2012

Artificial Neural Network Application In Environmental Engineering.

Xiaojue Tao
North Carolina Agricultural and Technical State University

ARTIFICIAL NEURAL NETWORK APPLICATION IN ENVIRONMENTAL ENGINEERING

by

Xiaojue Tao

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Department: Civil, Architectural and Environmental Engineering
Major: Civil Engineering
Major Professor: Dr. Shoou-Yuh Chang

North Carolina A&T State University

Greensboro, North Carolina

2012

School of Graduate Studies
North Carolina Agricultural and Technical State University

This is to certify that the Master's Thesis of

Xiaojue Tao

has met the thesis requirements of
North Carolina Agricultural and Technical State University

Greensboro, North Carolina
2012

Approved by:

Dr. Shoou-Yuh Chang
Major Professor

Dr. Manoj K. Jha
Committee Member

Dr. Sameer A. Hamoush
Committee Member

Dr. Sameer A. Hamoush
Department Chairperson

Dr. Sanjiv Sarin
Associate Vice Chancellor for Research and
Dean, School of Graduate Studies

BIOGRAPHICAL SKETCH

Xiaojue Tao was born on September 19th, 1982, in Minhang, Shanghai, China, to Honggen Tao and Shunmei Yu. In 2006 she received the Bachelor of Art degree in Environmental Art Design from Donghua University in Shanghai, China. She is a summa cum laude candidate for the Master of Science degree in Civil Engineering.

DEDICATION

This thesis is dedicated to my husband, Wen Fang, my father, Honggen Tao, my mother, Shunmei Yu, and my lovely daughter, Emily T. Fang, for their love and encouragement.

ACKNOWLEDGMENTS

I would like to thank Dr. Shoou-Yuh Chang for being my advisor and for his guidance and wisdom throughout the long-term commitment. His knowledge and experience helped me accomplish my goals.

I also would like to thank Dr. Sameer A. Hamoush, and Dr. Manoj K. Jha for being on my thesis committee. My thanks to all the faculty and staff members of the Department of Civil, Architectural and Environmental Engineering at North Carolina Agricultural and Technical State University. Their support and help made this research run smoothly. This work was sponsored by the Department of Energy Samuel Massie Chair of Excellence Program under Grant No. DF-FG01-94EW11425.

Finally, I offer words of thanks and gratitude to all of my colleagues for their friendship, support, and assistance. I especially thank Mr. Roger Dodson and Mrs. Poly Dodson for smoothing the words in the thesis.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE **PAGE**

LIST OF TABLES

ABSTRACT

Tao, Xiaojue. ARTIFICIAL NEURAL NETWORK APPLICATION IN ENVIRONMENTAL ENGINEERING**. (Major Professor: Dr. Shoou-Yuh Chang)**, North Carolina Agricultural and Technical State University.

The objective of this thesis research is to apply two artificial neural network (ANN) methods, back-propagation neural network (BPN) and radial basis function generalized regression neural network (RBFGRNN) in two environmental engineering case studies to explore their ability to modeling the complex environmental engineering systems. The traditional environmental engineering systems modeling are frequently using the physical-based modeling methods. Their performance is decided by the quantity of samples and quality of sampling methods, and it is also based on the physical laws they obeyed and the system knowledge they explored. But ANN offers a unique and alternative solution to bridge the cause and effect without knowing the detailed relationship between each other.

Two case studies are used to verify the performance of ANNs, landfill leachate flow rate modeling in Greensboro and total phosphorus concentration modeling in Te-Chi reservoir. The testing coefficient of determination R^2 of BPN applied in landfill leachate flow rate modeling is 0.728 and that in total phosphorous concentration modeling is 0.992. The testing coefficient of determination R^2 of RBFGRNN applied in landfill leachate flow rate modeling is 0.823 and in total phosphorous concentration modeling is 1. These results proved the ANNs are qualified to model complex environmental engineering systems modeling problems.

CHAPTER 1
INTRODUCTION

Modern society requires a highly secure degree of environment safety as a prerequisite for sustainable development, and environmental engineering is a key factor to meet this demand. However, the knowledge of the environmental system is limited, and most of the studies of environmental system modeling methods are based on the physical laws, called physically-based modeling. Generally, environmental engineers and researchers applied methods in this category to aid in decision-making, estimation, and prediction. However, the performance of the physically-based modeling method is dependent on the universal knowledge of study area. It includes climate information, geological conditions, human activities, and other related data sets as input parameters. It is an inherent issue of applying these physical-based modeling methods. Because of the practical difficulties of representing all the natural complexity and available measurements, it may not fit the physical law well. The model results are subject to a large number of uncertainties. The implication of these uncertainties is particularly significant when the models are used in practical applications for prediction or extrapolation purposes under varying environmental conditions. Also some physical laws are only tenable under some restricted conditions. When the study area expands to a very large scale, it is doubtful whether the per-defined physical laws are still tenable or not. As a result, using the available pieces of information together with the alternative modeling method, which is capable of directly establishing the complex nonlinear mapping between input and output without knowing the physical relationship, is crucial and effective for reducing the prediction or extrapolation errors caused by these uncertainties.

Currently, it is impossible to eliminate uncertainties from physically-based models due to the difficulties mentioned above, especially the uncertainties caused by inherent random process or variability of physical process. In the past, physically-based models were the only qualified modeling methods in environmental engineering fields, such as the hydrologic evaluation of landfill performance (HELP) model in landfill hydrology studies, MODFLOW in 3D subsurface ground water flow studies, and so on. Because of difficulties of measuring the model required data directly, many studies conducted research on model calibration and parameter estimation to improve the modeling accuracy (Zimmerman et al., 1998; Hill and Tiedeman, 2007). But statistically-based model calibration cannot guarantee the modeling accuracy as it may not be aware of the potential uncertainties in the system, even if the model bias and predictive uncertainties is reduced by using proper model and calibration method. Furthermore, the even a well-calibrated model may be developed based on insufficient samples or oversimplification, and it will result in an ‘ill-posed’ problem, which will yield an unstable system. With the development of sensoring technology, the sampling methods were strengthened, and related physically-based model performance was relatively more accurate. This causes another problem, which increases the cost of data collection. Meanwhile, it still did not overcome its major disadvantages, which are intensive data requirements; need to determine large number of parameters; and difficulties in finding the best set of calibration parameters.

Compared with the physically-based modeling method, the highlight of data driven approaches is the modeling of a desired system output (but not necessarily of the mechanics of the system) using historical data. Such approaches encompass “conventional” numerical algorithms, like linear regression or Kalman filters, as well as algorithms that are commonly found in the machine learning and data mining categories (Goebel and Saha, 2007). The latter data-driven approaches include fuzzy logics, genetic algorithms, artificial neural networks, and other approaches. A survey (Schwabacher, 2005) provides an extensive overview over data-driven methods in the context of computational intelligence.

