	**7203**
Ai	**25603**	**11135**	**5783**	**5706**	**3048**	**22934**	**5403**	**7304**	**9451**	**9886**	**18787**	**9933**	**7496**	**6117**	**7004**	**155590**

RMSE = 2007

Table C.21 The predicted flow values obtained for eighteen-node sized network through the simulation of back-propagation neural networks

	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
13	1615	6	1	37	1	1	1	4	9	3	1	129	166	1	210	36	1	1	**2223**
14	31	157	13	67	1	1	1	1	3	1	1	25	124	1	72	501	1	1	**1002**
15	1	2	1102	40	1	1	2	1	4	1	1	81	182	1	39	205	1	1	**1666**
16	72	25	62	23038	733	558	712	873	5	552	1011	2691	2955	162	6245	4	146	10	**39854**
17	14	9	14	906	3769	5	43	48	34	25	403	108	161	25	752	32	28	8	**6384**
18	1	1	1	924	9	374	110	60	1	34	53	188	28	1	43	424	1	1	**2254**
19	11	13	9	981	48	549	3463	376	1	36	67	2820	710	9	195	264	9	11	**9572**
20	13	88	43	1190	74	416	327	3639	13	552	76	1989	1946	7	269	195	19	18	**10874**
21	41	1	1	448	1	1	1	45	1	1	1	71	546	153	6278	149	1	5	**7745**
22	5	5	4	1467	37	23	65	1088	1	3684	154	553	621	7	111	175	3	6	**8009**
23	70	102	73	864	436	24	74	52	181	108	3580	395	360	99	1002	901	126	154	**8601**
24	67	85	73	7093	531	367	650	560	180	480	191	6299	44497	423	9888	37230	458	107	**109179**
25	339	292	380	2786	494	113	1023	974	301	267	413	53856	8377	448	32747	79531	636	242	**183219**
26	4	15	4	57	22	6	7	5	30	9	34	211	823	3299	1008	829	27	23	**6413**
27	156	66	95	3524	603	52	475	462	82	298	1460	6363	75033	647	9828	99897	2345	431	**201817**
28	140	17	27	97	148	16	14	26	9	9	884	1948	1159	865	44748	8194	849	809	**59959**
29	1	9	8	180	33	13	2	3	5	2	18	544	120	33	1816	941	3332	9	**7069**
30	1	1	1	114	1	1	1	1	4	1	1	19	141	1	732	642	3	42	**1707**
A_i	**2582**	**894**	**1911**	**43813**	**6942**	**2521**	**6971**	**8218**	**864**	**6063**	**8349**	**78290**	**137949**	**6182**	**115983**	**230150**	**7986**	**1879**	**667547**

RMSE = 9721

Table C.22 The predicted flow values obtained for eighteen-node sized network through testing of the NETDIM network

	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
13	2188	133	61	320	82	36	68	60	22	51	101	196	260	62	346	300	98	35	**4420**
14	126	1277	99	274	72	33	87	65	22	48	90	174	230	55	253	190	67	30	**3192**
15	62	97	1123	308	79	35	123	86	23	51	100	195	258	60	284	212	73	32	**3203**
16	274	164	256	7344	1393	466	1156	703	48	606	1173	1855	1725	332	2206	1837	484	119	**22139**
17	70	48	66	1335	2267	100	155	145	24	141	826	609	518	100	628	446	145	41	**7663**
18	40	31	39	596	123	1383	529	281	25	147	167	245	199	52	247	185	66	29	**4383**
19	57	53	97	1080	147	754	2671	277	33	154	210	334	271	64	338	251	85	32	**6909**
20	51	43	69	647	139	328	280	2899	274	424	233	367	297	69	371	275	92	34	**6892**
21	29	28	36	224	58	39	84	132	146	77	77	116	95	33	116	88	40	24	**1442**
22	46	36	45	610	145	141	166	419	54	1457	263	367	256	62	322	237	81	32	**4737**
23	76	53	74	1008	851	120	195	214	25	234	2158	1643	538	166	669	713	152	44	**8934**
24	146	94	141	1598	567	147	316	343	31	316	1742	5162	1227	441	1526	1877	335	77	**16085**
25	195	124	189	1517	479	122	264	286	30	226	561	1255	5456	314	2928	2294	724	103	**17067**
26	53	39	51	309	99	40	66	70	22	59	186	481	339	1132	542	847	143	39	**4516**
27	275	143	220	2064	607	160	351	381	33	304	743	1655	3020	533	6699	3179	3399	227	**23992**
28	262	120	182	1860	473	133	286	310	30	246	862	2189	2586	887	3391	8547	871	266	**23502**
29	81	44	60	453	142	48	88	94	23	77	169	367	715	142	2703	825	2538	134	**8704**
30	50	33	42	236	69	33	49	52	21	45	84	168	225	66	394	484	136	925	**3114**
A_i	**4080**	**2560**	**2850**	**21782**	**7793**	**4118**	**6935**	**6818**	**885**	**4663**	**9745**	**17380**	**18214**	**4570**	**23964**	**22787**	**9529**	**2222**	**170893**

