PB2000-101740

ROAD SURFACE CRACK CONDITION FORECASTING USING NEURAL NETWORK MODELS

Written by:

Zhenyu Lou
Research Assistant

Jian John Lu, Ph.D., P.E.
Associate Professor

and

Manjriker Gunaratne, Ph.D., P.E.
Associate Professor

Transportation Program
Development of Civil and Environmental Engineering, University of South Florida
Tampa, Florida 33620
(813) 974-5817

Presented to:
Florida Department of Transportation

A Project Sponsored by:
Florida Department of Transportation

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
0510816		

4. Title and Subtitle	5. Report Date
Road Surface Crack Condition Forecasting Using Neural Network Models	28 October 1999
	6. Performing Organization Code
7. Author's	8. Performing Organization Report No.
Zhenyu Lou, Jian John Lu, and Manjriker Gunaratne	
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Transportation Program Department of Civil and Environmental Engineering University of South Florida 4202 E. Fowler Avenue, ENB 118 Tampa, FL 33620	11. Contract or Grant No. BB275
	13. Type of Report and Period Covered Final, Fall 1997 to Fall 1999
12. Sponsoring Agency Name and Address Florida Department of Transportation 605 Suwannee Street Tallahassee, FL 32399-0450	14. Sponsoring Agency Code

15. Supplementary Notes

Prepared in cooperation with the US Department of Transportation and Federal Highway Administration

16. Abstract

Accurate forecasting of pavement crack condition is essential for pavement management systems (PMS) at either network or project level. Up to now, mechanistic-empirical and purely empirical models have been used to forecast pavement crack condition. A characteristic feature of these models is that they are formulated based on laboratory and/or field statistical data. Hence, selection of appropriate function forms could be difficult with a large data dimension. This report summarizes the results obtained from a research project sponsored by Florida Department of Transportation to develop a Backpropagation Neural Network (BPNN) model for the forecasting of pavement crack condition of Florida's highway network. The BPNN model, which is able to learn the hidden information from the historical crack condition data, has the capability to forecast future crack condition. In order to setup an effective model, the concept of BPNN was introduced along with its mathematical training algorithm. The neural network model was then trained and tested with field data collected from Florida's highway network. Further, the BPNN model was compared with a commonly used autoregressive (AR) model. Finally, a validation step was performed to identify the forecasting errors on the 1998 data set. It was concluded that the BPNN model was more accurate than the AR model and could be applied to forecast pavement crack condition.

17. Key Words	18. Distribution Statement
Crack, Forecasting, Pavement performance, Neural network, Surface distress	No restriction This report is available to the public through the National Technical Information Service, Springfield, VA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	83	

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Accurate forecasting of pavement crack condition is essential for pavement management systems (PMS) at either network or project level. Up to now, mechanistic-empirical and purely empirical models have been used to forecast pavement crack condition. A characteristic feature of these models is that they are formulated based on laboratory and/or field statistical data. Hence, selection of appropriate function forms could be difficult with a large data dimension. This report summarizes the results obtained from a research project sponsored by Florida Department of Transportation to develop a Backpropagation Neural Network (BPNN) model for the forecasting of pavement crack condition of Florida's highway network. The BPNN model, which is able to learn the hidden information from the historical crack condition data, has the capability to forecast future crack condition. In order to setup an effective model, the concept of BPNN was introduced along with its mathematical training algorithm. The neural network model was then trained and tested with field data collected from Florida's highway network. Further, the BPNN model was compared with a commonly used autoregressive (AR) model. Finally, a validation step was performed to identify the forecasting errors on the 1998 data set. It was concluded that the BPNN model was more accurate than the AR model and could be applied to forecast pavement crack condition.

ACKNOWLEDGEMENT

This project was sponsored by Florida Department of Transportation (FDOT). Project results were based on the database provided by FDOT. The support and assistance provided by FDOT are deeply appreciated.

The authors would like to specifically express appreciation to Mr. Bruce Dietrich, P.E., FDOT State Pavement Design Engineer and project manager of this research project, for his technical support and suggestions.

CHAPTER 1

INTRODUCTION

Road surface performance models, which provide forecasting of pavement condition over time, constitute a key component of a Pavement Management System (PMS). Development of pavement performance forecasting models has been an area of thrust in the last couple decades. However, pavement performance has been so complicated that it is difficult to select an appropriate functional form as required in traditional modeling. The main objective of the research was to develop a new pavement performance model based on the neural network algorithm. The following sections describe background information, problem statement, and study scope.

1.1 Background

In 1998, the Transportation Equity Act in the 21st Century (TEA-21) was signed into law calling for coordinated efforts to collect, manage, analyze, and store massive quantities of transportation related data. TEA-21 requires traffic monitoring and systems management monitoring of pavement, bridge, safety, congestion, public transportation facilities, and intermodal facilities. Development of these new management systems open the door for many advanced technologies and resources to be applied for state-of-the-art information storage, retrieval, and management processes. Among all these management systems, PMS has become a very useful management tool for highway agencies due to the increasingly challenging tasks of pavement maintenance. Today's transportation system includes marine, highway, air, rail, and pipeline transportation. Of these, only marine and pipeline transportation do not make use of pavements. In the highway system, pavements are the major structural load-carrying elements. The expenditures in the highway sector in the United States represent the largest amount in U.S. transportation and exceed $20 billion annually, within which pavements represent approximately one-half of this total highway expenditure. Although the function of pavement varies with the specific user, the purpose of the pavement is to serve traffic safely, comfortably, and efficiently. With the relatively large investments involved in pavements, even marginal improvements in managing this investment, and in the technology involved, may effect very large absolute dollar savings. In addition to the direct savings in capital costs and maintenance, the indirect benefits to the road user can be equally significant, although much more difficult to ascertain. Accordingly, the cost-effective management of the pavements is one of the most significant reasons for using PMS.

Pavement management typically operates at two major levels, network and project levels. The network level management has the development of a priority program and schedule of work within overall budget constraints as its primary purpose. Project level work thus comes on stream at the appropriate time in the schedule, and represents the actual physical implementation of network decisions. Most PMS consists of a few basic components: inventory, analysis, output, and feedback, as shown in Figure 1-1. Although the complexity of these components can vary dramatically, pavement performance models comprise the nucleus of functions of the analysis subsystem. The essential function of the PMS analysis subsystem is to consider the pavement improvement and maintenance needs and to arrive at a program of rehabilitation, new construction, and maintenance. This is accomplished through the following steps:

- Identification of needs and candidate projects for performance improvement,
- Generation of alternatives for each candidate project, and
- Technical and economic analysis of each alternative in terms of future performance and life-cycle cost and benefits.

In order to identify future improvement needs and perform technical-economic analysis for each alternative, application of deterioration or performance forecasting models are required. To be fully functional, the PMS must have performance models that relate the change in the pavement performance with time.

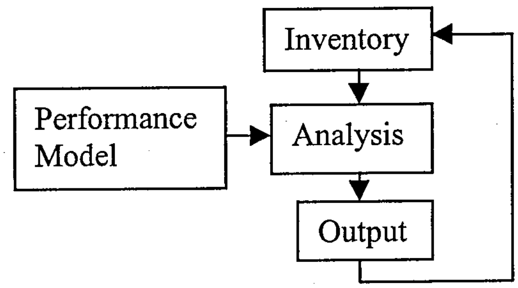


Figure 1-1. PMS Architecture

The concepts of pavement performance developed in the late 1950s and early 1960s in association with the AASHTO Road Test provided the necessary system output function for PMS. The serviceability–performance concept was defined as "The history of deterioration of the ride quality or serviceability provided to the user." However, the performance concept is so generic that loosely speaking, it can include roughness, distress, skid resistance, and structural adequacy. Roughness is derived from the longitudinal profile of the pavement surface which affects ride comfort or quality. Distress is the physical deterioration of the pavement surface, such as cracking, rutting et al. Skid resistance is primarily related to the surface friction and road safety issues. Structural adequacy is the ability of the pavement to carry loads without resulting in undue distress. In a complete sense, a good pavement provides satisfactory riding comfort to its users, does not require extensive maintenance for the repair of distress, is structural adequate for the traffic loads, and provides sufficient friction to avoid skidding accidents.

Two aspects of information on pavement performance are used in decision-making process: information on current performance, which is obtained through field inspection, and information on future performance, which is obtained using deterioration models. Definition of the functional behavior of the pavement as the ability to provide a smooth, comfortable, and safe ride requires the development of a rating method to characterize these attributes, which depend on the user's perception of the pavement condition. Thus, user opinions must be measured in order to rate the serviceability of the pavement. However, many arbitrary scales are used in a wide variety of applications. Development of a scale for rating pavements is complicated by the interaction between the vehicle parameters, staff experiences, and the roadway characteristics.

To evaluate pavement surface distresses, most highway agencies conduct periodic surface distress surveys of their pavements. Such surveys are directed in large part toward assessing the maintenance measures needed to prevent accelerated future distress. Distress surveys should typically identify the distress type, severity, and extent, and in some cases the location. Pavement distress survey may be performed by walking along the pavement section or from a moving vehicle. Although walking surveys provide the most precise data about the pavement condition, they are time consuming and it is not possible to survey all the road sections in a highway network. Consequently, highway agencies usually use walking surveys to define the sampling methodology with respect to site and selection of sample sections. Random selection of pavement samples for rating will produce a good assessment of the overall condition of the highway network, provided the sample size is adequate.

In order to forecast when the sections in a pavement network need to be maintained it is necessary to predict the rate of change of those measures for which criteria have been established. Figure 1-2 is a schematic illustration of how deterioration forecasting would be applied to an existing pavement section to estimate the rate of future deterioration and the needs year. It can be seen that pavement deterioration models provide the base inputs for pavement maintenance and rehabilitation planning. Knowledge of pavement performance characteristics allow several activities to occur:

- Determination of the action year in which a road section deteriorates to the minimum acceptable level,
- Forecasting of the future funding requirement to maintain the pavement network at acceptable level,
- Justification for annual rehabilitation budget, and
- Justification for project rehabilitation.

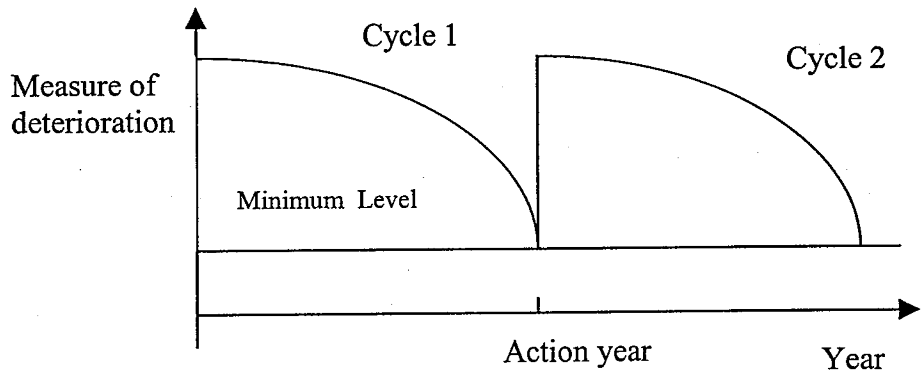


Figure 1-2. A Schematic Illustration of Performance Models

In addition to the applications in the management and maintenance scope, from pavement design perspective, pavement performance forecasting model is an important factor because it provides a framework upon which a judgement on the success or failure of a design procedure can be made. From the mix-design perspective, performance-based approach produces a mixture which can be evaluated to ensure that it will attain the desired level of performance in the specific pavement section in which it is to function.

1.2 Problems of Pavement Performance Forecasting Models

Predicting the future condition of a pavement is one of the most difficult problems in a PMS. Models based on mechanistic characteristics have been slow to emerge. On the other hand, it is often relatively easy to collect data from in-service pavement. Models based on field data, often referred to as empirical model can be very useful in situations where data are plentiful but the physical principles underlying the observations have not yet yielded a concise mathematical description. The general empirical performance model can be expressed as:

$$PPI = f(x, \theta)$$

where:

PPI = pavement performance index (PSI, PCI, Crack Index et al.),

x = a vector of explanatory variables affecting pavement performance, and

θ = parameters to be estimated.

The basic requirements for this type of empirical forecasting model should include the following:

- An adequate database,
- Inclusion of all significant explanatory variables, and
- Appropriate function form to represent the real-world situation.

A characteristic feature of the empirical models is that they are formulated and estimated statistically from field data. That is to say, the function $f(x,\theta)$ used in the equation is implicit in the data and can be established from those data. One typically selects the form of function $f(x,\theta)$ and determines the parameters through a process of minimizing the error between the output of the function and the observations available. However, when the dimension of the data domain is large, the selection of an appropriate function form can be very difficult. To date, modeling pavement performance has been extremely complicated, no PMS can consider more than a few of the parameters involved, and then only in highly simplified manner. Therefore, research into other modeling methods has been advocated which are capable of making generalizations and of offering solutions to complex forecasting problems.

1.3 FDOT's Current Practice

In FDOT's current practice, pavement crack condition survey is conducted by visual inspection. The survey crew drives at a reduced speed to look at entire section length to be rated and records the average condition for the rated section. Three different types of cracking have been defined by FDOT *(1)*:

Class 1B – Hairline cracks that are 1/8 inch wide and either in the longitudinal or transverse direction.

Class II – 1/8 inch to ¼ inch wide cracks either longitudinal or transverse. Also includes all cracks less than ¼ inch wide which have formed cells less than 2 feet on the longest side.

Class III – ¼ inch wide or greater and extends in a longitudinal or transverse direction and cracks which are opened to the base or underlying material. Also includes progressive Class II cracking resulting in severe spalling with chunks of pavement breaking out.

The crack condition is estimated as the percent confined to wheelpaths (cw) and percent outside of wheelpaths (co). Tables 1-1 and 1-2 give computer codes for percentages as well as numerical deduct *(1)*. Crack condition is then defined as a crack index (CI):

$$CI = 10 - (cw + co)$$

It can be seen that CI has a value between 0 and 10 with 10 representing the best and 0 representing the worst. In FDOT's practice, the rating value of 6 is considered the threshold, and hence a pavement section with a CI of 6 or less is considered deficient.

There are two types of forecasting models that can be used to forecast pavement crack performance. The first type of model is based on structural conditions of the given pavement section. This type of model is considered a static model. Development of static models is based on statistical analysis, and hence, the form of the models may appear very complicated because of the multitude of variables associated with the model. The second type of model is based on historical performance of pavement characteristics, irrespective of the pavement structural conditions. By understanding the dynamics of the past changing process, this type of model can forecast future conditions. Since static models have been often attempted in the past, this study focused on developing a dynamic neural network model to forecast pavement crack performance based on historical pavement condition data.

Table 1-1. Numerical Deductions for Crack Survey (cw)

Percent of Pavement Area Affected by Cracking	Predominate Cracking Class		
	1B Cracking Deduct	II Cracking Deduct	III Cracking Deduct
00 - 05	0.0	0.5	1.0
06 - 25	1.0	2.0	2.5
26 - 50	2.0	3.0	4.5
51 +	3.5	5.0	7.0

Table 1-2. Numerical Deductions for Crack Survey (co)

Percent of Pavement Area Affected by Cracking	Predominate Cracking Class		
	1B Cracking Deduct	II Cracking Deduct	III Cracking Deduct
00 - 05	0.0	0.0	0.0
06 - 25	0.5	1.0	1.0
26 - 50	1.0	1.5	2.0
51 +	1.5	2.0	3.0

1.4 Project Background

Pavement crack characteristic is one of the most important measures to evaluate pavement performance in a PMS. For the purpose of maintenance planning and project programming at the network level, highway agencies are particularly interested in forecasting the deterioration of crack characteristics. In 1997, the Florida Department of Transportation (FDOT) sponsored the Department of Civil & Environmental Engineering at the University of South Florida (USF) to conduct a research project, "Pavement Surface Crack Index Forecasting by Neural Network Method", to develop a pavement crack performance forecasting model for FDOT.

The research was to develop forecasting models that are based on historical information to forecast pavement surface crack performance. An artificial neural network model was developed in the research. The main objectives of the study were:

- to review the existing models used to predict pavement crack characteristics,
- to review the FDOT database to identify useful pavement crack data sources for the model development,
- to develop a neural network model for crack forecasting based on historical information, and
- to evaluate the performance of the neural network model in terms of comparison of other existing models.

This report summarizes the research project and presents results obtained from the research project. It was found from the results that the forecasting model developed in the project can be adequately applied to forecast pavement crack index.

CHAPTER 2

LITERATURE REVIEW

Several types of mathematical models currently exist and are used to forecast pavement performances. These models range from simple regression to complicated transition probability method. On the other hand, artificial intelligence techniques, which include expert system and neural network, have been applied in pavement modeling to some certain extent.

2.1 Review of Existing Pavement Performance Forecasting Models

Development of pavement performance models has been an active area in the last couple of decades, which constitute a key component of PMS. They are employed in determining maintenance and rehabilitation needs, life-cycle cost analysis, and pavement design. Typically, pavement performance forecasting models relate indicators of pavement conditions, such as roughness or crack index, to explanatory variables such as pavement structure, traffic loads, age, and environmental factors. Two basic types of performance models were suggested by Mahoney (Mahoney 1990): deterministic and probabilistic. Deterministic models are further divided into three subtypes:

1) Mechanistic-empirical Models

Mechanistic models are based on some type of primary response (such as stress, strain or deflection) under loading. These models are used in situations where the relationship between the response and explanatory variables is exactly known. So far, few of this type of models have been developed due to the complexity of the interaction between variables. However, the mechanistic-empirical models, in which a response variable is related to a measured structural or functional deterioration, such as crack or roughness through regression analysis, are widely used. A good example is provided by Queiroz (Queiroz 1983) for forecasting pavement roughness. The model used by Queiroz is:

$$\log(QI) = 1.297 + 9.22(10^{-3})(AGE) + 9.08(10^{-2})(ST) - 7.03(10^{-2})(RH) + 5.57(10^{-4})(SEN1)(\log N)$$

where:

QI = roughness (counts/km),

AGE = pavement age in years,

ST = surface type dummy variable (0 for as constructed and 1 for overlayed),

RH = state of rehabilitation indicator (0 for as constructed and 1 for overlayed),

$SEN1$ = strain energy at bottom of asphalt layer (10^{-4}kgf cm), and

N = cumulative equivalent single axle loads (ESAL).