The purpose of this master thesis research is to apply artificial neural networks (ANNs) as an alternative approach for quantifying the cause-and-effect relationship in different environmental systems. As a data-driven based technique, the advantages of ANN can be itemized as (Tu, 1996):

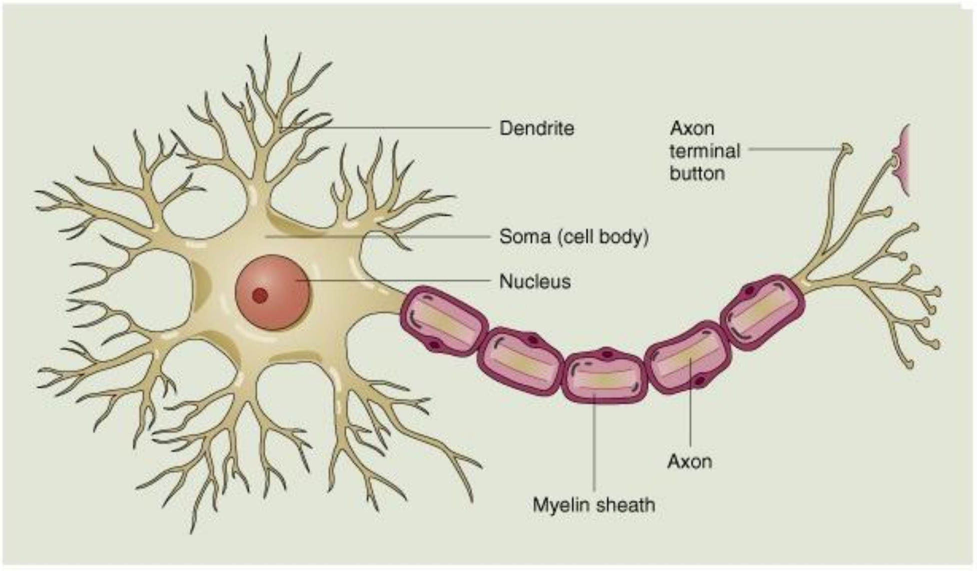
- Requiring less statistical training
- Ability to implicitly detect complex nonlinear relationships between dependent and independent variables
- Ability to detect all possible interactions between predictor variables
- The availability of multiple training algorithms

ANN is the group name of information processing systems, which mimic the metaphor of how biological nervous system operates. Generally, ANNs are composed of a large number of highly interconnected processing elements (PEs) working in unison to solve specific problems. Based on different learning algorithms applied, ANNs can be distributed to different taxonomy. The second chapter of this study will introduce the fundamental knowledge and structure of ANN and the single layer proceptron neural network (SLPNN) will be presented. Chapter 3 and 4 will give in depth presentations on two advanced neural networks, back propagation neural network (BPN), and the radial basis functional generalized regression neural network (RBFGRNN), which will be used to testify the ANN abilities of modeling the environmental engineering systems. Also a new clustering method for seeking the centers applied in the RBFGRNN will be introduced. After that, two study cases will be presented, which are leachate flow rate modeling in a municipal solid waste (MSW) landfill site at Greensboro, NC and total phosphorus concentration modeling of Te-Chi reservoir at Taiwan. Performance comparison between two different advanced neural networks will be made and also discussion and conclusion of the experiments and findings will be presented in the last chapter.

CHAPTER 2
INTRODUCTION TO ARTIFICIAL NEURAL NETWORK (ANN)

While developing an environmental system model, the features will be assigned degrees of importance based on past experience, physical laws, and other known and applicable information which has the cause-and-effect relationship with the current task and generalizations. Once the system model is derived, it is required to be a generalized application, which can be distributed to any similar system or predict the future status of the current task. As a result, the modeling method will be a dynamic and complex learning mechanism that utilizes both historical and environmental data. In biological level, this mechanism can be fully represented by how human brain operates. The human brain contains trillions of neurons with specific functions, and it can be described as a complex and parallel machine composed of trillions of processing elements. The figure below shows the structure of a biological neuron and its components.

There are four fundamental components that make up the composition of a neuron: dendrite, soma, axon, and axon terminal button (synapses). As shown in Figure 2.1, dendrites receive the bio-electronic signals and sent to the soma, the nucleus creates the response to the input signal and distributes to the synapses via the axon. The neuron is capable of achieving acquired knowledge for future use, while obtaining new knowledge to be processed. Massive biological neural networks of immense complexity can be created within the brain based on the neuron's capabilities and its simplistic structure. Artificial neural networks are algorithms that mimic the metaphor of the biological neuron or its combinations.



© 2000 John Wiley & Sons, Inc.

Figure 2.1 Architectural Graph of Biological Neuron

The single layer proceptron neural network (SLPNN) emulates the biological neuron and it is a fundamental sample of artificial neural networks. Figure 2.2 depicts its architectural graph. X represents an input sample with n characteristics, x_1, x_2,…, x_n, and a bias,x_0≡1. These n+1 features are assessed of their importance by n+1 dimensional weight matrix, *W*, and emerge to a final output by passing through an activation function or a linear summation layer, *Σ*. The whole process can be stated in the mathematic form as following equations:

$$I = W^T \cdot X \tag{2.1}$$

$$Y = T(I) = \frac{1}{1+e^{-\alpha \cdot I}} \tag{2.2}$$

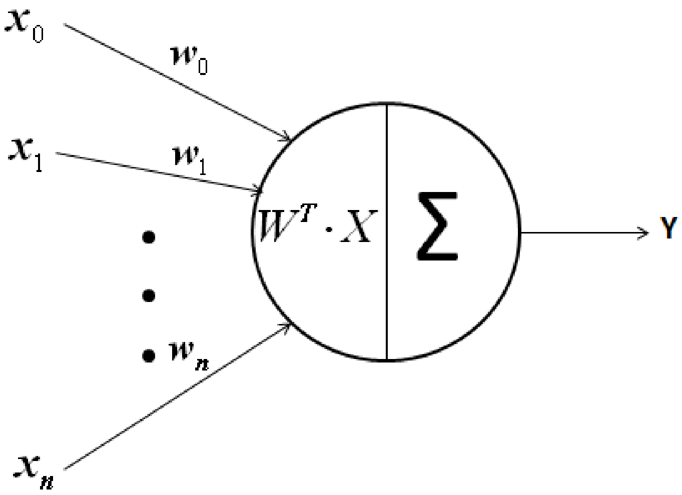


Figure 2.2 Architectural Graph of Single Layer Proceptron Neural Network

The activation function determines whether the neuron will be activated, which depends on the momentum, α. It also can be replaced by “IF…THEN” command, and the soft limiter switches to a hard limiter as a result.

Modeling a dynamic system by using artificial neural networks will required two separate portions, training section and testing section. The training section is a learning processing, and the testing section is aim to validate the training performance. Different samples are applied in two sections to testify its generalization ability.