RMSE = 1233

Table C.23 The predicted flow values obtained for eighteen-node sized network through testing of the modular network

	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
13	794	307	133	281	109	30	47	32	9	30	98	168	309	46	407	220	186	94	**3300**
14	308	585	273	276	107	29	132	55	20	32	104	177	326	45	283	109	92	56	**3010**
15	132	270	622	276	107	29	233	97	36	32	104	177	326	45	283	109	92	56	**3025**
16	288	281	284	7330	1885	1073	1604	705	270	799	1427	1419	1187	181	1407	681	458	222	**21501**
17	109	107	108	1850	1018	288	254	220	84	289	866	934	781	119	888	342	289	105	**8649**
18	30	29	29	1053	288	616	616	489	136	338	345	343	213	32	242	93	79	29	**5001**
19	47	132	234	1567	253	613	1203	507	194	304	310	308	191	29	217	84	71	26	**6288**
20	32	55	99	694	221	490	511	1249	408	617	369	367	227	35	259	100	84	31	**5846**
21	9	20	36	262	83	135	193	403	363	233	100	99	58	9	67	26	22	8	**2125**
22	30	32	32	787	290	339	306	617	236	842	477	474	214	33	249	94	79	29	**5159**
23	96	101	102	1361	841	335	303	357	98	462	1304	1722	675	276	768	657	250	91	**9800**
24	166	175	177	1376	922	339	306	361	99	467	1751	3726	1319	715	1501	1727	489	178	**15797**
25	311	327	330	1168	782	213	192	227	59	214	696	1338	3914	375	3328	2425	1092	340	**17332**
26	46	45	45	176	118	32	29	34	9	32	281	717	371	824	698	1223	285	88	**5053**
27	410	284	287	1387	893	243	220	259	68	250	794	1526	3337	708	5873	3091	2338	732	**22698**
28	221	109	110	671	344	94	84	100	26	94	679	1756	2431	1240	3089	8679	1269	870	**21867**
29	181	90	90	439	282	77	69	82	21	77	251	483	1063	280	2269	1232	1013	324	**8322**
30	95	57	57	219	106	29	26	31	8	29	94	181	341	90	731	869	334	516	**3811**
A_I	**3305**	**3006**	**3046**	**21173**	**8647**	**5003**	**6328**	**5825**	**2144**	**5139**	**10051**	**15917**	**17283**	**5082**	**22559**	**21759**	**8520**	**3796**	**168584**

RMSE = 1330

Table C.24 The predicted flow values obtained for eighteen-node sized network through testing of the unconstrained gravity model

	13	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**Pj**
13	11077	1401	772	703	358	2031	579	463	1160	1369	2310	1252	861	867	846	1414	813	124	**28400**
14	1458	3069	414	378	192	1090	311	249	623	735	1240	672	462	466	454	759	436	67	**13075**
15	820	423	951	223	108	613	166	216	189	361	573	310	181	229	394	266	98	80	**6201**
16	756	390	226	1186	157	689	187	243	327	237	364	283	292	150	177	165	55	115	**5999**
17	395	204	112	161	479	507	68	156	76	152	362	196	118	66	103	106	64	1	**3326**
18	2119	1093	602	669	481	12565	630	934	975	759	719	354	299	244	410	528	177	368	**23926**
19	598	309	162	180	64	624	700	1039	296	198	274	149	138	113	195	361	121	32	**5553**
20	501	259	220	244	153	969	1088	1615	439	308	426	231	147	175	302	562	188	0	**7827**
21	1172	605	180	308	70	944	290	410	2748	835	537	291	185	138	189	787	264	126	**10079**
22	1377	711	342	221	139	731	193	286	831	4203	950	515	208	94	2	1392	700	106	**13001**
23	2358	1217	550	346	335	703	271	402	543	965	8181	1087	805	426	735	840	281	230	**20275**
24	1324	683	309	278	188	359	152	226	305	541	1126	2490	452	444	591	472	158	129	**10227**
25	929	480	184	293	116	378	144	146	198	223	851	462	2709	252	243	359	120	161	**8248**
26	912	471	227	147	63	246	115	171	143	98	439	442	246	1764	496	294	99	128	**6501**
27	914	471	401	177	101	424	203	302	201	2	777	604	243	510	1319	507	170	227	**7553**
28	1476	762	261	160	100	528	365	542	812	1444	859	465	347	292	489	17940	1269	363	**28474**
29	878	453	100	55	63	183	127	188	282	752	298	161	120	101	170	1315	2084	194	**7524**
30	139	72	84	120	1	395	34	0	139	118	252	136	167	136	235	389	201	607	**3225**
Ai	**29203**	**13073**	**6097**	**5849**	**3168**	**23979**	**5623**	**7588**	**10287**	**13300**	**20538**	**10100**	**7980**	**6467**	**7350**	**28456**	**7298**	**3058**	**209414**