To develop this forecasting model, data of 63 flexible pavement test sections were used. Calculated responses included surface deflection, horizontal tensile stress, strain and strain energy at the bottom of the surface layer, and vertical compressive stress at the top of the subgrade. These responses were then related to observed roughness (in quarter-car index) through regression analysis.

2) Pure Empirical Models

Pure empirical models are the most widely used models for pavement performance forecasting, where the observed pavement deterioration is related to one or more independent variables such as layer thickness, load applications, and environmental factors. This type of models is particularly applicable where a long-term database has been acquired. The model can be linear or non-linear, depending on whether the relationship between variables can be defined as a straight line. Because of the large number of variables that can be involved in a regression analysis, techniques have been developed to simplify the process. These techniques involve the grouping of pavements into families with common characteristics, such as surface type, functional classification, and traffic levels. When families of similar characteristics are developed, the analysis can focus only on the major variables, such as road age, greatly reducing the number of variables in the model. A generalized model forecasting roughness progression in flexible pavements is reported by Paterson in the Highway Design and maintenance Standards model (Paterson 1991). The models were intended for use in pavement management applications and as a performance model for pavement design. The purpose of this study was "to develop summary algorithms for forecasting pavement roughness that would be universally applicable and serve as a primary performance model for pavement management forecasting or a pavement design method." Two generalized models predicting roughness progression in flexible pavements were developed. In the first model, pavement roughness was related to traffic loading, strength, age, environment, rutting, cracking, and patching to predict roughness at any pavement age. The model has the following form:

$$RI_t = e^{mt}[RI_0 + 134(SNCK^{-4.99})(NE^t)] + 0.114RDS_t + 0.0066CRX_t + 0.16PHV_t + 0.01PAT_t$$

where:

RI_t = roughness at pavement age t [m/km international roughness index (IRI)],

RI_0 = initial roughness (m/km IRI),

NE_t = cumulative ESALs at age t (million ESAL/lane),

t = pavement age since rehabilitation or construction (years),

m = environmental coefficient (0.023 for wet, non-freeze climate in the original estimation),

SNCK = (1+SNC-F (HS) (CRX)),

SNC = structural number modified for subgrade strength,

F = coefficient that was 0.0000758 in original incremental model and that in integrated form is approximated by half that value, that is, 0.00004 ,

HS = thickness of bound layers (mm),

CRX_t = area of indexed cracking at time t (%), in which the areas of each class of cracking are weighted by the crack width (2 mm for narrow cracks and 4 for wide cracks),

RDS_t = standard deviation of rut depths (mm),

PHV_t = volume of potholing (m^3/lane-km),

PAT_t = area of patching (%).

The second model was simpler and generally structural, omitting surface distress parameters and compensating for this through the primary structural, traffic, age, and environmental factors. It was adequate for use where moderately good maintenance standards were being applied. As concluded by the authors, the two summary models were suited for applications to pavement performance forecasting in pavement management systems and economic evaluation analyses. Its mechanistic component form also can be expected to be valid in most countries and environments.

3) Expert System Models

In cases where sufficient data are not available, expert models may be developed based on the opinions of experienced engineers and maintenance personnel who are familiar with deterioration patterns of different pavement types. In practice, some highway agencies may rely 100% on expert opinions for modeling deterioration, while others may rely 50% on expert opinion and 50% on historical database. For example, the South Dakota Department of Transportation used this approach for the development of its deterioration models (SD93-14). Experienced engineers were asked to provide estimates of the ages of pavements reaching particular conditions in terms of distress type, severity, and extent. Then the scaling approach was used to develop the deduct values associated with each of the severity and extent classifications for all defined distress types. By combining these two steps a regression analysis was performed to determine the coefficients for the model. The model form could be:

$$PPI = c + at^{b}$$

where:

PPI = Pavement performance index, could be individual condition index or the composite index;

c = the maximum value of the index;

a = Slope of the deterioration curve;

t = Age of the pavement; and

b = Exponent coefficient of the curve.

In addition to the deterministic models discussed herein, the Markov model is a popular probabilistic model in which the probability of pavement condition changing from one state to another is dependent only on the current state (Karan 1976). A transition probability matrix defines the probability that a pavement in an initial state will be in some future condition state. The development of Markov models requires that an agency define all possible condition states for each type of pavement, and the probability of pavements changing from each state to another. Disadvantage of this approach is the need for developing a transition probability matrix, and pavement history is difficult to include in the Markov model since the estimate of the future state of pavement is based only on the current state.

2.2 Application of Neural Networks in Pavement Management

Although numerous pavement performance forecasting models have been proposed in the last couple of decades, a majority of those have relied on regression techniques to link a pavement condition indicator to a vector of explanatory variables. In recent years, artificial neural networks have been advocated as an alternative to traditional regression models. Neural networks are application of an algorithm inspired by research into human brain which can "learn" directly from data. It can be defined as "highly simplified models of human nervous system, exhibiting abilities such as learning, generalization, and abstraction." One of the advantages of a neural network model is that a well-defined mathematical process is not required for algorithmically converting an input into an output. A collection of representative examples of desired translation will suffice. Once trained a neural network can perform classification, clustering and forecasting tasks.

Atton-Okine et al. (Okine 1994) applied neural network to develop a pavement roughness progression model. The training data was generated from RODEMAN, a Road Deterioration and Maintenance Submodel of HDM-III. The approach utilized a full empirical simulation model to generate roughness data. The neural network model was then developed to relate pavement roughness progression in flexible pavement to pavement structural deformation, incremental traffic loadings, extent of cracking and thickness of cracked layer, incremental variation of rut depth, surface defects, patching and potholing, and environmental and non-traffic-related mechanisms, such as road age. Three different architectures of the network were examined, which had one, two, or three layers. The back-propagation learning algorithm was used in the learning process. After training, the network output is compared with the desired results in terms of the mean-square error (MSE). As concluded, the application of neural networks in pavement deterioration modeling is feasible when a large database on pavement condition can be acquired, which forms the basis for developing a generic intelligent pavement deterioration process. On the other hand, since the training and testing were done using simulated data, it was recognized that this particular approach might not be general enough to perform well on other data set, especially on in-service pavements.

Horiki et al. (Horiki 1994) developed a neural network model for asphalt pavement distress forecasting. The output was MCI (Maintenance Condition Index), and the inputs include cumulative truck traffic, T_A (coefficient of relative strength of asphalt pavement corresponding to SN of AASHTO), and CBR (%). A basic neural network model was first constructed by training with various numerical data generated from AASHTO pavement design equation. Then the basic neural network was retrained by the back propagation algorithm with available field pavement performance data in the districts examined. The conformity analysis of the neural network output with the measured data showed that the correlation coefficient between those two is 0.81. It was suggested possible to prepare a design curve giving consideration to regional characteristics by retraining with local pavement performance data.

Owusu-Ababio (Owusu-Ababio 1995) developed a neural network model for modeling skid resistance on flexible pavements. In this study, the use of neural network models as an alternative to regression models for predicting skid resistance on flexible pavements was examined. By using data from in-service flexible pavements, separate skid resistance models were developed with both regression and neural network methods. As shown in the testing and comparison results of these two models, the neural network model was capable of fitting better to the actual data than the multiple linear regression model. In the mean while, the neural network model is dynamic, can be updated easily in less time, and can model the environment more convincingly than regression models.

Lu et al. (Lu 1991) reported an adaptive filter forecasting system for pavement roughness forecasting. Since pavement roughness change over time is caused by some important conditions and certain stochastic factors, an adaptive filter forecasting system was presented that can forecast pavement roughness conditions by means of an adaptive filter using roughness history. Then, selection of the adaptive filter structure and its stability, based on roughness data collected from in-service pavements, were discussed. It was concluded that the adaptive filter forecasting system can be used as a dynamic time-series predictor of pavement roughness condition. By careful selection of the model structure and parameters, the adaptive filter model can converge to a stable state.

Besides, neural networks have been used widely in pavement engineering other than performance forecasting. For example, Eldln et al. (Eldin 1995) Applied neural network method to successfully evaluate pavement condition. TAHA et al. (Taha 1995) built an evolutionary model with both neural network and genetic algorithm for pavement maintenance strategy selection. Fwa et al. (Fwa 1993) developed a neural network model for highway maintenance priority rating.

2.3 Review of Pavement Crack Performance Forecasting Models

In the past, many agencies have developed pavement performance forecasting models for a single aggregated index based either on roughness (PSI) or distress (PCI), which is becoming more widely used in North America through the adoption of standard procedures. However, it has been observed that a single aggregated index has distinct limitations when it is used to identify pavement rehabilitation needs. Considerable distress has occurred before the single aggregated index changes sufficiently to reach the minimum level for rehabilitation, thus restricting the selection of the more cost-effective rehabilitation strategies. On the other hand, the use of a single aggregated index also restricts the information about individual distress in PMS which could be used to analyze various treatments available for pavement rehabilitation. There is a clear advantage to develop forecasting models for individual distress indexes which can identify projects at earlier stages of pavement deterioration and provide more specific information to completely perform the optimization task.

Limitation of cracking in pavement is a major issue for both pavement management and design. Usually, pavement cracking can be divided into several categories: alligator or fatigue cracking, block cracking, longitudinal, and transverse cracking. Fatigue cracking is a series of interconnecting cracks caused by the failure of asphalt surface or stabilized base under repeated traffic loading. Blocking cracking divides the asphalt surface into approximately rectangular pieces, which is caused mainly by the shrinkage of hot mix asphalt (HMA) and daily temperature cycling. Longitudinal cracks are parallel to the pavement centerline, while transverse cracks extend across the centerline. They may be caused by the shrinkage of asphalt surface due to environment influence or result from reflective cracks caused by cracks beneath the asphalt surface. All these types of cracking are usually grouped into a single cracking index with some specific definition. Paterson (Paterson 1992) proposed a universal cracking indicator for pavements. The cracking indicator was defined by the product of the area extent, intensity, and crack width of a set of cracks in a dimensionless form with a scaling factor of 100,000. The cracking indicator was then evaluated from a number of perspectives, including measurability, relevance to performance impacts, diagnostic analysis, and maintenance decisions. It was concluded that the proposed cracking indicator was sound and powerful in concept, flexible, and therefore warranted consideration as a universal norm by standard organization.

In the past years, a few crack forecasting models have been developed. Two basic forms of asphalt concrete fatigue cracking model have been reported in the literature. (AI 1982 and Shell 1978) They are "Strain/modulus based models" and "Strain based models", as given in the following equations.

$$N_f = f_1[\varepsilon]^{-f_2}[E_{AC}]^{-f_3}$$

$$N_f = f_1[\varepsilon]^{-f_2}$$

where:

N_f = Allowable number of load repetitions to control fatigue cracking;

ε = Tensile strain, inch/inch;

E_{AC} = AC modulus, psi;

f_1, f_2, f_3 = Constants determined from laboratory tests or field observations.

The strain/modulus based model is more theoretical than the stain based model because it includes the parameter E_{AC} in the formulation. Since the parameter f_2 is much higher than f_3, the strain term becomes dominant. Therefore, many agencies use the strain based model for simplicity.

Queiroz (Queiroz 1983) developed separate regression equations which predict cracking initiation and the rate of cracking progression. The cracking initiation model used the number of equivalent single axles to cracking initiation as the dependent variable; while cracking progression was specified to be a function of both structural and age parameters of the pavement. The cracking progression equation is as follows:

$$CR = -8.70 + 0.258(HST)(\log N) + 1.006(10^{-7})(HST)N$$

where:

CR = Percent of pavement area cracked;

HST = Horizontal tensile stress at the bottom of the asphalt layer; and

N = Cumulative equivalent single axle loads (ESAL).

Besides the mechanistic-empirical models above, some direct regression models have been reported in literature. A cracking regression model was reported by Newton C. Jackson (Jackson 1996). The result showed that the forecasting curve fits the points with a R^2 value of 0.99 and a standard error of 2.42. The equation is as follows:

$$\text{Fatigue Cracking Index} = 100 - 0.11726(\text{Age})^{2.2}$$

Sood et al. (Sood 1994) introduced a regression model to predict change in cracking (%) in India. Both linear and nonlinear analyses were performed. The incremental approach, taking the difference between the two successive observations, was adopted in multivariate regression analysis as the most logical approach available for time-series data analysis for the forecastings of change over the preceding value. The final form was determined as:

$$\Delta CR_t = 0.55(\Delta CSAL)(MSN^{-5})(e^{m(PAGE)} + m(CR_i)t)$$

where:

ΔCR_t = Change in cracking (%) over a time t (years);

$\Delta CSAL$ = Change in cumulative standard axles (msa);

MSN = Modified structure number;

m = Environmental factor;

PAGE = Pavement age since last renewal/strengthening (years);

CR_i = Initial cracking (%); and

t = Time interval (years).

In this model, the change in cracking (%) was predicted by current crack condition, traffic volume and loading, pavement age, pavement strength, and maintenance level. The models were partly validated with the data available from some of the experimental sections. It was found that forecasts for pavement cracking could be made with reasonable accuracy by using this model in India.

The literature review in this study indicated that pavement performance model development has been an active research area in the past several decades. Although traditional models have been widely used by highway agencies, some advanced technologies, including neural networks, have been applied to forecast pavement performance. However, the literatures reviewed have not shown any studies on the topics related to forecasting pavement crack performance with neural network models based on the available literature sources in this study. Also, no sufficient researches have been seen related to the time-series modeling for pavement performance. This study attempted to fill these areas. A neural network model for forecasting of pavement crack performance based on historical time-series data was developed in this effort.

CHAPTER 3

METHODOLOGY

This chapter presents the methodology used in this study for crack performance forecasting. The forecasting models were based on neural network techniques. A detailed description of neural networks is provided in this chapter, along with the pragmatic method appropriate for pavement crack condition forecasting.

3.1 Artificial Intelligence and Neural Networks

Historically, the term artificial intelligence has been used to name the field of computer science dedicated to producing programs that attempt to be as smart as humans. Expert systems and neural networks are two forms of artificial intelligence, each with distinct strengths and weaknesses. Most implementations of artificial intelligence are programs that simulate either the deductive or inductive intelligence of human being. Deduction reasons in steps to a conclusion based on given premises. A deductive system, which can be simulated by expert systems, requires rules or instructions executed one at a time to arrive at the answer. By contrast, induction takes in a large amount of information all at once and then draws a conclusion. Neural networks can be used to simulate the inductive behavior of humans. Once trained, the neural network is able to look at input data and produce an appreciate answer.

A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the human brain in two respects:

- Knowledge is acquired by the network through a learning process, and

- Connection strengths between neurons, which are known as synaptic weights, are used to store the knowledge.

In computing terms, neural networks have a unique set of characteristics derived through its massively parallel-distributed structure and its ability to learn and generalize. These two information-processing capabilities make it possible for neural networks to solve complex problems in the real world. The key characteristics of neural networks can be summarized as follows:

- *Learning from experience*: Neural networks are particularly suited to solve problems whose solution is complex and difficult to specify, but which provide an abundance of observed data.

- *Generalizing from examples*: An important attribute of neural networks is the ability to learn from previous experiences and then give the correct response to data that it has not encountered.

- *Nonlinearity*: Many other techniques are based on the theory of linear systems. However, neural networks can be trained to generate nonlinear mappings, which often give them an advantage for dealing with complex, real-world problems. Nonlinearity is a particularly important property if the underlying physical mechanism is inherently nonlinear.

- *Computational efficiency*: Although the training of a neural network is computationally intensive, the computational requirements of a fully trained neural network applied on test data are modest. For large problems, speed can be gained through parallel processing, as neural networks are intrinsically parallel structures.

- *Adaptivity*: Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions.

A neural network is an excellent candidate for any application requiring pattern recognition. Neural networks are able to recognize patterns, which may consist of visual, numeric, or symbolic data, even when the data is noisy, ambiguous, or distorted. In general, neural network tasks may be divided into five types of distinct applications:

- *Classification*: Deciding into which category an input pattern falls into.
- *Association*: Acts as a content addressable memory that recalls an output given some part of it as an input.
- *Codification*: Encoding compresses inputs by producing output with reduced dimension. The opposite task, decoding, may also be of interest.
- *Simulation*: The creation of a novel output for an input that acts as stimulus. The network has been exposed to a sample of possible stimuli.
- *Modeling*: The network mapping process involves nonlinear functions that can consequently cover a greater range of problem complexity. Although other nonlinear techniques exist, the neural network is superior in its generality and practical ease in implementation.

3.2 Neural Networks and Statistical Modeling

A neural network may be considered as a data processing technique that maps, or relates, some type of input stream of information to an output stream of data. For example, the input may be the pavement crack condition historical data, pavement type, road age, etc., and the output may produce an estimate of the future pavement crack condition. On the other hand, the goal of statistical modeling is to find an equation that captures the general pattern of a relationship, which is usually derived from observed examples. Therefore, the fields of statistical modeling and neural networks are closely related in the context of input-output mapping. The principal difference between these two fields is that traditional statistical models typically need a equation to be specified, which could be difficult in complicated nonlinear cases, while neural networks have been mainly used to deal with nonlinear problems without requiring a pre-specified function form. However, with the appearance of the backpropagation neural network (BPNN), of which the learning paradigm is called supervised learning, these two fields touch most closely in solving mathematical modeling problems. This technique solves one of the central problems in neural networks, and it is a useful modeling tool as well.

Supervised learning involves the modification of the synaptic weights of a neural network by applying a set of training examples. Each example consists of a unique input signal and the corresponding desired response. The network is presented an example picked at random from the set, and the synaptic weights of the network are modified so as to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with an appropriate statistical criterion. The training of the network is repeated for many examples in the set until the network reaches a steady state, where there are no further significant changes in the weights. Thus, the network learns from the examples by constructing an input-output mapping for the problem at hand.