Single layer proceptron neural network applies an error feedback criterion to improve the modeling performance by adjusting the existing weights. If error is feedback, the old weights will be replaced by new weights as following equations:

$$\Delta W = \beta \cdot \frac{X}{\|X\|} \cdot \left(Y_{desired} - Y_{network\_output} \right) \tag{2.3}$$

$$W_{new} = W_{old} + \Delta W \tag{2.4}$$

where β is the learning rate. Equation 2.4 is called the delta rule, which is often applied in artificial neural network training. Once ΔW is significant small or equal to zero, the training section of single layer proceptron is finished (Haykin, 1998). Technically, single layer proceptron only can solve the linear separable problems unless the input feature space is expanded.

A number of different artificial neural networks have been developed with different structures, paradigms, and learning rules. The structures are defined in the ways how to connect layers. Layers have different functions and contain one or more neurons that process the same input information in parallel.

In this research, two types of artificial neural networks will be applied to model the environmental engineering systems. First, I will introduce the back-propagation neural network (BPN) in chapter 3, which is a supervised multiple layers proceptron utilizing the back-propagation algorithm. Second, the radial basis functional generalized regression neural network (RBFGRNN) will be presented in chapter 4, which is a generalized linear regression model with nonlinear input space transformation technology achieved by supervised selection of centers.

CHAPTER 3
BACK-PROPAGATION NEURAL NETWORK (BPN)

Werbos (1974) established the back-propagation algorithm and proposed the concept of hidden layers. However, this work went largely ignored until the development of back-propagation algorithm was reported by Rumelhart et al. (1986). This report has been a major influence in the use of back-propagation learning, which has emerged as the most popular learning algorithm for the training of multilayer perceptrons (Haykin, 1998).

3.1 Bio-directional Signal Flow

BPN is a type of multilayer perceptrons that applies the BP algorithm for network training. Figure 3.1 shows the architectural graph of a multilayer perception with one hidden layer and one output layer. The network shown is fully connected, which means any neuron in any layer is connected to all the neurons in the previous layer. The input signals are mapped into the input neurons, passed through the hidden neurons via different weighted connections at both sides of the hidden layer, and finally emerged to an output signal from the output neuron. The structure of an individual neuron in the hidden layer and output layer is identical to the processing element in the single layer proceptron neural network. Because the multilayer proceptron is able to have more than one hidden layers with different number of hidden neurons, its structure is more complicated than that of single layer proceptron neural network.

Figure 3.2 depicts a portion of the multilayer proceptron and two different signals are identified in this network (Parker, 1987):

1. The function signals pass through the network from input end to output end, called forward pass. The signals will be adjusted by the activated function contained in the neurons and the associated weights connecting the neurons. Finally, they will emerge as an output signal.
2. The error signals are the differences between the targets and the network outputs originally, and they pass through the network from the output end to the input end, called reverse pass. The error signal involves an error-dependent function to modify the weights which connect the different neurons in two layers.