RMSE = 1803

Table C.25 The predicted flow values obtained for twenty-one-node sized network through the simulation of back-propagation neural networks

	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
10	4090	167	251	195	109	44	1147	696	137	190	234	9	225	290	156	293	241	504	433	262	94	**9767**
11	245	7984	291	291	380	320	1269	312	23	71	24	1	24	312	228	175	85	151	347	335	136	**13004**
12	136	682	3010	74	226	76	450	83	5	120	17	1	7	71	154	321	44	400	226	79	32	**6214**
13	26	305	59	1951	52	38	353	71	1	3	1	1	1	283	9	165	1	387	364	55	20	**4146**
14	19	571	606	38	805	28	297	44	1	38	15	1	1	129	81	370	1	379	25	44	2	**3495**
15	300	423	54	35	40	1443	338	66	1	42	80	1	1	273	16	245	1	307	1	51	2	**3720**
16	1158	964	272	62	55	59	31276	692	845	618	347	34	611	1430	1605	810	56	917	247	133	34	**42225**
17	720	52	116	56	44	52	1005	4991	71	114	125	14	137	789	308	279	82	407	338	138	28	**9866**
18	46	6	8	1	1	1	473	32	720	295	227	15	45	217	393	143	1	346	2	31	1	**3004**
19	65	486	115	1	65	108	939	137	275	5033	170	33	155	81	24	399	1	10	50	13	1	**8161**
20	106	512	8	1	3	43	574	152	209	172	5009	104	291	70	24	165	1	10	6	32	1	**7493**
21	2	1	1	1	1	1	164	3	4	27	125	233	20	2	128	1	1	8	1	1	1	**726**
22	1	1	8	1	1	1	519	106	67	93	371	29	3324	118	244	19	1	34	1	31	1	**4971**
23	306	75	220	50	54	62	1072	610	128	187	178	23	246	4677	324	80	202	328	496	217	22	**9557**
24	297	408	310	44	66	55	1209	357	289	176	264	5	363	647	7934	308	301	259	864	365	49	**14570**
25	510	334	309	257	289	283	584	298	100	40	61	1	68	323	294	9814	292	1543	1534	434	251	**17619**
26	74	174	44	1	1	1	54	87	1	1	1	1	1	49	311	62	2478	399	553	112	14	**4419**
27	542	832	445	213	88	69	533	491	46	35	40	3	40	661	724	780	486	9213	2383	483	452	**18559**
28	222	134	42	37	24	27	255	60	20	30	33	1	28	262	1757	1653	814	2159	19533	779	656	**28526**
29	179	36	95	78	28	32	435	120	14	19	41	1	23	64	226	374	121	1234	583	3954	58	**7715**
30	44	326	10	10	2	2	188	14	1	1	1	1	1	22	269	445	17	481	413	74	377	**2699**
A_i	**9088**	**14473**	**6274**	**3397**	**2334**	**2745**	**43134**	**9422**	**2958**	**7305**	**7364**	**512**	**5612**	**10770**	**15209**	**16901**	**5227**	**19476**	**28400**	**7623**	**2232**	**220456**

RMSE = 869

Table C.26 The predicted flow values obtained for twenty-one-node sized network through testing of the NETDIM network