Statistical modeling techniques are used to derive relationships between variables from examples as well. In the case of pavement crack condition forecasting, the examples are crack condition data in the last few years. To derive the equation from the examples, the values of the independent and dependent variables for each example need to be known. In this case, the independent variables involves the crack condition history data and other pavement descriptive information, and the dependent variable is the future crack condition. During the running mode, a running file needs to be prepared, which contains the independent variables of each new example for which an estimate of the dependent variable is desired.

The prototypical example of a statistical modeling technique is linear regression. The equation produced by a modeling method can be thought of as a mapping, because it permits us to map any point in the space of the independent variables onto a point in the space of dependent variables. The error of the mapping function is comes from two sources. The first source of error is noise, which includes inaccuracies in the data introduced by measuring instruments, and inaccuracies due to the fact that the independent variables do not contain all the information needed to determine the dependent variables. The second source of error is the fact that the mapping function may not have the same form as the target function. The so-called target function could be an idealized and unknowable function that expresses the "true" relationship between the independent and dependent variables.

The fact that linear regression imposes a linear form on the mapping function can severely limit its accuracy. In cases where the problem domain is not a linear space, it is usually necessary to transform the variables so as to make the relationship linear. A better approach is to automate the process of deciding what shape the mapping function should have. What is needed, then, is a modeling technique based on a mapping function that is complex enough to be flexible. Although some simple curves, such as polynomial regression and exponential equation, have been used to simulate the real world condition, the optimum solution is a technique that can take on any form the data requires. One of these advanced modeling techniques is the backpropagation neural network.

3.3 The Neural Network Algorithm

There are many types of neural networks, but all have three things in common. A neural network can be described in terms of its individual neurons, the connections between them (topology), and its learning rule. Both biological and artificial neural networks contain neurons, real or simulated. These neurons have many connections to each other which transfer information. The knowledge of a network is distributed across the interconnections between the neurons. A typical neuron receives input, either excitation or inhibition, from many other neurons. A neuron calculates its own output by finding the weighted sum of its inputs, generating an activation level and passing that through an output on transfer function. The point where two neurons communicate is called a connection. The strength of the connection between two neurons is called a weight. The collection of weights arranged in rows and columns is called the weight matrix. A neural network learns by changing its response as the inputs change. Because the weights in the network can change, the relationship of the network's output to its inputs can be altered as well. In this sense, the learning rule is the very heart of a neural network, which determines the behavior of the network and how that behavior can change over time.

3.3.1 Single Neuron

Artificial neurons as information processing devices were first proposed more than fifty years ago. As shown in Figure 3-1, a neuron computes a weighted summation of its n inputs, the result of which is then thresholded to give a binary output y which is either +1 or −1. The bias weight, θ, is introduced whose input is fixed at +1. This bias weight is adaptive like the others and its use allows greater flexibility of the learning process.

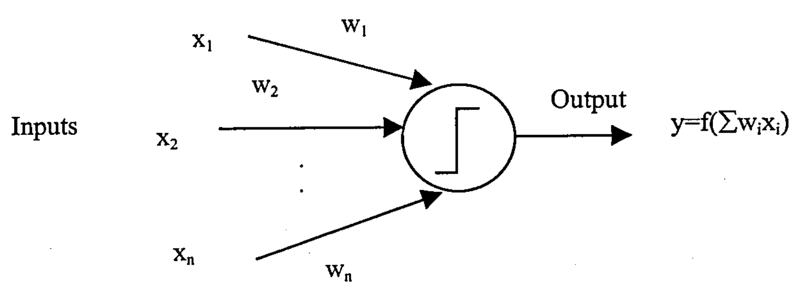


Figure 3-1. Schematic Diagram of an Artificial Neuron

For a classification problem, the neuron assigns input patterns, represented by the vector of numbers $x = (x_1, x_2, \ldots, x_n)$, either to class A (for which y would be +1) or class B (for which y would be −1). Thus:

$$y = f(\sum_{i=1}^{n} w_i x_i) = \begin{cases} 1 & \text{when } \sum_{i=1}^{n} w_i x_i > 0; \\ -1 & \text{when } \sum_{i=1}^{n} w_i x_i \leq 0. \end{cases}$$

In the above equation, y is the neuron output and f is a hard-limiting of threshold function, sometimes known as the neuron's transfer function, which gives an output of +1 whenever $\Sigma w_i x_i$ is greater than zero (the threshold value) or −1 whenever $\Sigma w_i x_i$ is less than (or equal to) zero.

The learning process is to adjust all the weights and let the output y approach the desired output so that the neuron performs the classification task correctly. Multi-class problems can also be solved by having a number of neurons operating in parallel.

3.3.2 Backpropagation Neural Network (BPNN)

By far, the BPNN is the most popular one used for mathematical modeling. Backpropagation is a supervised learning scheme by which a layered neural network with continuously valued neurons is trained to become a pattern-matching machine. It provides a way of using examples of a target function to find the weights that make a certain mapping function hidden in the neural network approximate the target function as closely as possible. As shown in Figure 3-2, the neurons of the networks are structured in multiple layers: input, hidden, and output. Each hidden-layer neuron receives input from all neurons in the input layer through weighted connections (w). In addition, each neuron is associated with a bias term, called the threshold, θ. This bias term works as a horizontal shift for the origin of the transfer function to accommodate the magnitude of incoming signals to the neuron. Specific values of both w and θ for a given neural network are determined during the training phase.

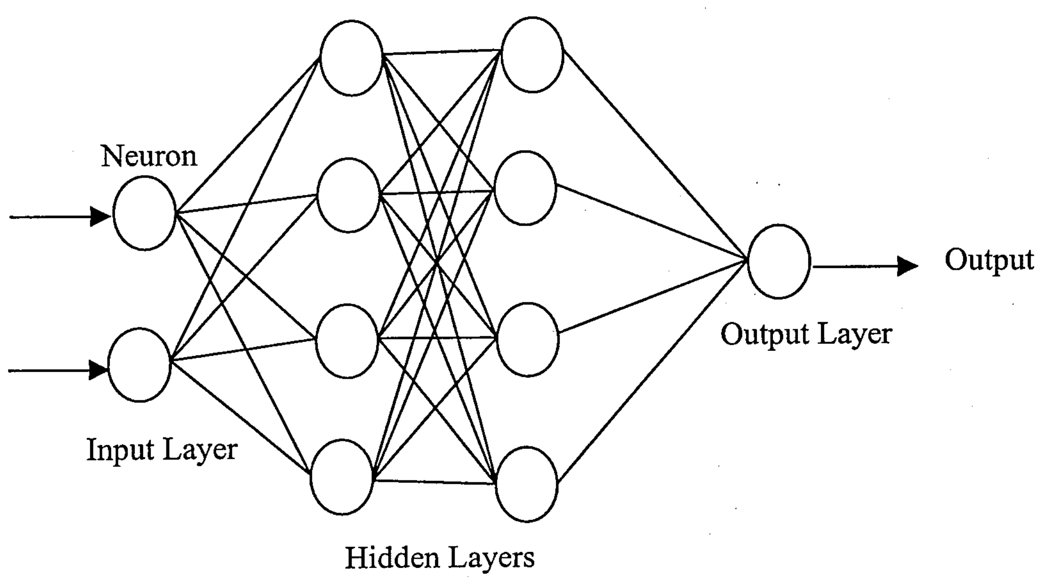


Figure 3-2. A Three-Layer Backpropagation Neural Network

The network operates in two modes: mapping and training mode. In mapping mode, information flows forward through the network, from inputs to outputs. In training mode, the information flow alternates between forward and backward. In the mapping mode, the network processes one example at a time, producing an estimate of the values of the dependent variables based on the values of the independent variables for the given example. First, a set of values for the independent variables is loaded onto the input layer of the network. The input-layer neurons do no calculation --- each neuron merely sends a copy of its value to all the hidden-layer neurons. Each hidden neuron calculates the weighted sum of the inputs using its unique connection strengths as weights. Next, each hidden neuron computes a transfer function of its input sum and sends the result to all the output-layer neurons. Then, each output-layer neuron performs a similar calculation and outputs the resulting value as an estimate of the dependent variable it represents.

The training mode refers to the process in which the network is exposed to examples with correct output values known. The training algorithm consists of three steps. In the first step, the training patterns obtained from the database are fed into the input layer of the network. These inputs are propagated through the network until reaching the output layer. The output of each neuron is calculated by the following transfer function:

$$a = \sum_{i=1}^{n} w_i x_i$$

$$O = f(a) = \frac{1}{1 + e^{-ga}}$$

where:

O = neuron output,

a = input to the transfer function,

x_i = i^{th} input,

w_i = weight of connection i,

g = gain of sigmoid function, and

n = number of inputs to one neuron.

In the second step, the neural network outputs are subtracted from the desired values to obtain an error signal. This error signal is the basis for the coming backpropagation step. The following equation defines the error signal:

$$E_{RMS} = \sqrt{\frac{\sum_{j=1}^{N_o}\sum_{k=1}^{N_e}(T_{jk} - O_{jk})^2}{N_o N_e}}$$

where:

E_{RMS} = root mean square error,

N_o = number of neurons in the output layer,

N_e = total number of patterns in an epoch,

T_{jk} = target (desired) value of the j^{th} neuron, and the k^{th} pattern, and

O_{jk} = output of the j^{th} neuron, and the k^{th} pattern.

In the third step, error is minimized by backpropagation of the error signal through the neural network. In this process, the respective contribution of each hidden neuron is computed and corresponding weight adjustments needed to minimize the error are derived.

For each output layer neuron k, compute the δ value, defined as follows:

$$\delta_k = (T_k - O_k)f'(x_k)$$

where:

δ_k = adjusted error for output neuron k;

T_k = target value of output neuron k;

O_k = output value of output neuron k; and

x_k = input to output neuron k.

Backpropagate the δ value through the network to the preceding hidden layer. For each hidden layer neuron j connected to the output neurons k, compute the new δ value:

$$\delta_j = f'(x_j)\sum_k \delta_k w_{jk}$$

where:

δ_j = adjusted error of hidden neuron j;

x_j = input to the hidden neuron j;

δ_k = adjusted error of output neuron k connected with hidden neuron j; and

w_{jk} = connection weight between neuron j and k.

The weight connecting any two neurons is updated by the following equation:

$$\bullet\, p \longrightarrow \bullet\, q$$

$$\Delta w_{qp} = \alpha \delta_q O_p$$

where:

Δw_{qp} = adjustment of weight between preceding layer neuron p and proceeding layer neuron q;

δ_q = adjusted error of proceeding layer neuron q;

O_p = output of preceding layer neuron p; and

α = learning coefficient (a positive constant).

The training process repeats steps 1 through 3 for all patterns in the training set until the overall error is acceptably low based on a given criterion. If the network has not converged then go back to step 1, otherwise stop training. Once trained, the neural network has the capability of adapting to changing input. If the trained network results in good accuracy on the testing and validation data set, the development process is completed.

Although theoretically complicated, the training process is typically implemented by a computer program, within which the training algorithm has been incorporated. Popular neural network development packages in the market include BrainMaker, DataEngine, NeuronDimesion, etc. These software packages vary in terms of training speed, pre- and post- data processing utility, and convenience of user interaction. Once trained, the BPNN can be incorporated into other programs, the running of this model can be implemented through a user-friendly interface. In this effort, BrainMaker was selected as the development tool, the detailed training and testing process with BrainMaker is described in Chapter 5.

3.4 Application of Neural Networks to Forecast Pavement Crack Condition

Primarily, there are two distinct types of models available for crack condition forecasting. The first type is a static model and can be conceptually described by the following equation:

$$CK_t = f(S_t, M_t, T_t, E_t, t, \text{etc.})$$

where:

CK_t = crack condition at age t,

S_t = pavement structural conditions at age t,

M_t = pavement materials characteristics at age t,

T_t = traffic conditions at age t, and

E_t = environmental conditions at age t.

Often, the development of such models is based on field and/or laboratory data and statistical analysis. As a result, formats of these models are generally complicated due to the multitude of variables associated. Further, these models may lack accuracy due to numerous uncertainties.

The second type, a dynamic model, can be described by the following equation:

$$CK_t = f(CK_{t-1}, CK_{t-2}, \ldots, CK_{t-N})$$

Pavement crack condition at age t, CK_t, is forecast using historical crack condition data at ages t-1, t-2, …, t-N. This type of model is based on historical performance of pavement characteristics, irrespective of other variables used in the static model. By understanding the dynamics of the changing process, this type of model can forecast future conditions based on the past conditions. It is realized that the structural condition is reflected in the historical changing process of pavement crack condition. Thus, if external and structural conditions are not significantly changed within a relatively short time period, the model based on historical information could produce a reasonably accurate forecast of future crack condition. This type of a model is called a time-series model, which has been successfully applied in transportation engineering.

A time-series model can describe time-dependant processes in which past data influence future data in the presence of underlying deterministic factors. These factors may be characterized by trends, cycles, and non-stationary behavior of the processes. It is these recurring patterns and relationships that the predictive models attempt to recognize. On the other hand, there is always certain level of randomness existing in the time-series. Both the deterministic trend and the randomness should be addressed in the forecasting model. A traditional statistical treatment of time series would include tests for randomness, analyses of series into component parts, smoothing, and the use of autoregressive models. However, the inherently nonlinear time series, such as that found in pavement crack deterioration process, are more suitable for analysis by the general nonlinear mapping provided by a neural network, than by linear based autoregressive models. Neural networks are nonlinear models that can be trained to map past and future data of a time-series, thereby uncovering the hidden relationships governing the data.

In this study a BPNN model was developed to capture the time varying characteristics of pavement crack deterioration. Usually, a system is said to be temporally dependent if the future state of the system depends on the current and/or previous states. In recent years, BPNN, which uses a vector (time lagging) containing past states as input, has been advocated for modeling time varying systems. In order to develop a forecasting model using BPNN, it is important that the forecasting model characterize the embedding dimension of the series, which defines the number of previous states of the time series considered for determining the future states. In this study, CI was chosen as the temporally dependent variable with the embedding dimension of four. This means that the pavement crack condition for the following year [CI(t+1)] can be forecast based on current pavement crack condition [CI(t)], and the previous two-year crack conditions, CI(t-1) and CI(t-2). In addition, other relevant variables that are not time-dependent such as pavement type, maintenance cycle, and road age, were included in the model as well.

CHAPTER 4

DATABASE REVIEW AND PREPROCESSING

The following sections describe the detailed efforts of database review and data reduction. The data items in the FDOT database included crack, roughness, rutting, road identification, section begin and end mileage, road age, road type, number of lanes, district, direction, maintenance cycle, etc. After erroneous data processing and reduction, the original data was transformed into a new format which would be suitable for further analyses.

4.1 Crack Database Review

The database used for this analysis included a time series of pavement crack indexes acquired by FDOT between 1972 and 1998. The original database contained 139421 records of historically surveyed data which described the pavement deterioration of the highway network in Florida. Each pavement section stored in the database was identified by location, pavement type, and maintenance cycle, etc. The distribution of pavement sections among different maintenance cycles and pavement type is presented in Figure 4-1. It can be seen that 80 percent of the road sections were flexible pavement, the other 20 percent were rigid and other types of pavement. As for the distribution of roadway sections among maintenance cycles, 70 percent of road sections were in the first and second cycle, indicating a relatively new highway network.

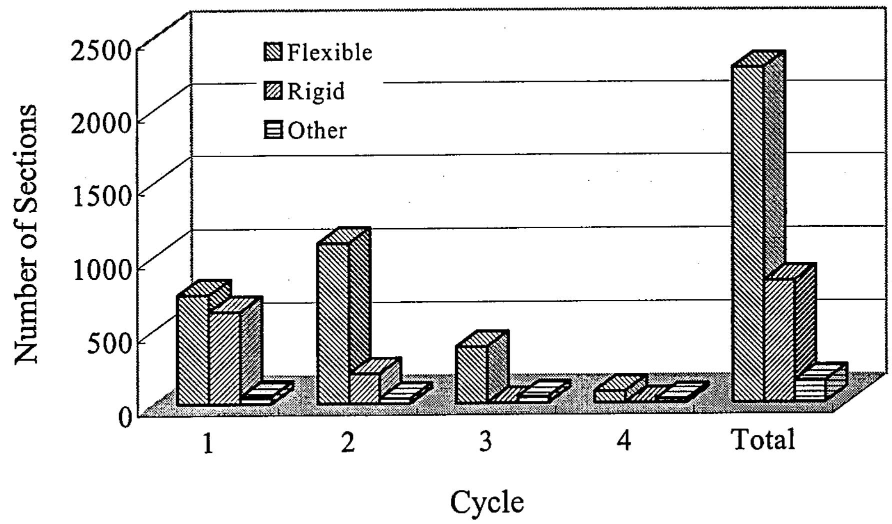


Figure 4-1. Distribution of Pavement Sections
(Types and Cycles)

To determine whether or not the database contained enough samples, two factors were considered important: (1) the form of the target function: to maintain a given accuracy, sample size needed to increase as the target function becomes more complex, and (2) the noise in the data: to maintain a given accuracy, sample size needed to increase as noise increased. Given a target function of a certain complexity, and a certain amount of noise in the data, there was an absolute limit to the accuracy the model can achieve. An infinite sample size would be needed to achieve the limit of accuracy. For neural network modeling, since the complexity of the target function was not a limit, noise alone determined the limit of accuracy. If the sample was large enough, the complexity of the network's mapping function could be increased to match the complexity of the target function. Consequently, as sample size increased neural network model's accuracy was limited only by the noise in the data. Usually, Neural network can benefit more from large samples than regression can. Because larger samples allow us to use more hidden neurons or to continue training longer, the accuracy can be improved by increasing sample size. On the other hand, neural network model does not require a larger sample than a regression model. As the sample size gets smaller, we can use fewer hidden neurons or halt training sooner to avoid overfitting. The basic rule, therefore, is to use the largest sample available, so long as it fits in the computer memory or storage. In this study, all the data after preprocessing and transformation were used for model development.