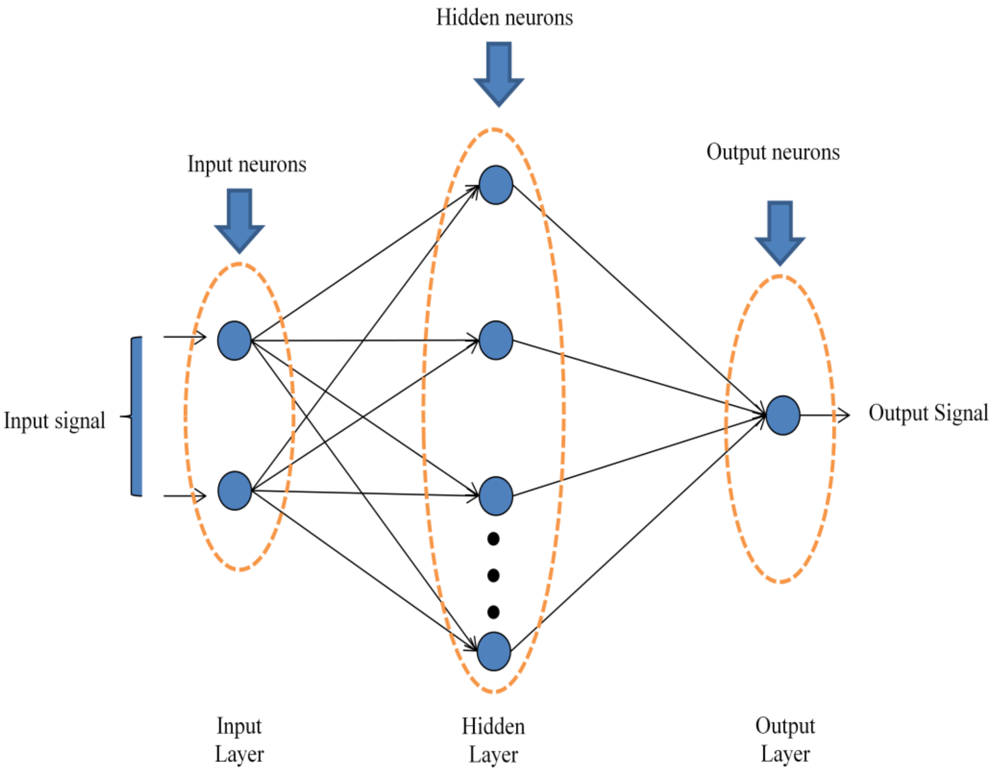


Figure 3.1 Architectural Graph of a Multilayer Proceptron with One Hidden Layer

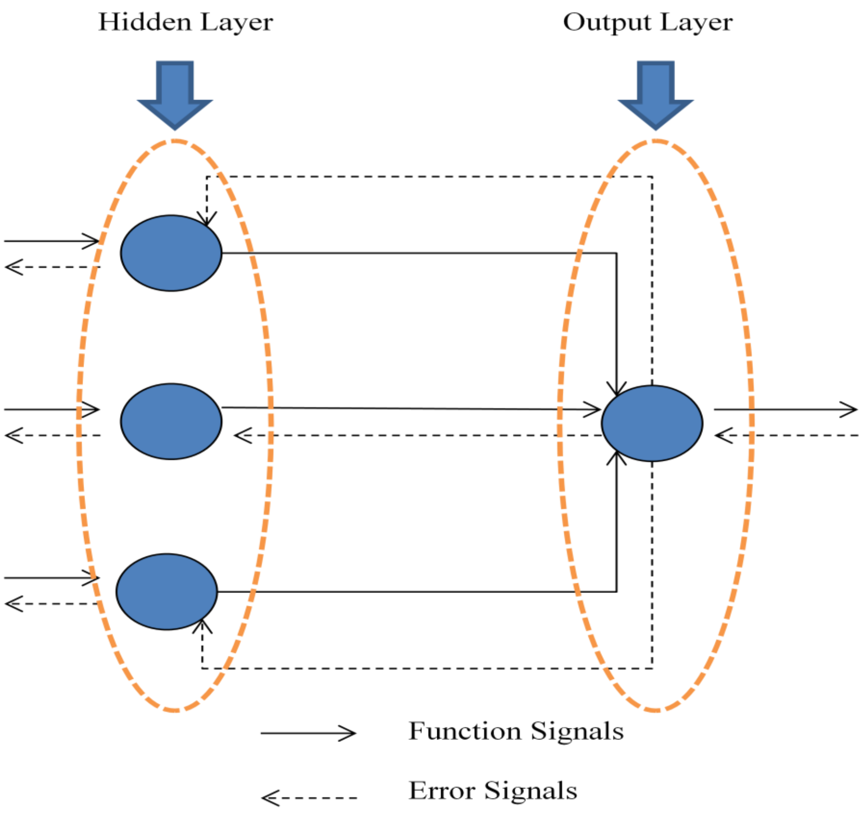


Figure 3.2 Illustration of Bio-directional Signal Flows

3.2 Back-propagation Algorithm

In this section, the details of BP algorithm will be explained. Figure 3.3 shows the architectural graph of a BPN with one hidden layer and one output layer. Compared with a standard multilayer proceptron, the fixed inputs (Bias ≡+1)was added in the architecture of BPN.

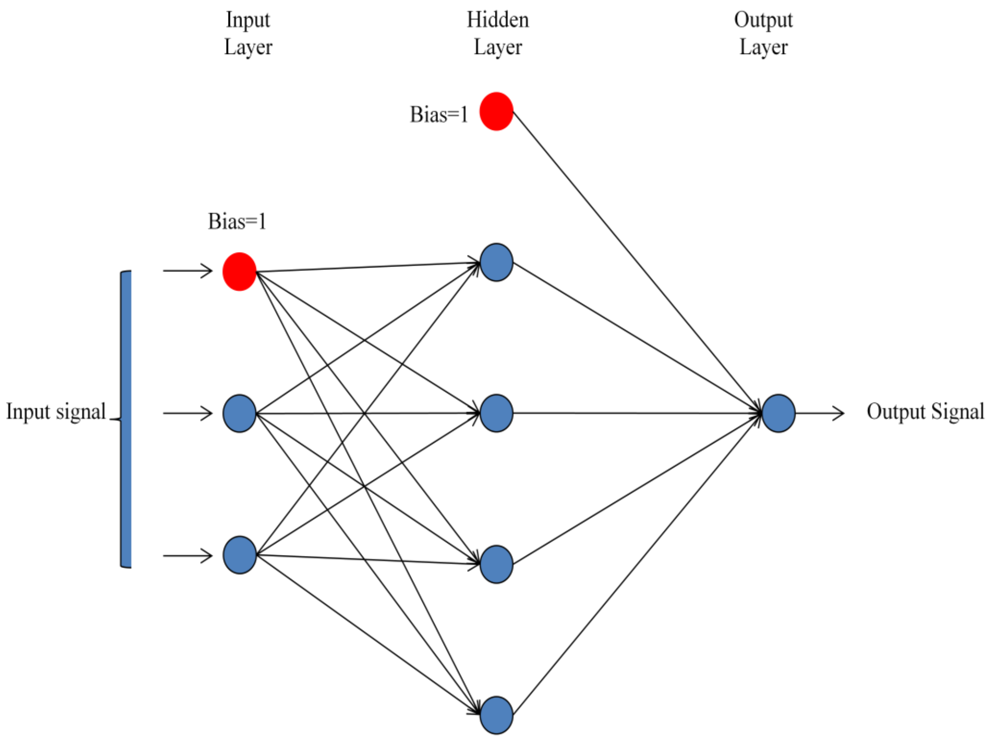


Figure 3.3 Architectural Graph of a Back-propagation Neural Network with One Hidden Layer and One Output Layer

3.2.1 Forward Pass

The forward pass is that the given input signals pass through the network and emerge to the output signals. As a one hidden layer and one output layer BPN, the hidden layer output signals will be calculated first and then act as the input signals of the output layer. The input signals of the hidden layer can be calculated by Equation 3.1.

$$I = W^T \cdot X \tag{3.1}$$

where I is the input signal of hidden layer, W is the weight matrix between input layer and hidden layer, X is the input sample, which contains fixed bias and the input features.

The output signal of hidden layer will be obtained by Equation 3.2.

$$H = T(I) = \begin{bmatrix} T(I_1) \\ T(I_2) \\ T(I_3) \\ \vdots \\ T(I_n) \end{bmatrix} \tag{3.2}$$

where *n* is the number of neurons in the hidden layer, *T(I)* is the activation function in the neurons. The graph of activation function is “s-shaped”, also called sigmoid function, which is defined as an odd, asymptotically bounded, completely monotone function of one variable. Mennon et al. (1996) presented a detailed study of two classes of sigmoids, simple sigmoids and hyperbolic sigmoids. In this research, a simple sigmoid function, tansig function, is applied as the activation function in the neurons, and its graph is shown in Figure 3.4 and the mathematic form is represented in Equation 3.3.

$$T(I) = \frac{2}{1+e^{-(2\cdot\alpha\cdot I)}} - 1 \tag{3.3}$$

where α is the momentum,α>0.

The output signals of hidden layer, *H*, associated with the weights, *V*, between hidden layer and output layer (Equation 3.4) will act as the input signals, *J*, of output layer.

$$J = V^T \cdot H \tag{3.4}$$

The output signal of output layer,$\vec{Y}$, can be calculated by Equation 3.5.