	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
10	2146	821	489	242	189	145	1327	1209	268	401	275	130	266	943	690	1425	246	1455	406	583	212	**13868**
11	827	3342	1639	457	681	652	737	389	148	182	144	96	143	336	464	927	182	753	240	338	218	**12893**
12	481	1553	1132	276	528	328	458	251	126	174	124	93	123	229	313	551	145	463	180	225	157	**7910**
13	246	458	281	919	329	177	235	159	103	112	99	80	100	150	178	276	112	321	178	219	142	**4874**
14	193	661	527	324	625	290	242	160	105	176	121	99	104	158	191	301	114	244	125	150	122	**5035**
15	147	641	332	176	292	684	233	158	103	254	148	109	102	155	186	292	112	236	121	148	122	**4751**
16	1377	751	468	230	237	229	6699	1871	959	1433	511	212	617	1234	909	710	154	754	289	342	186	**20173**
17	1203	392	254	158	159	157	1780	1246	298	266	236	134	299	1013	872	686	159	702	236	309	149	**10707**
18	269	151	127	104	105	104	907	294	562	620	478	144	328	357	304	205	102	208	114	137	104	**5726**
19	402	184	175	112	175	253	1376	265	644	1430	503	194	305	314	260	178	92	179	100	126	96	**7362**
20	278	146	124	99	120	147	508	236	494	506	1473	353	634	369	301	200	97	201	107	135	100	**6627**
21	133	98	94	80	99	110	217	135	142	191	320	88	204	147	136	111	78	110	77	96	83	**2748**
22	269	145	124	100	104	102	609	299	335	306	629	213	922	492	399	197	98	202	108	134	101	**5886**
23	928	333	227	146	154	152	1174	997	353	309	360	143	482	1699	1858	568	282	578	426	271	141	**11582**
24	693	463	313	174	186	181	889	878	301	256	296	131	394	1924	3862	972	615	985	1010	425	178	**15127**
25	1461	936	566	273	299	289	702	694	201	176	197	108	194	583	981	3864	301	2688	1512	1065	285	**17375**
26	248	184	146	112	114	111	156	159	102	93	97	78	98	287	612	302	858	512	862	293	137	**5559**
27	1495	761	471	317	240	232	746	710	203	176	198	107	198	593	997	2704	516	4899	1781	2492	570	**20404**
28	407	240	177	173	122	119	287	232	111	99	106	76	106	432	1023	1520	881	1785	6081	966	614	**15556**
29	571	336	224	215	148	146	337	302	135	126	134	95	133	272	420	1015	288	2310	927	1272	310	**9717**
30	216	223	160	144	123	123	193	151	105	97	101	82	101	145	184	287	139	553	598	312	314	**4353**
A_i	**13990**	**12819**	**8050**	**4829**	**5029**	**4730**	**19815**	**10796**	**5799**	**7383**	**6551**	**2764**	**5853**	**11832**	**15141**	**17290**	**5571**	**20136**	**15478**	**10039**	**4339**	**208235**

RMSE = 1184

Table C.27 The predicted flow values obtained for twenty-one -node sized network through testing of the modular network

	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
10	2800	473	312	77	44	14	789	1367	119	183	81	14	85	735	325	1107	69	1034	103	399	80	**10208**
11	478	3781	1913	245	573	476	251	148	15	22	10	2	10	91	133	454	24	291	29	112	67	**9124**
12	307	1864	2021	157	751	244	161	95	10	29	6	1	7	58	92	292	16	187	19	72	43	**6432**
13	76	241	159	1889	337	48	40	24	2	4	2	0	2	15	22	75	4	95	19	74	30	**3156**
14	43	560	753	334	1592	269	48	28	3	39	6	2	2	21	31	105	5	56	6	22	13	**3939**
15	14	466	246	48	270	1234	40	24	2	105	17	5	2	17	26	88	4	47	5	18	11	**2687**
16	834	262	173	42	52	43	6562	1604	747	1009	181	56	281	702	338	225	12	229	36	88	32	**13508**
17	1366	146	96	24	29	24	1517	1842	203	91	67	21	141	1029	587	391	21	366	36	141	28	**8167**
18	117	15	10	2	3	2	698	200	1449	851	522	84	291	228	109	39	2	37	4	14	3	**4680**
19	183	22	30	4	39	105	953	91	861	2033	337	105	139	109	52	19	1	17	2	7	1	**5109**
20	81	10	6	2	6	17	172	68	530	338	2131	481	591	153	73	26	1	25	2	9	2	**4724**
21	13	2	1	0	2	5	52	20	83	102	467	856	179	23	11	4	0	3	0	1	0	**1825**
22	85	10	7	2	2	2	265	141	294	139	588	183	1336	309	149	28	1	27	3	10	2	**3584**
23	727	89	58	15	21	17	657	1018	228	108	151	24	306	1970	1713	237	98	221	114	85	17	**7873**
24	330	134	95	22	32	26	324	596	112	53	74	12	151	1757	4130	460	340	430	413	165	33	**9691**
25	1133	459	303	77	109	90	218	401	41	19	27	4	28	245	464	4186	84	2141	794	828	120	**11773**
26	68	24	16	4	5	4	11	21	2	1	1	0	1	98	333	82	1129	217	500	134	19	**2671**
27	1067	297	196	98	58	49	224	377	38	18	25	4	28	231	437	2159	225	5049	945	2879	422	**14829**
28	107	30	20	20	6	5	36	38	4	2	3	0	3	120	424	807	522	953	6034	587	436	**10155**
29	392	109	72	73	21	18	82	138	14	7	9	1	10	85	160	796	132	2747	555	2025	313	**7760**
30	79	65	43	30	13	11	30	28	3	1	2	0	2	17	32	116	19	404	414	314	1261	**2885**
A_i	**10302**	**9058**	**6529**	**3164**	**3964**	**2703**	**13128**	**8269**	**4760**	**5153**	**4707**	**1854**	**3597**	**8014**	**9643**	**11697**	**2710**	**14576**	**10032**	**7984**	**2935**	**144779**