4.2 Software Selection for Data Preprocessing

The massive amount of data in this study required an integrated statistics software package which could provide complete control over data access, management, analysis and presentation. The Statistics Analysis System (SAS) software package was selected for data processing because of its power, flexibility, and ease of use. The SAS software package can take data in any format from any source and enable users to manipulate the data conveniently and, finally, it can export the data file to many other software packages for further processing.

Microsoft Excel was also selected as a data analysis tool because it was powerful for spreadsheet analysis and plotting of meaningful results. The data communication between SAS and Excel was achieved through the plain test file.

4.3 Data Preprocessing

Pavement sections were classified in the study to group pavement sections with similar properties. After carefully sifting and comparing, pavement type (flexible or rigid) and maintenance cycle were selected as grouping factors. Then, the subsets for each family were reconstructed in the lagged window format through which the network could monitor the movement of crack data over time. The data format stored in the original database is shown in Table 4-1, and the transformed data format is shown in Table 4-2.

There were some errors in the original database which might have originated during data collection or coding. Quality control of data was important to maintain the data consistency, as the potential for erroneous performance forecasting was imminent without quality control during data preprocessing. A computerized screening program was developed to identify obviously erroneous data such as duplicate records, incomplete data, and unrealistic values. For example, the effect of pavement maintenance activities had to be addressed. If the crack condition of a pavement section exhibited a significant improvement from the previous year in one cycle, the deterioration process of the section in this cycle should be divided into two series. The time series which clearly exhibited unexpected deterioration trends were excluded form the training data set. The database after preprocessing contained variables needed for BPNN training, such as historical crack condition data, pavement type, maintenance cycle, and road age. By omitting useless variables in the original database, the size of the data file was significantly reduced. Some administrative variables, such as beginning and ending milepost of each section, number of lanes, were also included in the database. After running the BPNN model, these variables would be used to calculate the total deficient lane miles of the whole highway network. The variables in the database after preprocessing include:

- CI(t-2): Crack index for the year (t-2);
- CI(t-1): Crack index for the year (t-1);
- CI(t): Crack index for the year t (current year);
- CI(t+1): Crack index for the year (t+1);
- RDWYID: Identification number of the section;
- YEAR: Current year;
- LANES: Number of lanes for the section;
- DISTRICT: District number of the section;
- BEGIN: Beginning lane mileage of the section;
- ENDING: Ending lane mileage of the section;
- CYCLE: Pavement maintenance cycle of the section; and
- AGE: Road age since last major maintenance activity.

It was found in this study that data preprocessing is necessary for BPNN model development. This preprocessed database was then used to train, test, and validate the BPNN model. In the running mode of the BPNN model, the newly collected database each year would experience the same preprocessing procedure before it can be fed into the model. The only difference is that CI(t+1), the crack index to be forecast, does not exist in the running mode. The detailed model development process is presented in Chapter 5: Model Development.

Table 4-1. Format of the Original Database

RDWYID	YEAR	LANES	DISTRICT	TYPE	BEGIN	ENDING	CI	CYCLE	AGE
1010000	1986	2	1	1	0	0.491	9.5	2	4
1010000	1987	2	1	1	0	0.491	8.5	2	5
1010000	1988	2	1	1	0	0.491	8.5	2	6
1010000	1989	2	1	1	0	0.491	8.5	2	7
1010000	1990	2	1	1	0	0.491	7	2	8
1010000	1991	2	1	1	0	0.491	8	2	9
1010000	1992	2	1	1	0	0.491	8	2	10
1010000	1993	2	1	1	0	0.491	7	2	11
1010000	1994	2	1	1	0	0.491	7	2	12
1010000	1995	2	1	1	0	0.491	7	2	13
1010000	1996	2	1	1	0	0.491	7	2	14
1010000	1997	2	1	1	0	0.491	7	2	15
1010000	1998	2	1	1	0	0.491	5.5	2	16
1010000	1995	2	1	4	0.491	4.98	7.2	1	13
1010000	1996	2	1	4	0.491	4.98	6.9	1	14
1010000	1997	2	1	4	0.491	4.98	6	1	15
1010000	1998	2	1	4	0.491	4.98	5.9	1	16
1010000	1986	2	1	1	0.491	4.98	8.5	1	9
1010000	1987	2	1	1	0.491	4.98	9.5	1	10
1010000	1988	2	1	1	0.491	4.98	9.5	1	11

Table 4-2. Format of the Transformed Database

CI(t-2)	CI(t-1)	CI(t)	CI(t+1)	RDWYID	YEAR	LANES	DISTRICT	TYPE	BEGIN	ENDING	CYCLE	AGE
9.5	8.5	8.5	8.5	1010000	1988	2	1	1	0	0.491	2	6
8.5	8.5	8.5	7	1010000	1989	2	1	1	0	0.491	2	7
8.5	8.5	7	8	1010000	1990	2	1	1	0	0.491	2	8
8.5	7	8	8	1010000	1991	2	1	1	0	0.491	2	9
7	8	8	7	1010000	1992	2	1	1	0	0.491	2	10
8	8	7	7	1010000	1993	2	1	1	0	0.491	2	11
8	7	7	7	1010000	1994	2	1	1	0	0.491	2	12
7	7	7	7	1010000	1995	2	1	1	0	0.491	2	13
7	7	7	7	1010000	1996	2	1	1	0	0.491	2	14
7	7	7	5.5	1010000	1997	2	1	1	0	0.491	2	15
7.2	6.9	6	5.9	1010000	1997	2	1	4	0.491	4.98	1	15
8.5	9.5	9.5	9.5	1010000	1988	2	1	1	0.491	4.98	1	11
9.5	9.5	9.5	9.5	1010000	1989	2	1	1	0.491	4.98	1	12
9.5	9.5	9.5	10	1010000	1990	2	1	1	0.491	4.98	1	13
9.5	9.5	10	10	1010000	1991	2	1	1	0.491	4.98	1	14
9.5	10	10	8.5	1010000	1992	2	1	1	0.491	4.98	1	15
10	10	8.5	8	1010000	1993	2	1	1	0.491	4.98	1	16
10	8.5	8	8	1010000	1994	2	1	1	0.491	4.98	1	17
8.5	8	8	8	1010000	1995	2	1	1	0.491	4.98	1	18
8	8	8	8	1010000	1996	2	1	1	0.491	4.98	1	19
8	8	8	7.5	1010000	1997	2	1	1	0.491	4.98	1	20

CHAPTER 5

MODEL DEVELOPMENT

The model development in this study included training, testing, and validation. Training a neural network involves repeatedly presenting a set of examples to the network. The network takes in each example, makes a response as the output, checks this response against the correct answer, and makes corrections to the internal connections. Testing the network is the same as training it, except that the network is shown with the examples it has never seen before, and no weight adjustments are made during testing. Validation occurs after the neural network has been developed. Validating a network consists of presenting it with new input data and gathering the network outputs. Unlike testing, there is no known output, only known inputs in the validation. The following sections provide the detailed discussions about the development process.

5.1 Overview of the Neural Network Design Process

To design a neural network, the problem must be defined precisely. The user should decide what tasks the network is to perform. These tasks could be forecasting, recognizing, or classification. One cannot just throw all the spreadsheet data at the network and expect it to figure out what to learn from the data. The technology has not yet reached that level of sophistication.

The user also needs to choose the information on which the neural network will base its forecasting, recognition or classification. This should consist of whatever information is available that is relevant in determining the desired output. Neural networks learn by making associations between inputs and outputs. A network can associate the inputs “red”, “medium”, “round”, and “fruit” with the output “apple”. A network can also associate a decrease in the price of steel with a rise in the price of GM stock.

The user does not need to figure out procedures, rules, or formulas in the neural network development. The user should think about what kinds of input data the neural network can use to make an association with the desired output. Having a variety of data types increases the chance that various significant correlations can be found within the data. A network would probably not be able to accurately predict stock prices based solely on a collection of daily stock prices. It is better to have one or two extra items of data than not enough. The neural network will learn to pay attention to the items that are important and to ignore the few that don’t matter.

Another important part of design process is preparing to train the network by gathering examples for which correct answers are known. For example, to recognize a face, a network would need to have seen a picture of that face before. The training data were organized as facts (patterns) in a spreadsheet format. A fact is a collection of inputs coupled with the correct output(s). Each fact can be thought of as a flash card that is used to train the neural network. One side of the card contains the input information, and the other side contains the known answer which the neural network will learn to output during training. The deck of flash cards is called the training set.

A randomly sampling of facts should be set aside from the training set of facts for testing the network. Since the network generalization capability depends on its performance on the testing data set, it is not as important for neural network to learn a training set perfectly as it is for it to be able to provide correct answers for inputs it has never seen before.

Once trained, the neural network can be called from within some other program, perhaps an integrated system. The network may also be downloaded onto a chip for fast running. A trained neural network is considered intellectual property and may be copyrighted in the United States.

5.2 Data Subdivision

In this research, the purpose of preprocessing the data was to establish a database which can be directly used for further model development. The processed data set was further divided into three sub data sets. As shown in Figure 5-1, approximately 59 percent of data were used for network training, 26 percent of data were used to test the generalization ability of the network when facing data unseen in the training period. Finally, district survey data of 1998 (15% of the database) were withheld for model validation.

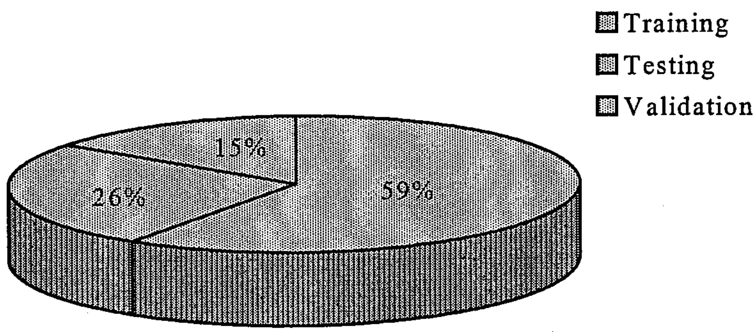


Figure 5-1. Database Subdivision

5.3 Software Selection

A developmental platform is usually a requirement to train neural networks. Often referred to as neural network simulators, these platforms are commercially available. Some factors worth considering when choosing a suitable neural network simulator are the required level of expertise, complexity, and the pre- and post-processing facilities. In this study, BrainMaker, a commercially available neural network simulator distributed by California Scientific Software *(15)*, was used for the development of the proposed neural network model.

The BrainMaker neural network simulator uses popular backpropagation training algorithm for network training. BrainMaker reads three kinds of neural network files: definition files, fact files, and network files. BrainMaker also creates different types of statistics and output files. They are all human-readable and editable. A definition file describes everything there is to know about the network to BrainMaker, such as the number of neurons in each layer, the type of data, and what is going to be displayed on the screen. BrainMaker uses the definition file to create the neural network. The default extension for the definition file is ".def". A fact file gets the data into BrainMaker. There are fact files for training, testing, and running. The default extension for the training fact file is ".fct", for testing it is ".tst", and for running it is ".in". A network file is created by BrainMaker during training using the data in the training fact file and the instructions in the definition file. The network file contains the actual connection information as well as training parameter information. The default extension for a network file is ".Net". The network file plus the testing fact file are used for testing. When the answers are not known, the trained network file plus the running fact file are used for running.

5.4 Neural Network Design, Training, and Testing

To develop a neural network model, one must decide precisely what the neural network is expected to forecast, generalize, or recognize. The original database needs some preprocessing before it can be used for training and testing. Then the neural network needs to be trained and tested with some certain rules. Finally, a new data set is used to validate the neural network.

5.4.1 The Framework for Neural Network Development

In this research, the BPNN model was developed through a procedure presented in Figure 5-2. Each developing stage shown in Figure 5-2 is discussed as follows:

- Database Review – Database investigations were performed to ensure that the database contained sufficient information for neural network development and for pavement crack forecasting.

- Data Preprocessing and Coding – The database was processed and reorganized to form a new database ready for model development. The new database was divided into several sub data sets to create a training data set, testing data set, and a validation data set.

- Network Design and Training – Several different architectures of the neural network were designed and trained in order to obtain the best architecture which was best on the testing performance.

- Network Testing and Error Analysis – Each architecture was tested independently using the testing data set. Assessments were made for generalization ability and accuracy.

- Network Implementation – The best network architecture was chosen and embedded in the working environment. Final testing was carried out within that environment.

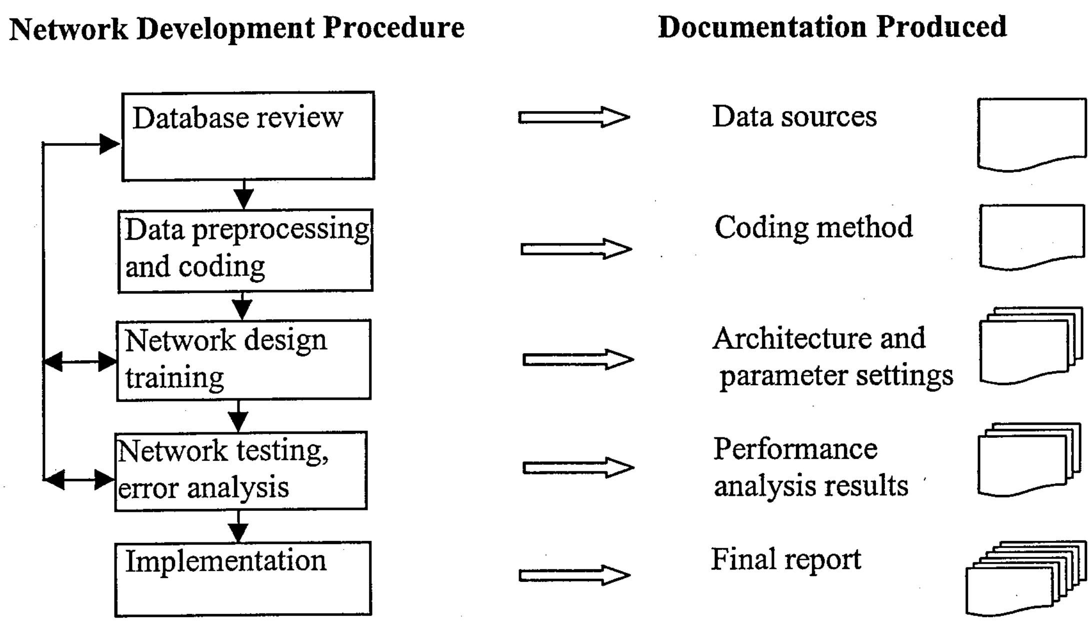


Figure 5-2. The Procedure for Neural Network Development

5.4.2 Neural Network Training and Testing

The specification of input and output for the BPNN is presented in Table 5-1. The BPNN was designed to forecast the CI for the next year, CI(t+1), given the CIs of previous three years, CI(t), CI(t-1), and CI(t-2). Training a neural network involves repeatedly presenting a set of examples (facts) to the network. The network takes in each input, makes a guess as to the output, checks this guess against the output (correct answer), and makes corrections to the internal connections (weights) if its guess is incorrect. This process is repeated for each fact in turn until the network learns the facts well enough to be useful.

TABLE 5-1. BPNN Architecture

Neuron Type	Neuron Number	Description	Range of Variables
Inputs	1	CI(t-2)	0-10
	2	CI(t-1)	0-10
	3	CI(t)	0-10
	4	Flexible or not	0 or 1
	5	Rigid or not	0 or 1
	6	Cycle	1-4
	7	Age	1-26
Output	1	CI(t+1)	0-10

In this research, it was found that approximately 70% of the total facts in the training set were easy-to-learn facts. However, training on the other 30% of the facts took as long as training on the first 70% of facts. Histograms and the Network Progress Display, two useful tools provided by BrainMaker, can help determine whether the network is making progress in training and still has the capacity to learn. Figure 5-3 shows the training histogram of a neural network (12 hidden neurons, 50 epochs) with the horizontal axis representing the values of connection weights, vertical axis representing the number of weights. This bell-shaped histogram indicates the network is healthy and still has the capacity to learn. Another tool is the Network Progress Display, as shown in Figure 5-4. The top part of the screen shows a histogram of the errors over a training run. It gives a quick snapshot of the distribution of errors, making it easy to see how close the network is to achieving the pre-specified tolerance level. The bottom part shows the progress of the Root Mean Square (RMS) error, which is defined in Chapter 3: Methodology, during training. This graph shows how well training is progressing over runs.