$$\vec{Y} = T(J) \tag{3.5}$$

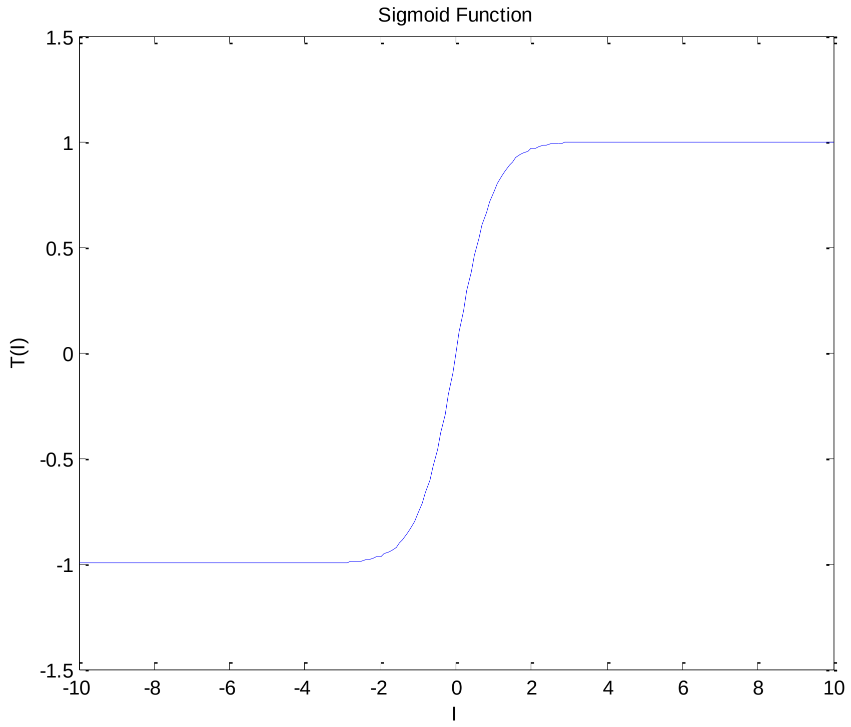


Figure 3.4 Graph of a Tansig Function, α=1

3.2.2 Reverse Pass

The reverse pass refers to the back-propagation of the error signal. Equation 3.6 defines the error signal.

$$e = Y_d - \vec{Y} \tag{3.6}$$

In the reverse pass, the goal is to adjust all weights in the network to reduce the error of the training process iteratively. The definition of the error energy of the output neuron is:

$$\varepsilon = \frac{1}{2} \cdot e^2 \tag{3.7}$$

The back-propagation algorithm applies a partial derivative $\frac{\partial \varepsilon}{\partial V}$ to correct the weight matrix, *V*. According to the chain rule, this gradient can be expressed as:

$$\frac{\partial \varepsilon}{\partial V} = \frac{\partial \varepsilon}{\partial e} \cdot \frac{\partial e}{\partial \vec{Y}} \cdot \frac{\partial H}{\partial J} \cdot \frac{\partial J}{\partial V} \tag{3.8}$$

After calculating the single terms in right side of Equation 3.8, Equation 3.8 yields:

$$\frac{\partial \varepsilon}{\partial V} = -e \cdot T^{'}(J) \cdot H \tag{3.9}$$

By using the delta rule, the adjustment of weight matrix *V* will be:

$$\Delta V = -\beta \cdot \frac{\partial \varepsilon}{\partial V} = \beta \cdot \delta_0 \cdot H \tag{3.10}$$

Where β is the learning rate and δ_0 is the local gradient of output layer defined by:

$$\delta_0 = e \cdot T^{'}(J) \tag{3.11}$$

then

$$V_{new} = V_{old} + \Delta V \tag{3.12}$$

To update the weights between the input layer and hidden layer, it is required to calculate the equivalent local gradient. Because the error signals fed back to the hidden layer associated with the weights between hidden layer and output layer. The local gradient for updating the weight matrix *W* is defined as:

$$\delta_h = \delta_o \cdot V \cdot T^{'}(I) \tag{3.13}$$

By using the delta rule, the adjustment of weight matrix *W* will be:

$$\Delta W = \beta \cdot \delta_h \cdot X \tag{3.14}$$

then

$$W_{new} = W_{old} + \Delta W \tag{3.15}$$

The weight matrices *W* and *V* will be adjusted iteratively until the stopping criteria were met.

3.2.3 Stopping Criteria

Generally, back-propagation algorithm was not guaranteed to be converged after the iterative training. Some previous studies formulate sensible convergence criterions as follows:

1. The back-propagation algorithm is considered to have converged when the Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold. (Kramer and Sangiovanni-Vincentelli, 1989)
2. The back-propagation algorithm is considered to have converged when the absolute rate of change in the average squared error per epoch is sufficiently small. (Haykin, 1998)
3. The back-propagation algorithm is considered to have converged when the maximum training epoch is reached.
4. The back-propagation algorithm is considered to have converged when the maximum training time is reached.

5. The back-propagation algorithm is considered to have converged when the rate of change in the average squared error per epoch is increasing; in other words, the validation check fails.

In this research, the author applied all of the stopping criteria to detect whether the training of back-propagation neural network is converged for keeping the network from over-training.

CHAPTER 4
RADIAL BASIS FUNCTION GENERALIZED REGRESSION NEURAL NETWORK (RBFGRNN)

The RBFGRNN is a modification of the traditional Generalized Regression Neural Network (GRNN) that was developed by Specht (1991) (an adaptation of the Nadaraya-Watson kernel regression approximator (Nadaraya, 1965)), and the figure below shows its three-layer structure.

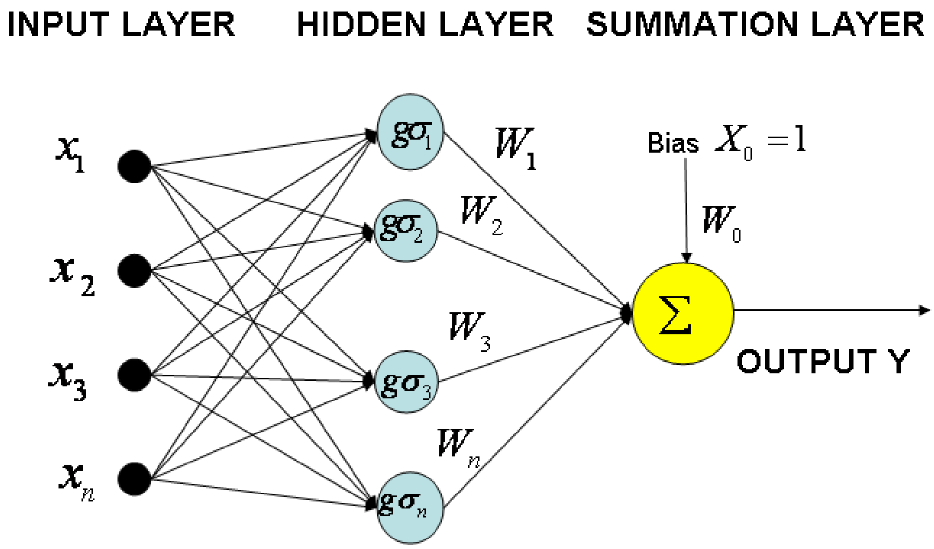


Figure 4.1 RBFGRNN Structure

This network is akin to the Radial Basis Functional (RBF) network in which there is a hidden unit centered at each cluster center. These RBF units in the hidden layer are called Gaussian displacement units (GDUs) and correspond to kernels functions in the Nadaraya-Watson kernels regression approximator. The GDUs require the sample covariance matrix from the training data as well as the input cluster centers.