RMSE = 1225

Table C.28 The predicted flow values obtained for twenty-one-node sized network through testing of the unconstrained gravity model

	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_i**
10	12635	1291	753	825	266	1793	582	616	1173	1076	2081	1167	760	691	826	2015	567	173	458	350	29	**30127**
11	1363	3567	411	450	145	979	318	336	640	587	1136	637	415	377	451	1100	310	94	250	191	16	**13773**
12	812	420	1240	268	102	452	135	146	233	256	720	287	287	164	196	307	93	62	126	90	32	**6428**
13	884	457	267	1308	161	463	138	150	203	338	451	253	215	217	101	179	65	50	74	71	33	**6078**
14	300	155	107	169	531	582	90	126	80	221	250	140	181	55	79	117	75	24	49	46	2	**3379**
15	1895	980	443	457	546	13562	620	922	764	1080	616	345	237	312	286	573	208	160	238	226	106	**24576**
16	597	309	128	132	82	601	623	927	264	242	304	171	187	147	68	453	165	39	103	79	7	**5628**
17	653	338	144	148	119	926	959	1427	406	372	468	263	112	226	105	697	253	60	122	121	10	**7929**
18	1202	621	222	194	73	741	263	392	2529	644	713	343	147	138	121	968	352	227	358	163	15	**10426**
19	1114	576	246	326	203	1058	244	363	651	4250	799	380	230	103	174	1085	606	115	602	255	22	**13402**
20	2142	1107	687	433	228	599	305	454	716	794	9035	1001	554	593	609	673	245	163	332	173	8	**20851**
21	1248	645	285	252	133	349	178	265	359	393	1040	2951	416	486	302	392	143	95	229	139	5	**10305**
22	829	429	291	219	175	428	199	116	156	242	587	425	3021	304	186	354	168	59	120	44	1	**8353**
23	724	374	160	212	51	309	150	223	141	104	604	476	292	1776	272	347	126	44	104	77	11	**6577**
24	924	478	204	105	79	302	74	111	133	188	662	315	191	291	2706	340	161	56	195	156	10	**7681**
25	2070	1070	292	172	107	557	454	676	970	1076	672	377	334	341	312	16642	1270	532	1259	418	51	**29652**
26	624	322	95	67	73	216	176	263	377	643	261	146	169	132	158	1358	2350	176	180	195	20	**8001**
27	206	107	68	56	25	181	45	67	265	133	189	106	64	50	60	618	191	1197	512	329	33	**4502**
28	527	272	134	80	49	258	115	132	402	668	370	246	126	115	200	1408	188	492	4774	554	64	**11174**
29	466	241	111	88	54	284	102	152	212	328	224	173	54	98	185	542	236	367	643	4525	71	**9156**
30	34	17	34	35	2	115	7	11	17	24	9	5	2	12	11	57	21	31	64	61	43	**612**
A_{ii}	**31249**	**13776**	**6322**	**5996**	**3204**	**24755**	**5777**	**7875**	**10691**	**13659**	**21191**	**10207**	**7994**	**6628**	**7408**	**30225**	**7793**	**4216**	**10792**	**8263**	**589**	**238610**

RMSE = 1611

Table C.29 The predicted flow values obtained for twenty-four-node sized network through the simulation of back-propagation neural networks