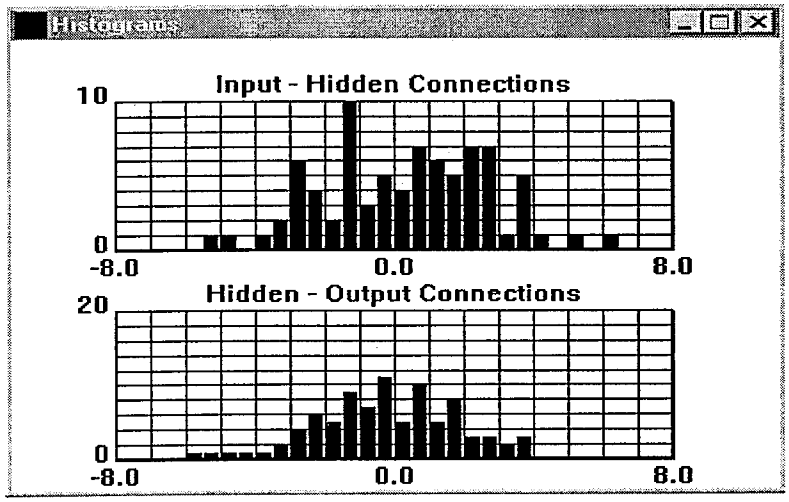


Figure 5-3. Connection Weights Histogram in Training

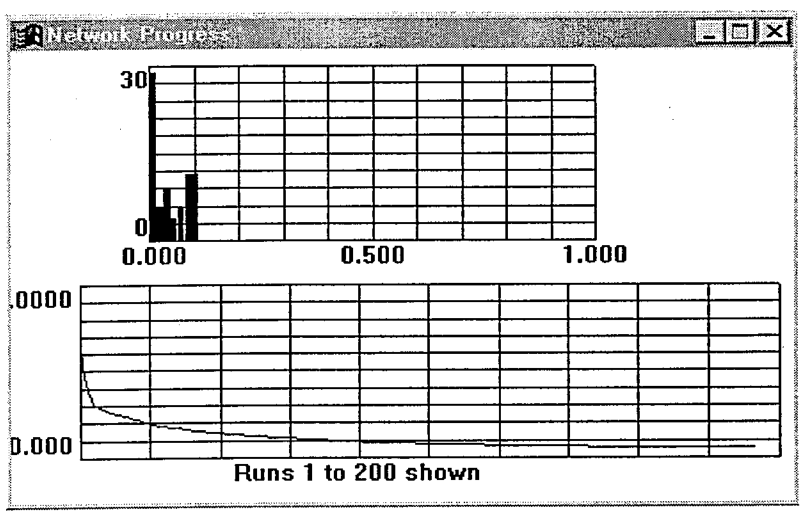


Figure 5-4. Network Progress in Training

In order to identify the best BPNN model for pavement crack condition forecasting, a variety of neural network architectures were experimented in this study. Table 5-2 presents the training and testing errors resulting from typical BPNN architectures. Since the generalization capability of the neural network is typically represented by the testing errors, the testing RMS error was selected as the major criterion to evaluate the BPNN performance. It can be seen from Table 5-2 that the input combination of CI(t-2), CI(t-1), CI(t), pavement type, maintenance cycle, and road age (model 9 presented in Table 5-2) resulted in the best BPNN model.

Table 5-2. Training and Testing Errors of Different BPNN Architectures

Model	Input & Output	Architec	$RMS_{TRAINING}$	$RMS_{TESTING}$
1	CI(t), CI(t-1), CI(t+1)	2-6-1	0.1063	0.0983
2	CI(t), CI(t-1), CI(t+1)	2-6-6-1	0.1590	0.1604
3	CI(t), CI(t-1), CI(t-2), CI(t+1)	3-7-1	0.0751	0.0774
4	CI(t), CI(t-1), CI(t-2), CI(t+1)	3-7-7-1	0.1031	0.1108
5	CI(t), CI(t-1), CI(t-2), CI(t-3), CI(t+1)	4-8-1	0.0802	0.0750
6	CI(t), CI(t-1), CI(t-2), CI(t-3), CI(t+1)	4-8-8-1	0.1227	0.1341
7	CI(t), CI(t-1), CI(t-2), type, CI(t+1)	5-9-1	0.0683	0.0699
8	CI(t), CI(t-1), CI(t-2), type, age, CI(t+1)	6-12-1	0.0654	0.0621
9	CI(t), CI(t-1), CI(t-2), type, age, cycle,	7-12-1	0.0631	0.0593*
10	CI(t), CI(t-1), CI(t-2), type, age, cycle,	8-18-1	0.0610	0.0593

5.4.3 Selection of Optimal Number of Hidden Neurons

Selection of optimal number of hidden neurons is an important issue in the neural network training process. The goal of training is to obtain a neural network with best generalization capability. Generalization is defined as the ability of a neural network to store in its weights general characteristics which are common to a group of examples. Usually, a neural network with too few hidden neurons will not be able to learn sufficiently from the training data set, whereas a neural network with too many hidden neurons will allow the network to memorize the training set instead of generalizing the acquired knowledge for future unseen examples. Unfortunately, there is no precise formula for determining the ideal number of hidden neurons needed for a given application. There are several ways to determine a good number of hidden neurons. One solution is to train several networks with varying numbers of hidden neurons and select the one that tests best. A second solution is to begin with a small number of hidden neurons and add more while training if the network is not learning. In this research, the first method was used to train several neural networks with varying number of hidden neurons. The neural network that resulted in the least testing error was selected, resulting in the best generalization capability. In the experiment procedure, a small number of hidden neurons were experimented as a starting point. Then gradually more neurons were added to the hidden layer. The "testing while training" method was used to trace the testing errors (generalization ability) of the neural network during training process. After training, it was convenient to find the best network with the least testing errors.

Figure 5-5 presents the RMS error and average error of the neural networks with different number of hidden neurons. It can be seen that the network with 12 neurons in the hidden layer results in the least RMS error and average error. The schematic architecture of the final neural network in this research is presented in Figure 5-6. This neural network contains an input layer with seven input neurons, one hidden layer with twelve hidden neurons, and an output layer with one output neuron.

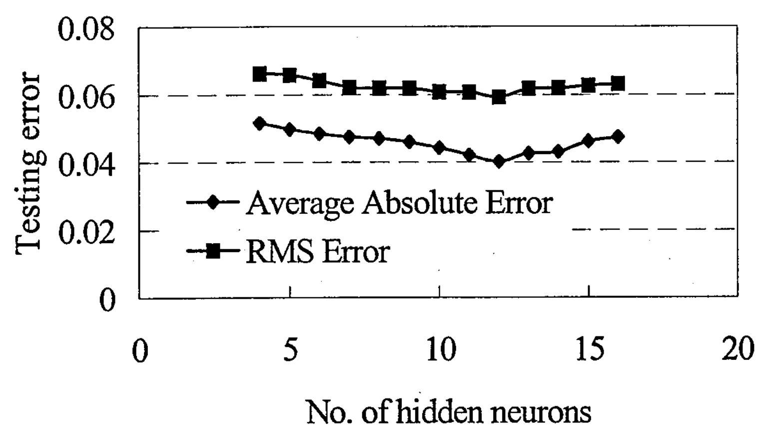


Figure 5-5. Testing Errors with Different Number of Hidden Neurons

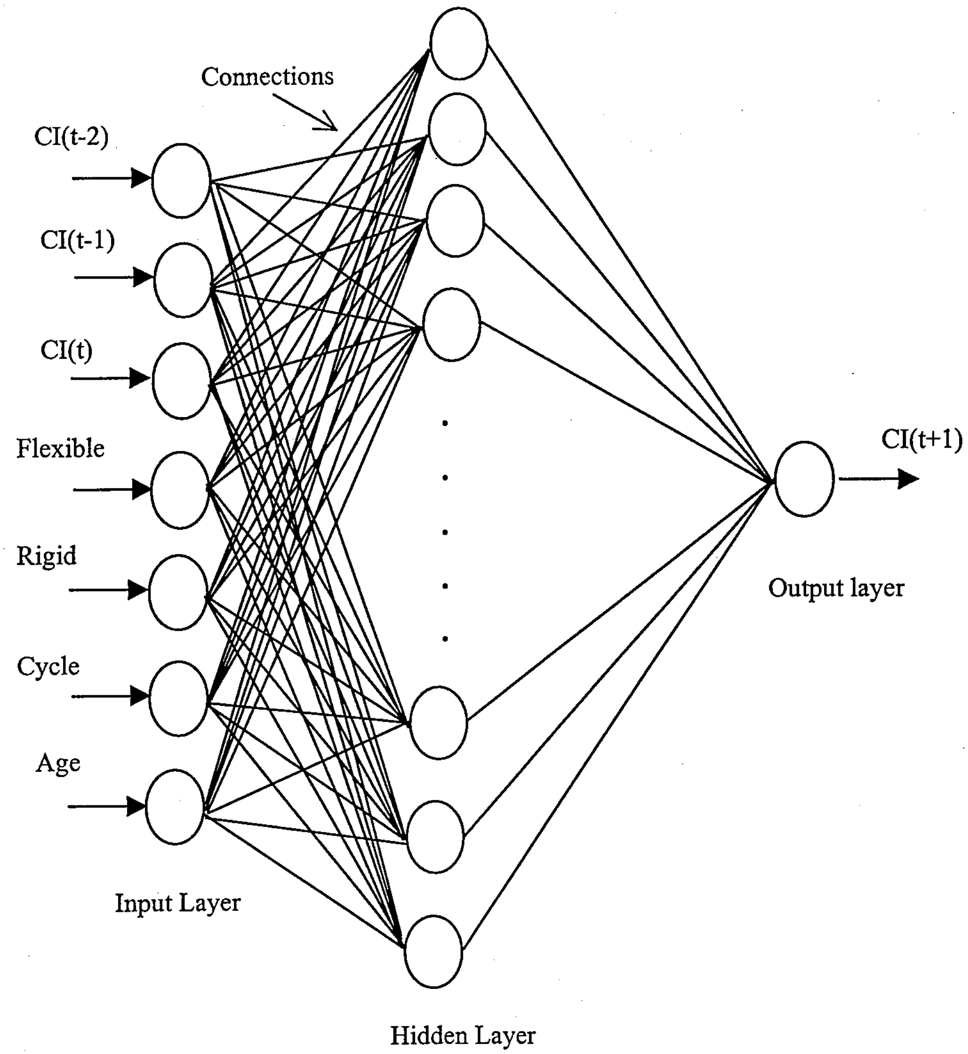


Figure 5-6. BPNN Architecture for 1-year Pavement Crack Forecasting

5.5 Development of Multi-Year Forecasting Models

The neural network model discussed herein was successfully applied to forecast the crack condition. However, some transportation agencies may be interested in multiple-year forecasting in order to develop long-term pavement maintenance plans. Accordingly, in this project, 2-year and 3-year neural network models were developed separately to extend the function of neural network model to forecast multi-year crack condition. The neural network model architectures for 2-year and 3-year forecasting are shown in Figures 5-7 and 5-8, respectively. In the 2-year model, crack indexes in previous 3 years, namely, CI(t-2), CI(t-1), and CI(t), are used to forecast crack index in the year after next year, or CI(t+2). After error examination and noise filtering, 79168 original records in the historical database, which resulted in 19792 training patterns, were utilized in the training process. Since the training mechanism is mainly based on pattern recognition, it was observed that the neural network was able to learn sufficiently from the given data set, resulting in reasonably acceptable error. The neural network model for the 3-year crack index forecasting follows the same logic as the 2-year model. 44152 original records in the historical database, which resulted in 11308 training patterns, were utilized to train the neural network. In order to facilitate the model implementation, the same input combinations were retained for both 2-year and 3-year BPNN models. Tables 5-3 and 5-4 present the training and testing errors of multi-year BPNN with different number of hidden neurons. It was found that the best BPNN architectures were 14 hidden neurons for the 2-year model (Architecture 7-14-1) and 21 hidden neurons for the 3-year model (Architecture 7-21-1).

Table 5-3. Training and Testing Errors of the 2-year BPNN Model

Architecture	$RMS_{TRAINING}$	$RMS_{TESTING}$
7-11-1	0.0901	0.1025
7-12-1	0.0743	0.0759
7-13-1	0.0614	0.0625
7-14-1*	0.0612	0.0619
7-15-1	0.0610	0.0634
7-16-1	0.0610	0.0639
7-17-1	0.0607	0.0682

* The best BPNN architecture.

Table 5-4. Training and Testing Errors of the 3-year BPNN Model

Architecture	$RMS_{TRAINING}$	$RMS_{TESTING}$
7-13-1	0.0965	0.1007
7-15-1	0.0897	0.0904
7-17-1	0.0853	0.0891
7-19-1	0.0762	0.0780
7-21-1*	0.0754	0.0767
7-23-1	0.0752	0.0771
7-25-1	0.0751	0.0775

* The best BPNN architecture

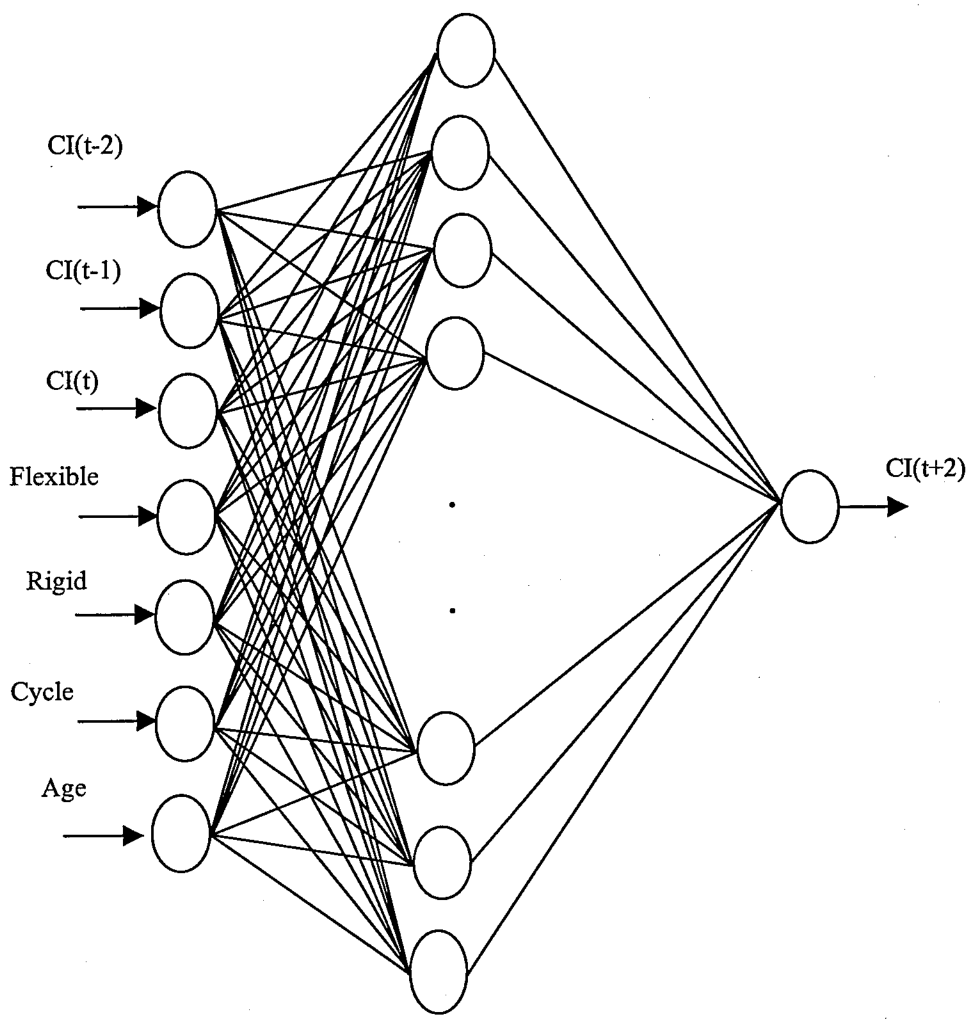


Figure 5-7. BPNN Architecture for 2-year Pavement Crack Index Forecasting

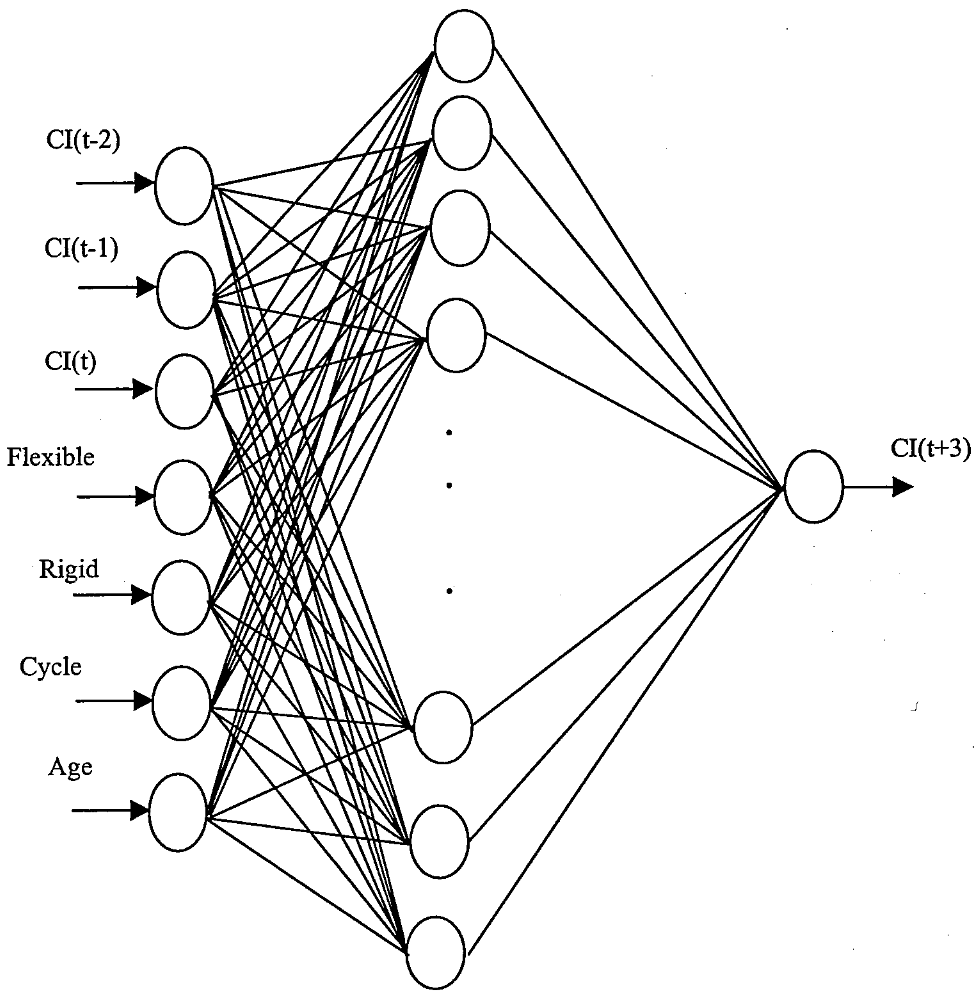


Figure 5-8. BPNN Architecture for 3-year Pavement Crack Index Forecasting

CHAPTER 6

PERFORMANCE ANALYSIS OF THE NEURAL NETWORK MODELS

After training, the pavement deterioration information was stored in the weights of the BPNN model, and the neural network had the capability to forecast future crack condition. The next step in this research was to evaluate the performance of the developed BPNN model. In order to validate the model with a new data set, the 1998 crack condition survey database was used as an independent data set, which was unseen by the BPNN during the training and testing process.