The computation of the GDUs is governed by the Gaussian distribution function as following:

$$g_{\sigma}(x_i, t_k) = e^{-\frac{1}{2{\sigma_k}^2}\left[(x_i - t_k)^T C^{-1} (x_i - t_k)\right]} \tag{4.1}$$

where x_i is the i_{th} input vector, t is localized centers representing clusters of the input vectors, C is the covariance matrix of the input samples in cluster k (Haykin, 1998), and σ_k is the spread parameter of k_{th} cluster, estimated by the Equation 4.2:

$$\sigma_k = \frac{1}{2n} \sum_{\alpha=1}^{n} \max_{i,j=1...p} \left[\left| (x_i - x_j) \right|_{\alpha} \right] \tag{4.2}$$

where x_i and x_j is any pair of the p samples in cluster k, and n is the dimension of a sample. Figure 4.2 depicts 3D graph of the Gaussian distribution function with spread = 0.2 and center = (0, 0). The center is located at [0 0], represented by the red peak point displayed in Figure 4.2.

The spread or called standard deviation σ is defined as the width of the cluster whose center is located at (0, 0) and it shows how much variation exists from the mean. A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data points are spread out over a large range of values.

Ideally, the centers and spreads in Equations 4.1 and 4.2 can be acquired by unsupervised clustering methods, such as K-means, C-means, Divisive Analysis (DIANA), Kohonen self organized mapping (KSOM), and so on. However, these approaches need to fix the number of centers and may require a large training set for a

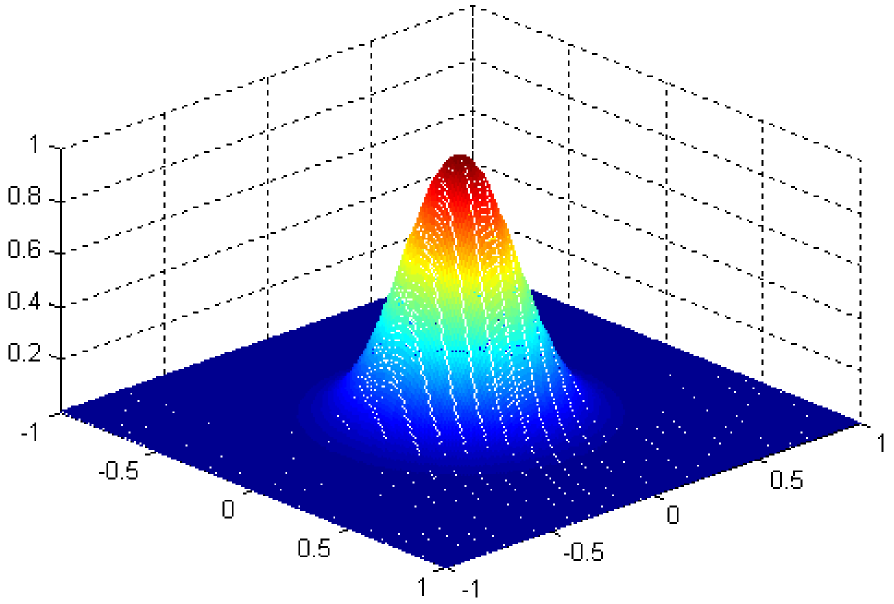


Figure 4.2 3D Graph of the Gaussian Distribution Function with Spread = 0.2 and Center = [0, 0]

satisfactory level of performance. If the training set is not large enough, it limits that the RBFGRNN and can only achieve a local optimum solution that depends on the initial choice of cluster centers. For this reason, a supervised selection of centers will be applied in this research. The centers and spreads of the radial-basis functions undergo a supervised learning procedure and it will be discussed at the end of this chapter.

The output from the Gaussian displacement layer is then fed into a linear regression network in order to map the GDU outputs to target training data. Allowing x to be a set of input vectors and y to be the corresponding target output, a relationship can be established such that a set of weights, w, can be found that represent the mathematical connection between the input and output.

$$G = \begin{bmatrix} 1 & g_\sigma(X_1,T_1) & g_\sigma(X_1,T_2) & \cdots & g_\sigma(X_1,T_m) \\ 1 & g_\sigma(X_2,T_1) & g_\sigma(X_2,T_2) & \cdots & g_\sigma(X_2,T_m) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & g_\sigma(X_p,T_1) & g_\sigma(X_p,T_2) & \cdots & g_\sigma(X_p,T_m) \end{bmatrix} \tag{4.3}$$

where $g_\sigma(X_p,T_m)$ is the GDU with p_{th} sample and m_{th} center, and the original *p-by-n* input space is expanded to a *p-by-m* space with the bias in the first column, $n \ll m$. The linear relationship is established as following equation.

$$G \cdot \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_m \end{pmatrix} = \begin{pmatrix} y_{d0} \\ y_{d1} \\ \vdots \\ y_{dm} \end{pmatrix} \tag{4.4}$$

where $w_0 \cdots w_m$ are the linear associate weights, and $y_{d0} \cdots y_{dm}$ is the desired output. The weight vector is obtained by taking the inverse of Gaussian matrix and multiplied by the desired output Y_d. If the inverse of Gaussian matrix does not exist, pseudo-inverse of the Gaussian matrix is an alternative.

$$W = (G)^{-1} \cdot Y_d \tag{4.5}$$

$$W = (G^T G)^{-1} G^T Y_d \tag{4.6}$$

The predicted results will be obtained by

$$Y_{pred} = G(x_{text}, T) \cdot W \tag{4.7}$$

where $G(x_{text}, T)$ is the GDU with testing samples and trained centers, *T*.

The supervised center selection method is a mechanism which selects the centers and equivalent spread based on RBFGRNN testing performances through trial-and-error processes. However, it is different from conventional error-feedback algorithms, because the test performance evaluated by the mean square error of the test samples of each attempt is only mapping of a number of centers and a unified spread, which means the clusters created have different centers but the same width. By varying the number of centers and value of spread, the near globe optimal, decided by the step size between pervious value of spread and current value of spread, but an acceptable solution will be found.

The first step in the development of such a supervised center selection method is to select a training sample as the initial center with a large spread. The larger the spread is, the smoother the function approximation. Too large a spread means a lot of neurons are required to fit a fast-changing function. Too small a spread means many neurons are required to fit a smooth function, and the network might not generalize well.

After the selection of initial center and spread, perform the RBFGRNN training and testing by using equations 4.1, 4.3-4.7, and calculate and record the mean square error between network outputs and desired outputs of the testing section. Then build up a linear regression model of network outputs and target outputs in the training section as shown in figure 4.3.The point (Y_j, T_j) has the maximum distance from $Y = T$, so the j_{th} training sample will be selected as another center.

Keep finding the training samples whose pair of (Y, T) has the maximum distance form $Y = T$ until all the training samples are selected as the centers. Then a profile of mean square errors of each attempt has been recorded.

Reduce the value of the spread by a small step size, and repeat the steps above, and then another profile of networks' performances will be obtained. Implement the iterative operations above until the spread is reduced to zero, and find the number of centers and value of spread which are mapped to the minimum mean square error in the testing section. It will be the parameters which lead to the optimal modeling solution.