	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
7	621	508	403	79	279	56	1	1	1	165	2	1	1	3	25	2	1	16	391	1	399	26	452	4	**3438**
8	223	3620	344	54	630	160	2	5	3	331	14	4	29	72	28	32	22	461	394	4	513	88	845	28	**7906**
9	253	822	4320	1449	619	262	11	3	12	768	261	13	52	31	1	21	50	320	756	41	560	2	59	2	**10688**
10	163	247	1731	4218	260	162	48	14	11	810	1032	30	208	236	1	106	108	375	330	131	199	44	172	9	**10645**
11	279	361	504	271	9488	220	222	273	233	278	640	1	57	70	1	21	221	74	9	32	9	436	521	4	**14225**
12	127	124	74	29	212	3208	8	491	51	68	4	9	10	79	21	44	10	9	269	16	36	1	4	1	**4905**
13	2	2	4	3	85	7	2235	167	1	1	3	37	82	72	2	54	17	2	3	56	1	1	2	1	**2840**
14	1	2	1	1	401	924	115	1004	39	46	1	1	17	1	15	32	34	1	1	1	18	1	1	1	**2659**
15	1	2	4	12	400	30	1	64	1175	1	2	34	3	5	10	56	8	1	2	47	1	1	4	2	**1866**
16	264	391	823	599	281	231	81	34	63	42874	997	737	895	381	5	385	614	305	151	58	67	7	445	4	**50692**
17	9	18	132	622	294	7	7	2	6	715	4784	33	15	14	1	14	309	742	635	32	358	1	18	1	**8769**
18	1	1	2	1	1	3	19	13	18	591	8	587	635	217	1	43	4	1	1	35	1	1	1	13	**2198**
19	3	33	42	70	40	20	34	2	7	755	18	280	4713	78	1	19	82	68	15	32	2	1	22	1038	**7375**
20	11	67	26	92	7	65	23	24	13	101	16	361	69	4599	32	576	64	94	32	37	3	1	109	11	**6433**
21	5	6	1	1	1	11	29	4	1	1	1	1	1	267	3	1	1	1	1	46	1	1	10	28	**423**
22	7	31	12	16	8	49	31	36	34	319	10	119	11	264	6	3497	31	40	20	42	2	1	55	33	**4674**
23	14	93	119	100	1162	62	17	8	14	890	751	62	194	160	2	151	2851	129	774	110	303	43	146	3	**8158**
24	46	405	619	197	69	226	9	4	8	200	605	18	237	375	1	294	398	7594	5	451	2	33	539	1	**12336**
25	530	404	683	442	12	353	57	29	54	387	589	3	49	54	1	27	217	3	8107	275	251	50	541	16	**13134**
26	1	4	29	30	39	21	48	35	44	1	29	25	32	42	1	34	26	424	55	3201	82	577	12	1	**4793**
27	872	599	647	232	52	282	94	14	36	176	559	5	63	84	1	47	367	13	226	421	10101	147	188	539	**15765**
28	78	362	996	801	56	70	27	6	14	28	433	5	42	47	1	28	791	133	567	848	671	18988	902	402	**26296**
29	292	1196	31	42	166	4	2	3	8	3	14	6	114	100	14	48	35	94	679	12	240	613	4100	91	**7907**
30	4	7	1	1	1	1	1	1	1	1	1	5	38	22	29	12	1	1	1	1	192	287	32	148	**789**
A_i	**3807**	**9305**	**11548**	**9362**	**14563**	**6434**	**3122**	**2237**	**1847**	**49510**	**10774**	**2377**	**7567**	**7273**	**203**	**5544**	**6262**	**10901**	**13424**	**5930**	**14012**	**21351**	**9180**	**2381**	**228914**

RMSE = 1181

Table C.30 The predicted flow values obtained for twenty-four-node sized network through testing of the NETDIM network

	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
7	586	834	637	609	710	402	281	254	255	615	327	176	207	178	134	176	318	478	818	232	1009	494	578	247	**10554**
8	878	1344	839	773	920	540	298	320	315	730	400	189	227	188	135	187	372	774	1436	282	1350	620	823	338	**14278**
9	662	842	1695	1636	1109	640	342	273	211	1512	756	346	503	357	183	255	675	582	1053	246	1101	379	505	273	**16137**
10	629	779	1652	2369	992	605	324	261	201	1497	1376	354	505	356	183	350	1113	853	1616	330	1627	510	720	291	**19492**
11	737	934	1124	998	3631	1840	581	844	800	873	499	205	248	197	133	197	438	594	1094	250	893	314	444	301	**18169**
12	400	526	622	589	1733	1222	352	623	408	561	325	173	234	169	127	169	303	409	665	199	564	241	298	215	**11128**
13	281	297	341	323	573	356	1004	410	238	307	215	142	154	136	108	137	204	243	358	155	406	239	291	197	**7116**
14	254	316	272	261	810	622	406	722	368	319	218	145	238	166	136	144	217	262	391	158	321	172	207	170	**7296**
15	255	312	211	202	775	412	237	369	767	305	214	142	327	201	151	140	211	253	376	154	308	166	203	169	**6858**
16	639	749	1584	1555	892	581	307	320	306	6472	2050	1127	1595	618	296	743	1402	1066	831	210	864	362	440	258	**25267**
17	331	398	753	1355	498	330	215	218	214	1930	1362	377	343	306	187	379	1154	1038	813	217	822	308	397	206	**14151**
18	176	189	341	349	207	174	142	145	142	1044	369	611	704	551	196	400	446	398	272	141	274	156	189	143	**7762**
19	206	226	502	504	249	235	154	238	328	1515	342	737	1536	594	262	385	403	344	239	126	238	136	173	132	**9802**
20	177	188	358	358	199	170	136	166	201	608	307	576	598	1529	432	723	460	390	264	133	263	146	185	138	**8702**
21	134	136	183	183	134	128	109	136	152	291	186	192	254	387	113	264	202	189	152	105	151	104	132	112	**4131**
22	176	187	256	350	199	170	137	143	140	722	379	410	386	716	280	1010	597	507	262	134	266	147	184	139	**7894**
23	314	366	666	1093	434	302	202	214	209	1332	1147	450	398	452	201	591	1893	2108	695	372	699	530	359	196	**15226**
24	486	785	586	861	596	415	241	260	251	1047	1062	404	343	388	187	510	2193	4241	1156	776	1154	1182	554	251	**19930**
25	864	1491	1077	1653	1105	691	360	396	379	821	831	273	238	263	151	261	714	1161	4005	394	2821	1664	1241	382	**23234**
26	231	280	246	330	251	200	154	157	154	211	217	140	127	133	105	134	376	761	391	996	626	1010	383	191	**7803**
27	1079	1407	1134	1676	906	582	410	322	309	858	843	274	237	262	149	265	719	1166	2848	639	4876	1911	2687	713	**26273**
28	502	626	379	512	315	240	236	169	163	359	306	154	135	144	103	145	540	1191	1673	1042	1909	5964	1134	770	**18712**
29	578	799	493	699	437	297	287	204	201	429	389	187	173	183	131	183	358	540	1174	376	2474	1077	1429	399	**13497**
30	243	327	270	289	300	217	196	171	169	259	206	144	132	138	112	139	199	253	373	191	669	726	395	363	**6481**
A_i	**10816**	**14336**	**16219**	**19527**	**17976**	**11370**	**7111**	**7339**	**6880**	**24615**	**14328**	**7928**	**9841**	**8611**	**4197**	**7886**	**15508**	**19801**	**22958**	**7859**	**25687**	**18557**	**13951**	**6595**	**319895**