6.1. Comparison of Testing Errors

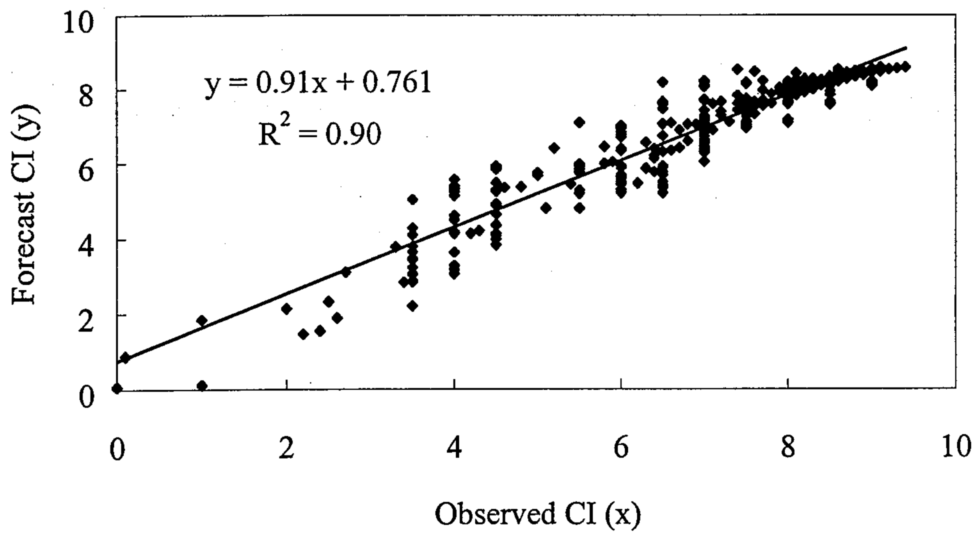
For comparison purpose, an AR model was developed to forecast CI as well. In the AR model, the CI in the next year, CI(t+1), was forecast by extrapolating CI's in the previous three years, CI(t), CI(t-1), and CI(t-2). The comparison results based on the testing data set are presented in Table 6-1. It can be seen that the BPNN model was more accurate than the AR model in terms of the RMS error and average error.

TABLE 6-1. Testing Errors of the BPNN Model and AR Model

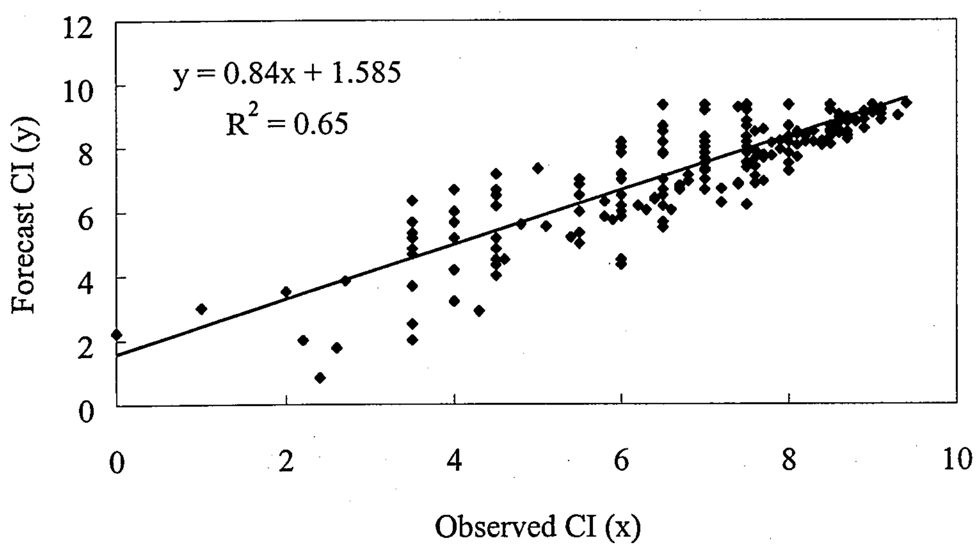
Model	Average Absolute Error	RMS Error
BPNN	0.0402	0.0593
AR	0.0692	0.1091

6.2 Goodness of Fit

Performance of the forecasting models was further assessed by plotting the forecast CI values and observed CI values on a correlation graph. Figures 6-1 and 6-2 depict the forecast CI values and observed CI values, with Figures 6-1 representing the results for flexible pavements and Figures 6-2 for rigid pavements, respectively. As shown in the figures, the BPNN model ($R^2 = 0.90$ for flexible pavement, $R^2 = 0.91$ for rigid pavement) outperformed the AR model ($R^2 = 0.65$ for flexible pavement, $R^2 = 0.72$ for rigid pavement). This result can also be found in the correlation equations in the form of $y = a\ x + b$, which are shown in the same figures. In this equation, y represents the forecast CI and x represents the observed CI. Under ideal conditions, the parameter a should approach 1 and b should approach 0. As seen in those figures, the a value resulting from the BPNN model (a = 0.91 for both flexible and rigid pavements) was closer to 1 than that from the AR model (a = 0.84 for flexible pavement and a = 0.85 for rigid pavement). On the other hand, the b value resulting from the AR model (b = 1.59 for flexible pavement and b = 1.48 for rigid pavement) was approximately twice of that from the BPNN model (b = 0.76 for flexible pavement and b = 0.68 for rigid pavement). It was found as well that there was a fairly good correlation between the observed and forecast CI values by the BPNN model when CI values were greater than 5, which covers the normal management range of FDOT. When CI values were less than 5, which indicates that the given pavement section requires major maintenance, differences between the two values were relatively larger.

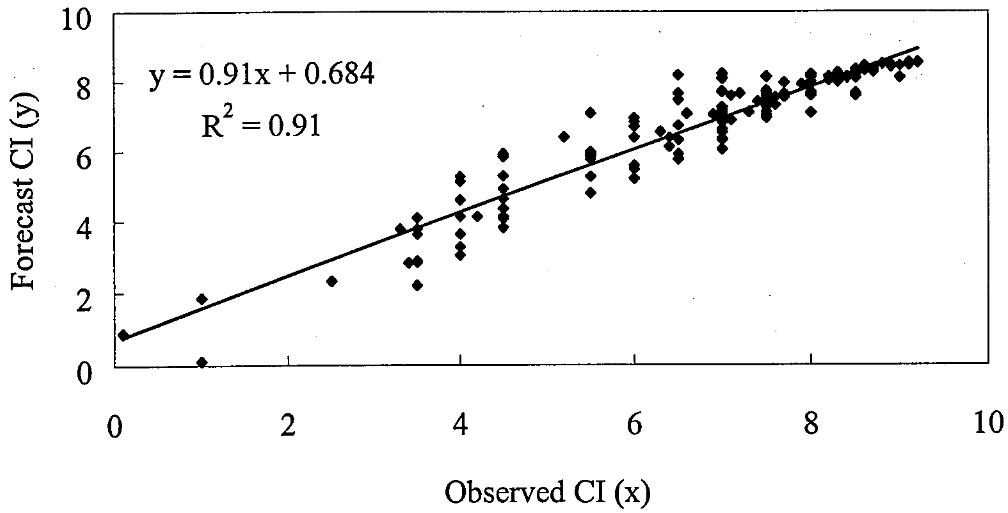


a) Forecast by BPNN Model

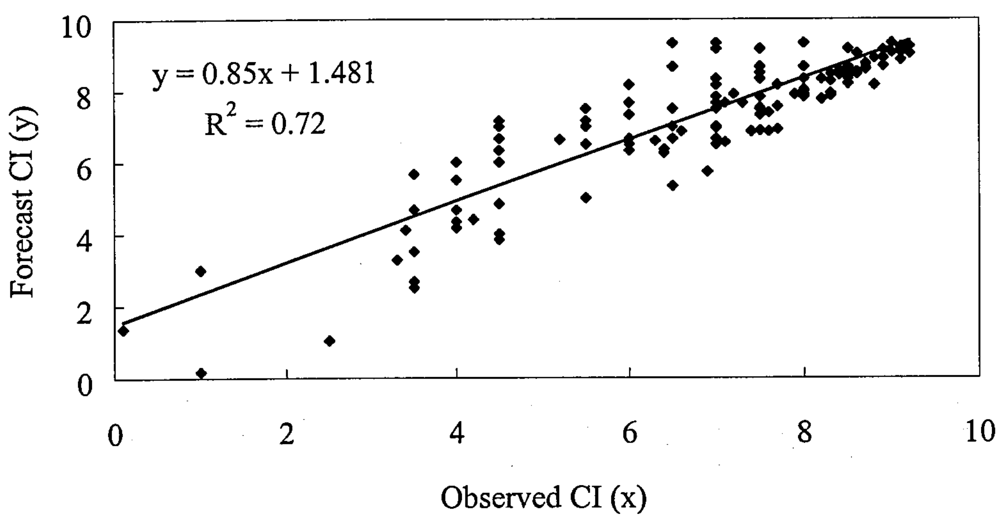


b) Forecast by AR Model

Figure 6-1. Goodness of Fit (Flexible Pavements)



a) Forecast by BPNN Model

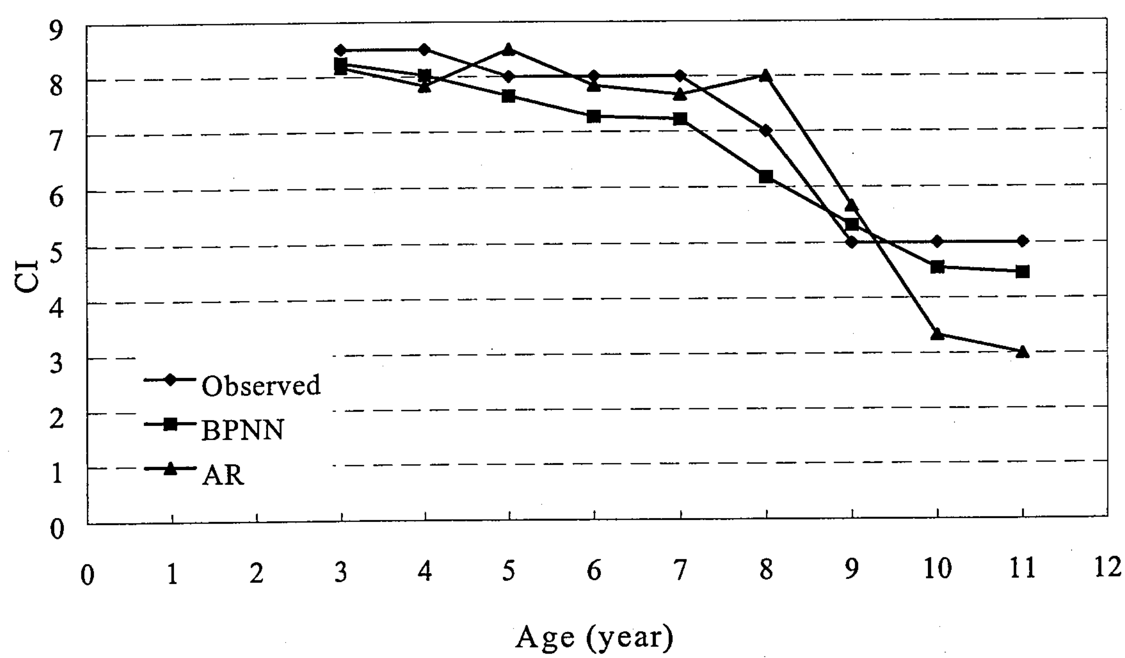


b) Forecast by AR Model

Figure 6-2. Goodness of Fit (Rigid Pavements)

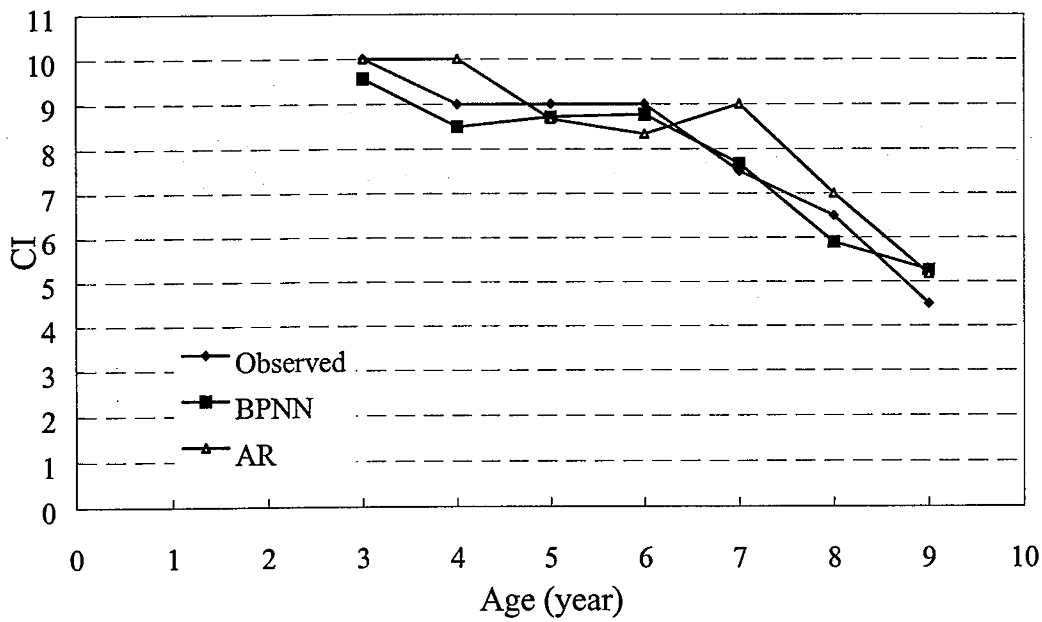
6.3 Case Study of Individual Sections

Several pavement sections, which were not included in the training process, were initially selected for the case study. CI's of each section during a specific cycle were forecast by the BPNN model and the AR model. The forecasts of two of the sections are plotted on the same graph as shown in Figure 6-3. It was found that the BPNN model and AR model had comparable forecasting accuracy during the initial years of pavement deterioration. However, the BPNN model outperformed the AR model as the crack condition began to deteriorate at a higher rate. It is obvious that the forecasts of the AR model tend to lag behind the observed values, as it occurs often in the conventional time-series models.



a) Section 1, Flexible, Cycle 2

Figure 6-3. Case Study of Individual Sections



b) Section 2, Rigid, Cycle 1

Figure 6-3. Case Study of Individual Sections

6.4 Model Validation with 1998 Data Set

6.4.1 Forecasting of Deficient Lane Miles for 1998

The BPNN model was validated by the 1998 data set, which was excluded from the training and testing process. In this validation step, deficient lane miles, which is defined as the total lane miles of pavement sections with CI equal to or less than six, were forecast by the BPNN model. The forecasts were then compared with those of AR model and the observed deficient lane miles. The comparison results are shown in Table 6-2. On a statewide basis, the forecasts of the BPNN model were closer to the observed values than those of the AR model. As seen in table 6-2, the BPNN model resulted in an error of 189.5 lane miles, while the AR model had an error of 351.1 lane miles. On a district-by-district basis, both models had comparable forecasting performance in Districts 3 and 7. However, the BPNN model forecasts had much better accuracy in Districts 1, 2, 4, 5, and 6.

TABLE 6-2. Comparison of Forecast and Observed Deficient Lane Miles for Year 1998 Using BPNN Model and AR Model

FDOT Districts	Forecast		Observed	Difference	
	BPNN	AR		BPNN	AR
1	1132.7	885.7	1057.3	75.4	-171.6
2	1498.7	1375.7	1540.0	-41.3	-164.3
3	1230.4	1210.0	1241.0	-10.6	-31.0
4	564.2	465.7	528.1	36.1	-62.4
5	1183.7	1058.5	1153.6	30.1	-94.7
6	199.9	243.6	148.8	51.1	94.8
7	377.4	405.8	327.7	49.7	78.1
Total	6187	5645	5996.5	190.5	-351.1

6.4.2 Error Analysis on A Section-by-Section Basis for 1998

Another validation was performed to compare the forecasts of these two models on a section-by-section basis. In this analysis, the forecast crack condition (deficient or not) was compared to the observed condition, on a section-by-section basis, to determine the forecasting errors. If the forecast crack condition does not match the observed condition for a given section, a forecasting error is detected. Hence, there are two types of errors to be identified: over-estimation error and under-estimation error. The former is defined as the lane miles, which are actually deficient but forecast as not deficient. The latter is defined as the lane miles, which are actually not deficient but forecast as deficient. These two types of errors of the BPNN model are summarized in Table 6-3, along with those of the AR model. It is noted that, on a statewide basis, the BPNN model had significant improvements in terms of overestimation and underestimation error, respectively, as compared to the AR model. Further, as shown in Table 6-3, the BPNN model gave more accurate forecasts than the AR model in most Florida districts. Because the forecasting accuracy directly affects the decision-making process and budget allocations in the PMS, these improvements in the forecasting capability are considered significant.