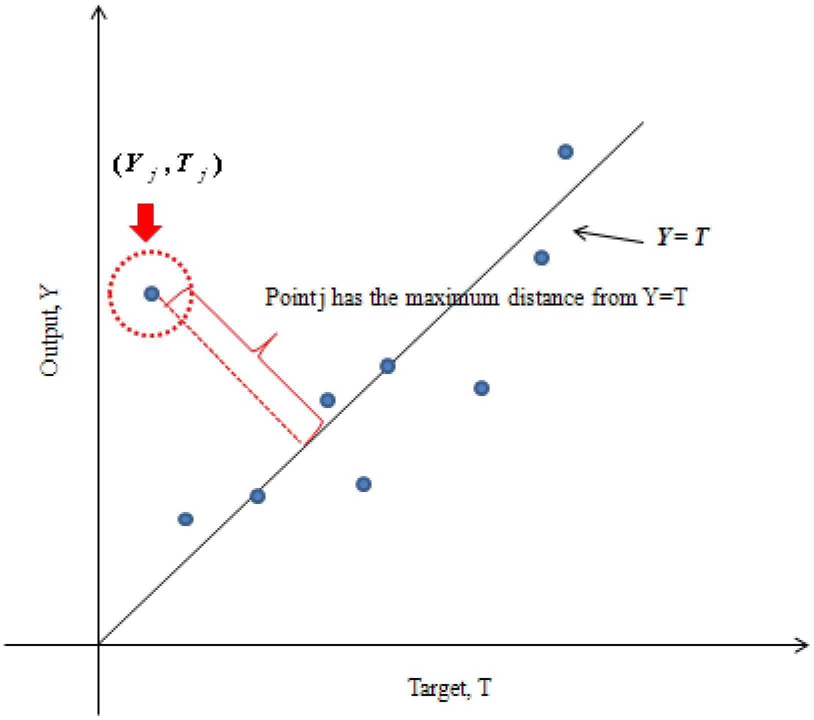


Figure 4.3 A Linear Regression Model of Network Outputs and Target Outputs in RBFGRNN Modeling

CHAPTER 5
CASE STUDIES

5.1 Model Performance Validation Methods

In this section, three model validation methods will be introduced to check the modeling performance, which are mean square error *MSE*, coefficient of correlation *R*, and coefficient of determination *R*^{2}.

The mean square error measures the average of the squares of errors. The error is the difference between which the value implied by the estimator and the quantity to be estimated. The mathematic form of mean square error is described by Equation 5.1.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_{pred,i} - Y_{desied,i})^2 \tag{5.1}$$

Where $Y_{pred,i}$ is the i_{th} predicted output, $Y_{desied,i}$ is the i_{th} desired output, and $i = 1 \cdots n$, n is the length of the output vector.

The value of the coefficient of correlation *R* is such that -1 ≤ *R* ≤ +1. An *R* value of exactly +1 indicates a perfect positive fit, and an *R* value of exactly -1 indicates a perfect negative fit. If there is no linear correlation or a weak linear correlation, *R* is close to 0. The mathematic form of coefficient of correlation is described by Equation 5.2.

$$R(i,j) = \frac{Cov(i,j)}{\sqrt{Cov(i,i) \cdot Cov(j,j)}} \tag{5.2}$$

Where $R(i,j)$ is the correlation coefficient of vector *i* and vector *j*, and $Cov(i,j)$ is the covariance matrix of vector *i* and vector *j*. Vector *i* represents the predicted outputs of sub-network testing and vector *j* represents the desired outputs of sub-network testing.

To evaluate the performance of the designed model, the coefficient of determination, R^2 test, is introduced. The R^2 test is a statistical indicator that compares the accuracy of the proposed model and is described in Equation 5.3.

$$R^2 = 1 - \frac{\Sigma_{i=1}^{n}(Y_{pred,i} - Y_{desied,i})^2}{\Sigma_{i=1}^{n}(Y_{pred,i} - Y_{mean})^2} \tag{5.3}$$

Where $Y_{pred,i}$ is the i_{th} predicted output, $Y_{desied,i}$ is the i_{th} desired output, and $i = 1 \cdots n$, n is the length of the output vector. The R^2 test gives the proportion of the variance of one variable that is predictable from the other variable. A perfect fit would result in R^2=1, while R^2=0 indicating a very poor fit.

5.2 Case 1: Leachate Flow Rate Prediction in Greensboro, North Carolina

5.2.1 Introduction

Landfill is the oldest and most common method of solid waste disposal by burying the collected municipal solid waste (MSW). Early landfills were put in convenient and on the least expensive land. As rain washes through the waste tip, it dissolves some of the solids and mixes the liquids. The water can become acidic and eat into the waste in containers and produces a contaminated fluid called leachate. Leachate escapes from most old landfills, contaminates the surface and underground water systems, and threatens the drinking water supply and other water uses. Modern landfills are designed to protect the environment from pollution. More recently, landfills have had barriers designed to keep the leachate within the landfill systems. Engineers line the landfill with clay or synthetic materials to prevent the pass through of leachate. Pipes are then used to collect the leachate for storage in tanks and for special treatment. However, the USEPA has stated that the barriers "will ultimately fail," while the site remains a threat for "thousands of years," suggesting that modern landfill designs delay but do not prevent ground and surface water pollution. Based on these facts, it is important and significant to estimate the leachate flow rate at the bottom of a MSW landfill to prevent the mixing of leachate with the streams which flow towards the major ground water systems. Many previous studies indicate that artificial neural network methods are the effective approaches to modeling different types of nonlinear systems. Burke, et al. (1994) proved the back-propagation neural network can perform as well as the best traditional methods for the breast cancer outcome prediction, and that they can capture the power of non-monotonic predictors and discover complex genetic interactions. Khoa, et al. (2006) introduced a neural network based method to forecast the stock price, and demonstrated the ability of back propagation neural network to model a nonlinear process without a prior knowledge about the nature of the processing. A back propagation neural network was proposed for modeling the leachate flow-rate in a municipal solid waste (MSW) landfill site (Ferhat and Bestamin ,2006**).** In this thesis, the radial basis functional generalized regression neural network based leachate flow rate estimator has been developed, and a case study was performed to validate the proposed model.

5.2.2 Data Selection

Most of the neural network model for leachate flow rate prediction in previous studies cannot capture all of the features which affect the leachate flow rate, both the peak and average. Ferhat and Bestamin (2006) selected 11 input features of Back Propagation Neural Network (BPN), including pH value (collected leachate), temperature (collected leachate), conductivity (collected leachate), months, temperature (meteorological parameter), pressure, cloudiness, relative humidity, precipitation, maximum temperature and minimum temperature. Chang and Wang (2009) selected porosity, field capacity, wilting points, saturated hydraulic conductivity and the layer thickness among 23 available parameters of Hydrologic Evaluation of Landfill Performance (HELP) model as the input features of their back propagation neural network. The sensitivities of these five parameters affecting the leachate flow rate were analyzed individually. In this study, leachate flow rate prediction modeling is also based on these five parameters, but this thesis will focus on the synthesis effect on the leachate flow rate caused by these five parameters. The data generation process is based on the well-known computer program that computes estimates of water balances for municipal landfill, HELP. The input features are generated randomly in a qualified range and are able to be implemented by the HELP model.