RMSE = 1004

Table C.31 The predicted flow values obtained for twenty-four-node sized network through testing of the modular network

	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**P_j**
7	1830	1771	704	468	457	297	110	97	78	203	127	10	16	6	1	7	75	149	578	33	772	125	643	163	**8719**
8	1788	2493	740	492	528	343	71	112	90	213	134	11	17	7	1	7	79	321	1240	45	950	154	791	199	**10823**
9	712	741	2603	1963	666	415	91	49	14	853	535	121	190	75	11	28	305	145	559	22	540	40	204	68	**10951**
10	474	494	1970	3317	444	288	60	33	9	716	1501	99	156	62	9	67	740	290	1121	53	1006	74	381	65	**13431**
11	467	535	674	447	4452	2148	217	571	457	194	122	10	14	6	1	6	69	104	403	16	238	17	90	52	**11310**
12	295	338	408	283	2090	2368	137	801	224	122	77	6	20	4	1	4	44	72	255	10	150	11	57	33	**7810**
13	110	71	90	60	213	138	2236	329	37	26	16	1	2	1	0	1	10	14	56	2	70	11	59	22	**3575**
14	97	111	49	32	557	805	327	1908	257	32	20	2	29	4	1	1	14	22	83	3	40	3	15	9	**4421**
15	78	89	14	9	447	226	37	257	1391	26	16	1	86	11	3	1	11	17	67	2	32	2	12	7	**2842**
16	217	226	902	755	203	132	28	35	28	6830	1589	690	937	136	39	226	622	268	167	7	164	20	62	21	**14302**
17	128	133	534	1493	120	78	16	21	16	1500	2109	182	72	51	15	119	1085	566	352	14	315	23	119	20	**9082**
18	10	10	119	98	9	6	1	2	1	643	180	1688	902	517	72	274	205	88	27	1	25	2	9	2	**4890**
19	16	17	190	155	14	21	2	29	86	884	72	912	2319	307	89	116	87	37	12	0	10	1	4	1	**5379**
20	6	7	76	62	6	4	1	4	11	128	51	525	308	2403	484	581	126	54	17	1	15	1	6	1	**4877**
21	1	1	11	9	1	0	0	1	3	36	14	71	86	470	990	163	16	7	2	0	2	0	1	0	**1885**
22	7	7	28	67	6	4	1	1	1	213	119	277	116	578	167	1477	283	122	18	1	17	1	6	1	**3516**
23	75	78	302	731	68	44	10	14	11	583	1077	206	86	125	17	281	2229	1865	199	79	178	84	67	12	**8420**
24	154	328	148	295	105	74	15	22	18	259	578	91	38	55	7	125	1920	4893	410	312	367	344	139	24	**10722**
25	598	1270	572	1144	408	265	58	86	69	161	361	28	12	17	2	19	205	411	4729	64	2159	696	820	98	**14252**
26	33	45	22	53	16	10	2	3	2	6	14	1	0	1	0	1	79	304	62	1249	178	444	114	14	**2655**
27	807	984	558	1037	243	158	73	42	33	161	327	26	11	16	2	18	186	372	2183	185	5479	820	3215	386	**17321**
28	132	160	41	77	18	12	12	3	2	20	24	2	1	1	0	1	88	351	707	464	824	6407	528	391	**10265**
29	639	778	201	373	87	57	58	15	12	58	117	9	4	6	1	6	67	134	788	113	3055	499	2393	305	**9774**
30	162	195	67	64	50	33	22	9	7	19	20	2	1	1	0	1	11	23	94	13	366	370	305	1495	**3330**
A_i	**8835**	**10879**	**11023**	**13482**	**11208**	**7924**	**3585**	**4443**	**2859**	**13885**	**9202**	**4971**	**5423**	**4857**	**1912**	**3528**	**8557**	**10629**	**14127**	**2690**	**16954**	**10151**	**10042**	**3386**	**194550**