TABLE 6-3. Comparison of Over- and Under-estimates for Year 1998

FDOT Districts	BPNN						AR					
	Observed Deficiency	Over-Estimation	Observed Non-Deficiency	Under-Estimation	%Over-Estimation	%Under-Estimation	Observed Deficiency	Over-Estimation	Observed Non-Deficiency	Under-Estimation	%Over-Estimation	%Under-Estimation
1	1057.3	52.2	3164.7	127.6	4.93	4.03	1057.3	369.2	3164.7	197.6	34.91	6.24
2	1540.0	118.3	3716.5	77.0	7.68	2.07	1540.0	301.3	3716.5	137.0	19.56	3.69
3	1241.0	133.5	2496.1	122.9	10.76	4.92	1241.0	253.5	2496.1	222.5	20.43	8.91
4	528.1	28.6	3883.7	64.7	5.42	1.67	528.1	183.5	3883.7	121.1	34.75	3.12
5	1153.6	63.5	3478.8	93.6	5.50	2.69	1153.6	272.5	3478.8	177.8	23.62	5.11
6	148.8	2.7	1765.6	53.8	1.81	3.05	148.8	47.0	1765.6	141.8	31.59	8.03
7	327.7	29.6	2271.4	79.3	9.03	3.49	327.7	135.6	2271.4	213.7	41.38	9.41
Total	5996.5	428.4	20776	618.9	7.71	2.98	5996.5	1563	20776	1211.5	26.07	5.83

6.5 Multi-year Model Performance Analysis

For validation purposes, the 1998 observation data set was used again to validate the multi-year model performance. In this process, CI in 1998 (CI98) was forecast by the three developed BPNN models:

- CI95, CI96, and CI97 using the 1-year model.
- CI94, CI95, and CI96 using the 2-year model.
- CI93, CI94, and CI95 using the 3-year model.

The forecasts of 1998 deficient lane-miles by using these three models are presented in Table 6-4. It can be seen that the 1-year BPNN model resulted in better accuracy than the 2-year and 3-year models for most districts, and the errors generally increased with the length of forecasting period. As the forecasting period is getting longer, the hidden relationship between the past crack indices and the future condition becomes more hazy and difficult to learn, thus resulting in the less forecasting accuracy of the multi-year BPNN models. However, the multiple-year BPNN models developed can also describe the crack deterioration with reasonable accuracy, which would greatly benefit the long-term decision-making in pavement management.

Table 6-4. Model Comparison of Deficiencies Using 1998 Data Set

District	Observed Deficiencies	1-year Model	2-year Model	3-year Model
1	1057.3	1132.7	1236	1384.2
2	1540.0	1498.7	1523	1561.1
3	1241.0	1230.4	1267.7	1271.5
4	528.1	564.2	609.2	583.9
5	1153.6	1183.7	1280.9	1348.0
6	148.8	199.9	243.2	236.1
7	327.7	377.4	430.5	489.0
Total	5996.5	6187	6590.5	6873.8

CHAPTER 7

BPNN MODEL IMPLEMENTATION

The BPNN models discussed before resulted in encouraging forecasting results and hold great promise to be applied in DOT's pavement management practice. However, the BPNN is so complicated that special expertise is generally required to understand the neural network modeling mechanism. This chapter presents the forecasting results of 1999 and 2000 by the developed BPNN model. Then, the model implementation process is described, which could be used as a guideline of software development.

7.1 Forecasts of Pavement Deficiency during Year 1999 and 2000

Once trained, the two multi-year neural network models presented in Chapter 5 were used to forecast deficient lane-miles for Year 1999 and 2000, based on the crack conditions in 1995, 1996, and 1997. As shown in Table 7-1, on a state-wide basis, 8212 lane-miles out of 29515 lane-miles would be ranked as deficient for 1999 if no actions had been taken in 1998. On a district-by-district basis, District 2 would result in the highest deficient lane miles (1761 lane miles) in 1999 based on the 2-year forecasting model. The forecasts for Year 2000 were based on the 3-year neural network model. It was found that 11323 lane miles would become deficient if no maintenance activities were taken in 1998 and 1999, and District 2 would yield the highest deficient lane miles, which was consistent with the forecasts during 1999. The district-by-district deficient lane-miles for year 1998, 1999 and 2000 were also plotted in Figure 7-1. Combining these two forecasting models with the single-year model previously developed, FDOT would be able to make efficient maintenance project programming, budget allocating, and long-term pavement management plans based on crack condition.

Table 7-1. Forecasts of Deficient Lane Miles during Year 1998, 1999 and 2000

District	Total Lane Miles	1998		1999		2000	
		Deficient	%Deficient	Deficient	%Deficient	Deficient	%Deficient
1	4221.9	1057.3	25.0	1314.3	31.1	1736.2	41.1
2	5256.5	1540.0	29.3	1760.8	33.5	2485.7	47.2
3	3737.0	1241.0	33.2	1556.9	41.6	1939.4	51.9
4	7153.1	528.1	7.3	1500.2	20.9	2064.4	28.8
5	4632.4	1153.6	24.9	1386.9	29.9	1947.2	42.0
6	1914.	148.8	7.7	233.0	12.1	291.8	15.2
7	2599.1	327.7	12.6	460.0	17.7	857.8	33.0
Total	29514.4	5996.5	20.3	8212.1	27.8	11322.5	38.3

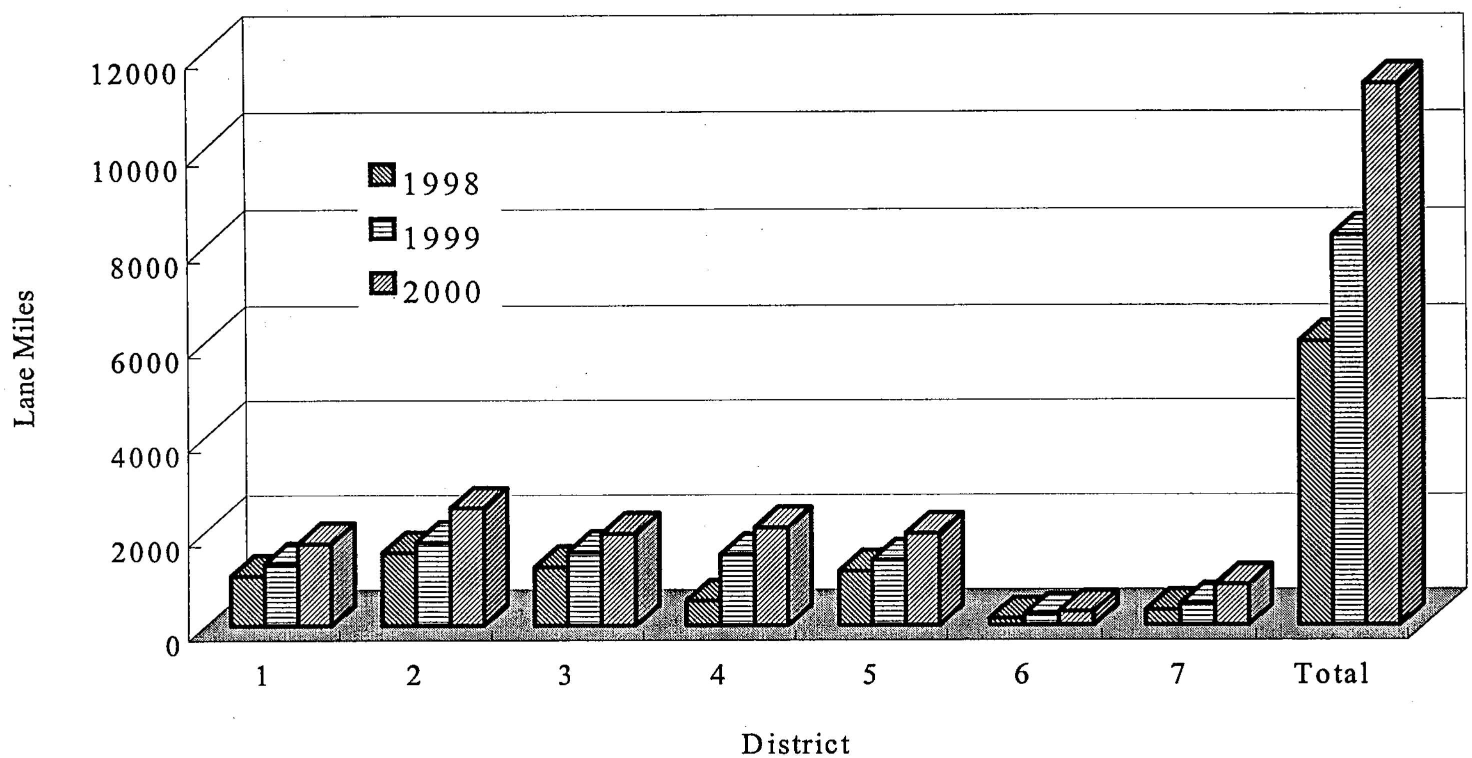


Figure 7-1. Deficient Lane Miles in 1998, 1999, and 2000

7.2 Consideration for Future Model Implementation

7.2.1 Software Architecture

The IPPP (Integrated Pavement Performance Prediction System) to be implemented contains three major operational modules: data preprocessing, BPNN analyzing, and data postprocessing, as shown in Figure 7-2. To coordinate these three major modules in the forecasting system, a main controlling interface is needed, which is suitably programmed by popular interface languages.

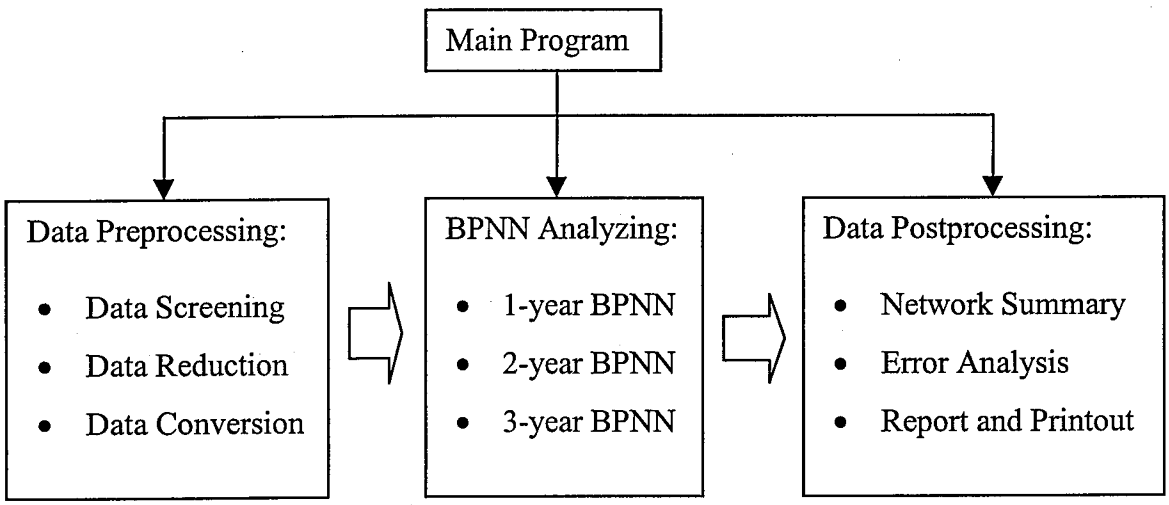


Figure 7-2. Software Architecture

7.2.2 Data Preprocessing Module

The original pavement performance survey database is the data source to be used as input to the BPNN model. However, these data cannot be utilized directly because of the potential errors and unsuitable data format. A data preprocessing module is required to handle all these problems, which includes four sub-modules: data screening, data reduction, data conversion, and data normalization.

- Data Screening: This module is used to examine and delete potential errors associated with the original database, such as missing values, unrealistic data, and duplicate records.

- Data Reduction: This module is used to reduce the original database to include only the variables needed for the specific task. The original database contains many administrative variables which have resulted in the huge size of the database. However, these variables are useless for the modeling process. The reduced data set includes all the information needed for the BPNN modeling and is easy to maintain and transport between programs.

- Data Conversion: This module is used to convert the data format in the original database into the suitable format required by the BPNN model. A conversion example is provided in Chapter 4 (See Figure 4-1 and Figure 4-2).

After running this module, the source data is cleaned, transformed, and ready to be used as input into the second module: BPNN analyzing module.

7.2.3 BPNN Analyzing Module

In this module, the users can select one of the three integrated BPNN models: 1-year model, 2-year model, or 3-year model depending on their specific task. The control is then switched to BrainMaker, the neural network platform. Within the BrainMaker environment, the user is provided convenient utilities to run the specified BPNN model. The output of this module would be written into a pre-specified file for use in the third module: data postprocessing module. Equations shown in Appendix A present the internal weights of all the three BPNN models. In those equations, W_1 is the weights matrix between the input and hidden layers, W_2 is the weights matrix between the hidden and output layers. These weights were obtained through the training procedure conducted in the project.

7.2.4 Data Postprocessing Module

The output of the BPNN model represents the crack condition of each road section in the year to be forecasted. Although these outputs are useful, DOTs are particularly interested in the summarized results, such as the state-wide deficient lane-miles or district-by-district analysis. In this module, the user may set up certain criteria to evaluate the pavement performance and select appropriate maintenance strategy. Further, the budget level for the highway network in each fiscal year can be estimated based on the forecasting results. Finally, the system can provide multiple printout formats and styles, which would facilitate the decision-making process and presentations to the public.

CHAPTER 8

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

8.1 Summary

This report documents the research that was conducted to develop new pavement performance forecasting models for the state of Florida. Pavement performance forecasting model is a key component in FDOT's PMS. At the project level, the model can help identify the maintenance needs and strategies for each roadway section. At the network level, the decision-makers can propose the annual budget and maintenance plans by using forecasts of these models. According to the literature review results, many highway agencies are now using regression models to forecast pavement performance. Although widely used and easy to understand, these models are less accurate due to the complexity of deterioration mechanism involved in pavement forecasting models. On the other hand, development of the traditional models needs a function form to be pre-specified. This could be difficult in the presence of a huge pavement performance database because of the multitude of variables associated. As an alternative, in this study, future pavement crack condition was forecast through the historical crack condition data by a neural network model, which does not require a pre-specified function form. Although complicated in the neural network training algorithm, the development of neural network is usually implemented by professional software packages. In this research, the BPNN model was developed with FDOT's pavement survey database by using BrainMaker, a popular neural network training platform. After the training process is completed, the BPNN model can remember all the necessary information in its weight matrix and is able to make forecasts of future crack condition with reasonable accuracy.

Further, the results from the BPNN model were compared with those from a commonly used AR model. Both models were utilized to forecast the deficient lane mileage for year 1998. The forecasts were then compared against the observation of deficient lane mileage in 1998. Finally, the 2-year and 3-year BPNN models were developed separately, which would meet the needs of FDOT in multi-year pavement performance forecasting and decision-making.

8.2 Conclusions

This study explains the development of a BPNN model for pavement crack condition forecasting. Based on the foregoing analysis, the BPNN model provided an effective alternative to the current pavement performance forecasting methods that exist.

The theoretical foundation of BPNN is ideal for pavement crack condition forecasting. By learning the dynamics of crack deterioration history, the BPNN model can extract the hidden information contained within the database and then make reasonable forecasts of crack condition in the future. This has been validated by the BPNN forecasts comparing to 1998 observations.

As found in this research, the original FDOT pavement condition survey database could not be used directly to develop the BPNN model. Accordingly, a data preprocessing procedure is required to screen, reduce, and transform the original database. This procedure is also required to run the developed BPNN model after it has been incorporated in FDOT's PMS. A series of computer programs would be suitable to perform these data preprocessing tasks automatically.

Although the BPNN training algorithm itself is complicated, some neural network software packages, such as BrainMaker, proved to be very effective in training and testing neural networks. The developed BPNN can be further connected to data pre- and post- processing modules. These three modules constitute the pavement crack condition forecasting system. Once implemented, all the modeling processes can be done automatically, while the user still can communicate effectively with the system through a user-friendly interface.

As seen in the comparison results with the testing data set, the BPNN model outperformed the AR model in terms of either average or RMS error. Further, based on the forecasts of deficient lane miles with 1998 validation data set, it was found that the errors of BPNN model were well distributed, with fewer overestimates and underestimates than the AR model, indicating appropriate generalization ability. Further, the BPNN model has been extended to be a capable of forecasting multi-year pavement crack performance with reasonable accuracy.

The BPNN model is easy to implement. For highway agencies other than FDOT, all that is required is to retrain the neural network with a region-specific database. Once trained, the BPNN model can make forecasts adaptive to local pavement conditions. On the other hand, while the BPNN model is developed for crack condition forecasting in this study, the same methodology can be used to forecast other pavement distress indices, such as roughness, rutting, and PCR, which is the lowest of the rut , crack and ride ratings.

Considering the above, it was concluded in this research that the proposed neural network model was an effective tool to be used in the pavement maintenance planning process in a PMS. Not only does it significantly improve the forecasting accuracy, but it can be easily updated by retraining with newly collected performance data. In view of the limitations of the existing models, neural networks offer an attractive alternative for the problem at hand. Pavement performance modeling based on historical database is likely to play a key role in PMS. We hope that an algorithm like BPNN will prove useful as a means of processing data and building useful mathematical models for pavement management.

8.3 Recommendations

In the previous chapters, the design, training, and testing of the BPNN model have been addressed. In order to incorporate this model into FDOT's PMS, which is written in SAS language, the three modules, namely, the data preprocessing, BPNN analyzing, and data postprocessing, should be implemented by certain computer programs, such as Visual Basic. It is suggested that the Dynamic Data Exchange (DDE) technique is used to exchange data between the PMS database and the BPNN software. It is recommended that the software package be developed to make the BPNN model practical.

The BPNN models were developed based on FDOT's historical crack condition survey database. Since the pavement crack deterioration pattern may change in the future, it is helpful to compare the model forecasts with the surveyed data each year. After several years, it is recommended to retrain the BPNN model with newly collected data, through which the neural network may keep learning the updated information and adjust its hidden weights to ensure the forecasting accuracy.

LIST OF REFERENCES

American Association of State Highway and Transportation Officials (AASHTO1993), *AASHTO Guide for Design of Pavement*, Washington D. C., 1993.

Asphalt Institute (AI) (1982), *Research and Development of the Asphalt Institute's Thickness Design Manual (MS-1)*, 9th Edition, 1982.

Attoh-Okine, N. O., "Predicting Roughness Progression in Flexible Pavements Using Artificial neural Networks", *Third International Conference on Managing Pavements, Conference Proceedings*, Vol. 1, 1994, pp.55-62.

Banan, M. R. and Huelmstad, K. D., "Neural Networks and AASHO Road Test", *Journal of Transportation Engineering*, Sep. 1996, pp358-366.

Chen, D. H., Zaman, M., and Laguros, J. G., "Assessment of Distress Models for Prediction of Pavement Service Life", *the 3rd Material Engineering Conference*, 1994, pp.1073-1080.

DSDOT, *Enhancement of DSDOT's Pavement Management System*. Final Report, Research Study SD93-14, December 1994.

Eldin, N. N. and Senouol, A. B., "A Pavement Condition-Rating Model Using Backprogagation Neural Networks", *Microcomputers in Civil Engineering* , 10, 1995, pp.433-441.

Eldln, N. N. and Senouol, A. B., "Condition Rating of Rigid Pavements by Neural Networks", *Canadian Journal of Civil Engineering*, 22, 1995, pp.861-870.

Florida Department of Transportation, *Pavement Condition Survey Handbook*, April 1994.

Fwa, T. F. and Chan, W. T., "Priority Rating of Highway Maintenance needs by Neural Networks", *Journal of Transportation Engineering*, Vol. 119, No. 3, 1993, pp. 419-431.

Hass, R. and Hudson, W. R., *Pavement Management Systems*, McGraw-Hill, Inc., 1978.

Hass, R., Hudson, W. R. and Zaniewski, J., *Modern Pavement Management*, Krieger publishing Company, Malabar, FL,1994.

Haykin, S., *Neural Networks -- -A Comprehensive Foundation.* Macmillan College Publishing Company, New York, 1994.

Horiki, K. I. and Fukuda, T., "Pavement Performance Model and Its Application to Pavement Management System", *Journal of Materials, Concrete Structures and Pavements*, 1994, pp.99-102.

Jackson, N. C., "Development of Pavement Performance Curves for Individual Distress Indexes in South Dakota Based on Expert Opinion", *Transportation Research Record 1524*, TRB, National Research Council, Washington, D.C., 1996, pp.130-136.

Karan, M. A. et al, "Determining Investment Priorities for Urban Pavement Improvements." *Journal of Association of Asphalt Paving Technology*, Vol. 45, 1976.

Lawrence, J. and Fredrickson, J., *BrainMaker User's Guide and Reference Manual 7th Edition.* California Scientific Software, Nevada City, CA, 1993.

Leahy, R. B., "Framework for Performance-based approach to mix design and analysis.", *Asphalt Paving Technology: Association of Asphalt Paving Technologists-Proceedings of the technical sessions*, Vol. 64, 1996, pp.431-473.

Lu, J., Bertrand C., Hudson, W. R., and McCullough, B.F., "Adaptive Filter Forecasting System for Pavement Roughness, *Transportation Research Record, 1344*, TRB, National Research Council, Washington, D.C., 1992, pp.124-129.

Mahoney, J., *Introduction to Prediction Models and Performance Curves*, Course Text, FHWA Advanced Course on Pavement Management, Nov. 1990.

Martin, T. C. and Taylor, S. Y., "Life-Cycle Costing: Prediction of Pavement Behavior", *Proceedings 17th ARRB Conference*, Part 6, pp.187-207.

Owusu-Ababio, S., "Modeling Skid Resistance for Flexible Pavements: A Comparison Between Regression and Neural Network Models", *Transportation Research Record 1501*, TRB, National Research Council, Washington, D.C., 1995, pp. 60-71.

Paterson, W. D., "Proposal of Universal Cracking Indicator for Pavements", *Transportation Research Board, 77th Annual Meeting, Preprint*, Washington, D. C., 1996.

Paterson, W. D. and Attoh-Okine, B., "Summary Models of Paved Road Deterioration Based on HDM-III", *Transportation Research Record 1344*, TRB, National Research Council, Washington, D.C., 1991, pp. 99-105.

Shahin, M. Y., *Pavement Management for Airports, Roads, and Parking Lots*, Chapman & Hall, New York, 1994.

Shekharan, A. R., "Effect of Noisy Data on Pavement Performance Prediction by Artificial Neural Networks", *Transportation Research Board, 77th Annual Meeting, Preprint*, Washington, D. C., 1998.

Shell International Petroleum, Co. Ltd., *Shell Pavement Design Manual*, 1978.

Sood, V. K., Sharma, B. M., Kanchan P. K., and Sitaramanjaneyulu K., "Pavement Deterioration Modeling in India", *Third International Conference on Managing Pavements, Conference Proceedings*, Vol. 1, 1994, pp.47-54.

Swingler, K., *Applying Neural Networks – A Practical Guide*. Academic Press, San Diego, CA, 1996.

Taha, M. A. and Hanna, A. S., "Evolutionary Neural Network Model for The Selection of Pavement Maintenance Strategy", *Transportation Research Board*, 74th Annual Meeting, Preprint 950192, Washington, D. C., 1995.

APPENDIX A-1

Weights Matrix of the 1-year BPNN Model (12 Hidden Neurons)

$$
W_1 = \begin{array}{cc}
 & \begin{array}{ccccccc} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} & w_{17} \end{array} \\
\begin{array}{c} w_{11} \\ w_{21} \\ w_{31} \\ . \\ . \\ . \\ . \\ . \\ . \\ . \\ w_{111} \\ w_{121} \end{array} &
\left[\begin{array}{ccccccc}
0.2812 & -3.7652 & -5.1740 & -4.3776 & -4.7532 & 6.0224 & 3.2308 \\
-4.6308 & -6.9532 & -7.5394 & -0.4994 & -0.4322 & 0.7074 & -2.7720 \\
3.9610 & -3.5970 & 7.2908 & -0.8842 & -0.7920 & -6.9086 & 1.8840 \\
3.3094 & -0.8708 & 1.3082 & -5.5436 & -4.7444 & -5.3242 & -7.3184 \\
-4.1426 & -3.8472 & -7.1336 & -1.5880 & -3.0696 & 6.1986 & -2.3700 \\
0.7160 & 3.6108 & -6.2826 & -4.4330 & -5.1302 & 3.8908 & 1.3872 \\
-3.1502 & -3.6232 & 6.3894 & 2.5630 & 1.0284 & 6.6102 & -5.8800 \\
1.2142 & -0.8204 & 6.6932 & 0.1292 & -0.1102 & 5.9514 & 0.9632 \\
0.6696 & -5.2820 & -5.0124 & -0.6862 & -2.4906 & -4.7972 & 0.5220 \\
4.8652 & -6.2986 & 1.7974 & -0.8656 & -0.9620 & 1.7830 & 1.0736 \\
-7.8800 & -6.7820 & 0.9110 & -0.3510 & -0.1480 & -6.1930 & 1.1252 \\
1.9656 & 1.2422 & 5.1032 & -0.1616 & -0.2524 & -2.8244 & 0.6580
\end{array}\right]
\end{array}
$$

a) Input-Hidden Weight Matrix

$$
W_2 = [-7.0702 \quad -7.9998 \quad 0.4454 \quad 2.2900 \quad -7.9998 \quad -0.4560 \quad -1.5266 \quad 2.6662 \quad -2.0190 \quad -0.4070 \quad -7.9998 \quad 0.9834\,]^T
$$

b) Hidden-Output Weight Matrix

APPENDIX A-2

Weights Matrix of the 2-year BPNN Model (14 Hidden Neurons)

$$
W_1 = \begin{bmatrix}
-6.6560 & -2.3454 & -1.4162 & -2.9736 & -7.5544 & -0.5702 & -2.9036 \\
-1.5962 & 0.1772 & -6.3164 & 4.7554 & 1.8456 & 4.9070 & -7.9998 \\
2.1660 & -0.9504 & -3.0102 & 3.8022 & 1.1074 & -3.4814 & 7.5772 \\
-1.9464 & 0.1766 & -7.9436 & -1.1350 & 0.1750 & 7.8780 & 5.4506 \\
-1.2806 & -3.3964 & -5.1490 & -1.6272 & -0.4992 & 0.7830 & 2.4614 \\
2.9474 & 2.9474 & -5.4206 & 0.3950 & 0.6312 & 2.9474 & 7.9994 \\
7.9994 & 7.9994 & 7.9994 & -7.9998 & -7.9998 & 7.9994 & -7.9998 \\
5.8314 & 2.3386 & -4.6914 & 0.7112 & -0.1810 & 1.4244 & -7.9998 \\
3.6210 & -4.0234 & -7.0190 & 4.5464 & -0.1432 & 3.1860 & 3.2916 \\
-3.1884 & -7.9910 & -7.9984 & 0.3742 & -4.9204 & -7.9920 & -1.9760 \\
-1.7554 & -0.2382 & -4.5420 & -1.1096 & -0.4182 & 6.2408 & -5.9382 \\
0.0576 & 0.4012 & 7.9872 & -5.8054 & 3.4022 & -6.1782 & -0.8056 \\
7.9994 & 7.9994 & 7.9994 & -7.5044 & 7.9994 & 7.9994 & 7.2474 \\
7.9994 & 7.9974 & 6.6930 & 1.0316 & 7.9970 & 7.9994 & 6.6442
\end{bmatrix}
$$

a) Input-Hidden Weight Matrix

$$
W_2 = [-7.9998\ \ -0.6906\ \ -4.2194\ \ -3.4332\ \ -7.9998\ \ -0.3814\ \ 0.2254
$$
$$
-1.4424\ \ -2.8390\ \ -7.9998\ \ -2.0554\ \ 4.7104\ \ -0.8084\ \ -2.4076]^T
$$

b) Hidden-Output Weight Matrix

APPENDIX A-3

Weights Matrix of the 3-year BPNN Model (21 Hidden neurons)

$$W_1 = \begin{bmatrix} 0.0044 & 5.7346 & 5.3916 & -0.8282 & -0.5584 & -4.9822 & -7.8404 \\ -0.1126 & 5.6792 & -2.9012 & -2.0234 & -2.8860 & 0.2956 & 6.4124 \\ -5.2350 & -2.0296 & -3.5040 & -7.2880 & -0.7052 & 0.7130 & -3.5554 \\ -2.8364 & 3.5116 & -6.7820 & 7.9994 & -5.5282 & 5.1336 & 7.9994 \\ -7.9330 & -0.6182 & -4.3780 & -3.3910 & -3.8266 & -6.9304 & -7.9998 \\ 7.9994 & 7.9994 & 7.9994 & -4.2860 & 5.7966 & 7.9994 & -7.9998 \\ 0.6434 & 2.3872 & -6.5266 & 6.3144 & 2.3354 & 3.1382 & -3.7044 \\ -7.9330 & -0.6782 & 1.9556 & 5.5640 & -4.8284 & -6.7416 & -7.9998 \\ 2.6692 & -0.0962 & 7.9994 & 7.8376 & -1.3116 & 7.2536 & 6.7446 \\ -7.9998 & -7.9998 & -7.9998 & -0.0950 & -7.9998 & -7.9998 & -7.9998 \\ 1.2106 & -6.3460 & -0.2586 & 7.7200 & 2.6696 & 0.6292 & -0.4020 \\ 0.2984 & 0.7140 & 7.9672 & 2.5540 & 0.5686 & -7.7682 & -7.9998 \\ -0.2442 & 4.4874 & -7.6276 & 1.7826 & 0.8544 & 1.5566 & -6.2390 \\ 7.9994 & 7.9994 & 5.4524 & 4.0708 & 5.0126 & 7.9994 & -7.9998 \\ -0.2116 & 1.8106 & -4.3806 & -2.1462 & 0.1630 & 0.6854 & 7.9994 \\ -7.9998 & -7.9998 & -7.9998 & 1.2570 & -7.9998 & -7.9998 & -7.9998 \\ 7.9994 & 7.9994 & 7.9994 & 7.9994 & -2.3966 & 7.9994 & -7.9998 \\ -4.2624 & -1.8686 & 3.8276 & -0.5030 & -1.2320 & -5.0242 & -1.5452 \\ 3.0094 & -1.4306 & -4.7382 & 6.3030 & 1.3166 & 3.5310 & -1.9606 \\ 3.4782 & -3.0452 & -1.3362 & 3.2660 & -0.6266 & 1.0122 & -7.9998 \\ 7.9994 & 7.9994 & 7.9994 & -0.5936 & 1.9930 & 7.9994 & -6.2390 \end{bmatrix}$$

a) Input-hidden Weight Matrix

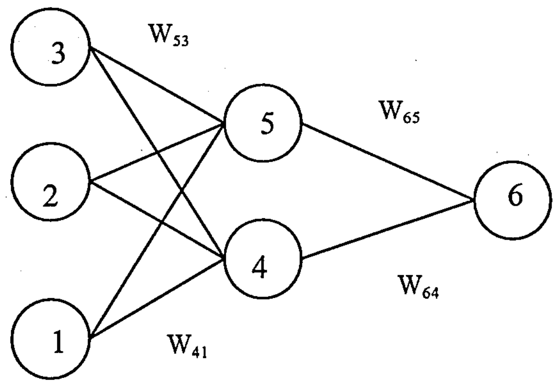
$$W_2 = [2.7504\ \ -0.8280\ \ -1.5106\ \ -0.7070\ \ 7.9994\ \ -0.3554\ \ 0.1004$$
$$7.9994\ \ -0.7850\ \ 0.0008\ \ -0.0946\ \ 2.3950\ \ -0.5986\ \ -0.0822$$
$$-0.8940\ \ 0.0276\ \ -0.3554\ \ -0.2708\ \ -0.3434\ \ -0.4816\ \ -0.7536]^{T}$$

b) Hidden-Output Weight Matrix

APPENDIX B

An Example of BPNN Training Process

In this example, the BPNN architecture is a 3-layer neural network as shown in the figure:



Training Data Set:

$$x = \begin{bmatrix} 9 & 8 & 7 & 5 \\ 10 & 10 & 9 & 8 \\ 7 & 6 & 6 & 5.5 \\ & \dots & & \end{bmatrix}$$

After normalization:

$$x = \begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.5 \\ 1 & 1 & 0.9 & 0.8 \\ 0.7 & 0.6 & 0.6 & 0.55 \\ & \dots & & \end{bmatrix}$$

Initial Weight Matrix (generated by random distribution):

$$W_1 = \begin{bmatrix} -0.6076 & 0.4214 & 1.6124 \\ 0.0030 & -0.3536 & 0.3980 \end{bmatrix}$$

$$W_2 = [0.6584 \ \ -2.1106]^T$$

For the first pattern in the training set [0.9 0.8 0.7 0.5],

a) Forward Pass

$a_{51} = w_{51}x_{11} + w_{52}x_{21} + w_{53}x_{31} = 0.0030*0.9 + (-0.3536)*0.8 + 0.3980*0.7 = -0.0016$

$a_{41} = w_{41}x_{11} + w_{42}x_{21} + w_{43}x_{31} = (-0.6076)*0.9 + 0.4214*0.8 + 1.6124*0.7 = 0.9190$

Assuming gain = 1,

$$O_{51} = \frac{1}{1+e^{-a_{51}}} = \frac{1}{1+e^{0.0016}} = 0.4996$$

$$O_{41} = \frac{1}{1+e^{-a_{41}}} = \frac{1}{1+e^{-0.9190}} = 0.7148$$

$a_{61} = w_{65}O_{51} + w_{64}O_{41} = (-2.1106)*0.4996 + 0.6584*0.7148 = -0.5838$

$$O_{61} = \frac{1}{1+e^{-a_{61}}} = \frac{1}{1+e^{0.5838}} = 0.3581$$

b) Backward Pass (Weight Adjustments)

For transfer function

$$O = f(a) = \frac{1}{1+e^{-ga}}$$

$f'(a) = O(1-O)$, so

$$\delta_{kp} = (T_{kp} - O_{kp})f'(a_{kp}) = (T_{kp} - O_{kp})O_{kp}(1 - O_{kp})$$

Output Layer Weight Adjustment: (Assuming $\alpha = 0.9$)

$\delta_{61} = (0.5 - 0.3581) * 0.3581 * (1-0.3581) = 0.0326$

$\Delta_1 w_{65} = \alpha\delta_{61}O_{51} = 0.9 * 0.0326 * 0.4996 = 0.0147$

$\Delta_1 w_{64} = \alpha\delta_{61}O_{41} = 0.9 * 0.0326 * 0.7148 = 0.0210$

Hidden Layer Weight Adjustment: (Assuming $\alpha = 0.9$)

$$\delta_{jp} = f'(a_{jp})\sum_k \delta_{kp}w_{kj} = O_{jp}(1 - O_{jp})\sum_k \delta_{kp}w_{kj}$$

In this example, since only one neuron in the output layer, k=1.

$\delta_{51} = O_{51}(1 - O_{51})\, \delta_{61}w_{65} = 0.4996 * (1-0.4996) * 0.0326 * (-2.1106) = -0.0172$

$\delta_{41} = O_{41}(1 - O_{41})\, \delta_{61} w_{64} = 0.7148 * (1-0.7148) * 0.0326 * 0.6584 = 0.0044$

$\Delta_1 w_{53} = \alpha\delta_{51}O_{31} = 0.9 * (-0.0172) * 0.9 = -0.0139$

$\Delta_1 w_{52} = \alpha\delta_{51}O_{21} = 0.9 * (-0.0172) * 0.8 = -0.0124$

$\Delta_1 w_{51} = \alpha\delta_{51}O_{11} = 0.9 * (-0.0172) * 0.7 = -0.0108$

$\Delta_1 w_{43} = \alpha\delta_{41}O_{31} = 0.9 * 0.0044 * 0.9 = 0.0036$

$\Delta_1 w_{42} = \alpha\delta_{41}O_{21} = 0.9 * 0.0044 * 0.8 = 0.0032$

$\Delta_1 w_{41} = \alpha\delta_{41}O_{11} = 0.9 * 0.0044 * 0.7 = 0.0028$

After adjustmets, the weight matrix would be:

$$W_{1,new} = \begin{bmatrix} -0.6048 & 0.4246 & 1.6160 \\ -0.0078 & -0.3660 & 0.3841 \end{bmatrix}$$

$$W_{2,new} = [0.6794 \;\; -2.0959]^T$$

This adjustments continue until all the patterns are fed to the BPNN, which is called an epoch. The training process would continue for many epochs until the BPNN converges.