RMSE = 1048

Table C.32 The predicted flow values obtained for twenty-four-node sized network through testing of the unconstrained gravity model

	7	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**Pi**
7	11921	1722	980	713	352	1803	566	430	1141	1284	2905	1555	869	863	760	1238	743	166	367	302	35	242	601	668	**32226**
8	1802	3025	505	368	181	929	291	222	588	662	1497	801	448	445	392	638	383	85	189	155	18	124	310	344	**14402**
9	1053	519	1005	213	106	543	170	179	188	290	579	310	185	195	317	204	106	46	102	82	30	104	108	120	**6754**
10	777	383	216	1232	153	605	190	200	289	196	350	265	285	132	167	193	70	47	48	97	33	62	154	84	**6228**
11	391	192	109	155	519	529	67	79	63	128	317	170	113	67	77	126	54	14	32	32	2	41	49	55	**3381**
12	1926	949	540	593	509	13728	622	874	979	648	756	285	166	259	421	637	232	154	160	163	109	206	509	276	**25701**
13	599	295	168	184	64	616	752	1057	298	180	269	144	133	101	129	320	117	43	95	78	9	99	155	221	**6126**
14	474	233	184	202	79	900	1099	1544	435	262	393	211	152	148	188	467	170	37	139	114	13	144	160	533	**8281**
15	1187	585	182	276	59	953	292	411	3059	1001	538	288	207	111	196	761	277	144	526	155	20	104	673	345	**12350**
16	1336	658	281	187	120	631	177	248	1002	3811	851	456	163	105	203	1204	863	125	356	173	13	219	979	814	**14975**
17	2919	1438	542	322	287	711	256	359	520	822	8267	1298	709	390	635	805	293	128	180	107	37	135	299	392	**21851**
18	1627	802	302	255	160	279	142	200	290	458	1352	2467	240	398	501	316	164	71	171	127	21	75	167	218	**10803**
19	945	466	187	284	111	270	137	150	217	170	767	249	3100	249	158	391	143	78	88	26	1	32	226	159	**8604**
20	917	452	193	128	64	257	102	143	113	107	412	404	243	1759	427	291	52	30	114	51	14	145	182	202	**6802**
21	829	408	322	167	76	430	132	186	206	212	689	522	158	438	1401	486	177	97	318	126	10	40	178	197	**7805**
22	1335	658	205	191	123	643	326	458	790	1249	864	325	388	296	481	17753	1257	421	932	381	24	879	1650	846	**32475**
23	828	408	110	72	54	242	123	172	297	924	325	174	146	54	181	1298	2433	219	161	143	19	331	802	486	**10002**
24	204	100	53	53	16	178	50	41	170	148	157	84	89	34	110	480	242	1028	439	393	22	381	414	217	**5103**
25	448	220	116	54	35	182	109	154	616	417	218	199	98	130	356	1053	177	435	4996	661	47	496	430	178	**11825**
26	407	201	104	120	39	206	99	140	202	225	143	164	32	64	156	477	174	431	733	4366	103	1073	543	280	**10482**
27	39	19	31	34	2	113	10	13	22	14	41	22	2	15	10	25	19	20	43	85	15	80	32	33	**739**
28	301	148	121	71	46	240	116	163	125	262	167	89	37	170	46	1014	369	386	506	989	90	1886	1067	547	**8956**
29	674	332	112	158	50	534	164	162	724	1054	333	178	232	192	183	1713	807	377	395	450	32	960	2102	1533	**13451**
30	784	386	131	90	58	304	245	569	389	919	458	245	171	223	212	921	512	208	171	243	35	516	1607	9256	**18653**
Aj	**33723**	**14599**	**6699**	**6122**	**3263**	**25826**	**6237**	**8154**	**12723**	**15443**	**22648**	**10905**	**8366**	**6838**	**7707**	**32811**	**9834**	**4790**	**11261**	**9499**	**752**	**8374**	**13397**	**18004**	**297975**

RMSE = 1672

VITA

Serkan Tapkın was born in Ankara, Turkey on 7 October 1973. He completed his secondary education in 1990 at T.E.D. Ankara College. He graduated in 1994 from Middle East Technical University with the B.Sc. degree in Civil Engineering followed by the M.Sc. degree in 1998. He has been a research and teaching assistant in Middle East Technical University since 1996. He is of Turkish nationality.