Once the samples are generated, a normalization method is applied to scale the values of input and output from 0 to 1 by using Equation 5.4.

$$Data_{Normalized} = \frac{Data - Min(Data)}{Max(Data) - Min(data)} \tag{5.4}$$

After the normalization, 75% data sets will be randomly selected as the training samples and the rest 25% data set will be the testing samples.

Once the predicted output is obtained, it will also be de-normalized by Equation 5.5.

$$Data = Data_{Norm} \times \left[Max(Data) - Min(Data) \right] + Min(Data) \tag{5.5}$$

For illustrating and validating the application of the RBFGRNN model, a case study is performed under the simulated environment in Greensboro, North Carolina. This simulated environment is based on the parameters of general climate data, daily climatologic data, soil characteristics, and design specifications from the HELP model, and the annual leachate flow rate was carried out by iterative calculation by the HELP model, which is the desired output for the network training and network testing.

The HELP model has a default evapotranspiration database for 183 U.S. cities, containing data for latitude, evaporative zone depths, leaf area indices, growing season, average wind speed, and average quarterly relative humidity. A default precipitation database is included, containing 5 years of daily values for 102 cities throughout the United States. This model also has a synthetic weather generator with coefficients for 139 cities for daily precipitation data generation and for 183 cities for daily temperature and solar radiation data generation. The model contains a default soil database of characteristics for 42 types of materials (soils, waste, and geosynthetics). In this case study, the essential landfill design parameter and the climate data set are listed in Table 5.1. The monthly mean temperature and monthly precipitation are shown in Figure 5.1 and Figure 5.2. A snapshot of the data set containing input and output samples is listed in Table 5.2.

5.2.3 BPN Model Description and Results in Case 1

As mentioned in chapter 3, there are a number of key parameters in the back propagation neural network with one hidden layer and one output layer. First of all is the size of hidden layer that implicates the number of hidden neurons in the hidden layer.

However, there is no efficient approach to determine the optimal number of hidden neurons. Hence, the back-propagation model will vary the number of hidden neurons from 10 to 50, called initial screen.

Table 5.1 Landfill Design Parameters for HELP Model at Greensboro, NC

Type of Data	Parameters	Value
General Climate Data	Start of Growing Season	90 days
	End of Growing Season	305 days
	Average Wind Speed	7.6 MPH
	First Quarterly Relative Humidity	66.00%
	Second Quarterly Relative Humidity	68.00%
	Third Quarterly Relative Humidity	74.00%
	Fourth Quarterly Relative Humidity	70.00%
Daily Weather Data	Evaporative Zone Depth	35 in
	Maximum Leaf Area Index	3.5
	Latitude	35.13
	Average Temperature	57.875 $^{\circ}F$
	Precipitation and Mean Temperature	See Figures 5.1 and 5.2
	Porosity	0.671
Soil Characteristics	Field Capacity	0.292 m^3
	Wilting Point	0.077 kg/m^3
	Sat. Hydr. Conductivity	0.01 cm/day
	Initial Moisture Storage	0.300 m^3
	Runoff Curve Number	82.2
Design Specifications	Landfill Area	15 acres
	Municipal Waste Specific Weight	900 lb/yd^3
	Slope	3.00%
	Soil Texture	9
	% of Area Where Runoff is Possible	100%

The sub-network with minimum mean square error and coefficient of determination will be chosen as the candidate for further experiment. The stopping criteria in the initial screen are:

1. Maximum number of epochs to train is 5000.
2. Performance goal (mean square error of the training result) is 0.
3. Minimum performance gradient is 1×10^{-10}.
4. Maximum validation failures equal to 1.

Table 5.2 Generated Data Set Snap Shot by Using HELP Model

Layer Thickness (in)	Porosity	Field Capacity	Wilting Point	Sat. Hydr. Conductivity(cm/s)	HELP (in/yr)
150	0.671	0.292	0.077	0.001	7.328
300	0.671	0.292	0.077	0.001	7.298
182	0.671	0.292	0.077	0.001	7.34
300	0.736	0.292	0.077	0.001	7.004
300	0.363	0.292	0.077	0.001	8.961
300	0.671	0.419	0.077	0.001	8.418
300	0.671	0.587	0.077	0.001	11.259
300	0.671	0.448	0.077	0.001	8.549
300	0.671	0.292	0.017	0.001	9.064
300	0.671	0.292	0.026	0.001	8.507
300	0.671	0.292	0.077	0.004	8.614
300	0.671	0.292	0.077	0.007	9.266

At each training epoch, the testing samples will be applied to validate the network training performance. If the mean square error is increased, the validation fails. It prevents the network over training. Once the candidate is chosen after the initial screen, this network will be initialized and pass through the training process again without the

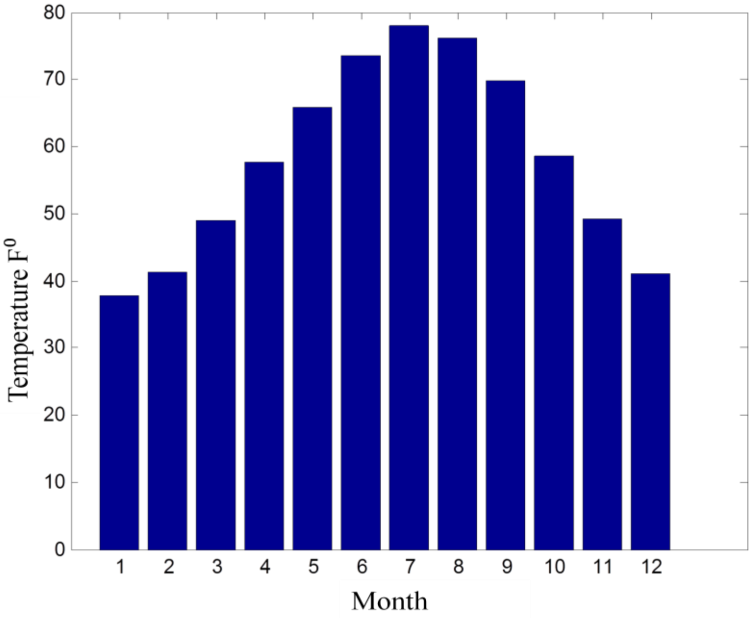


Figure 5.1 Monthly Mean Temperatures in City of Greensboro, NC

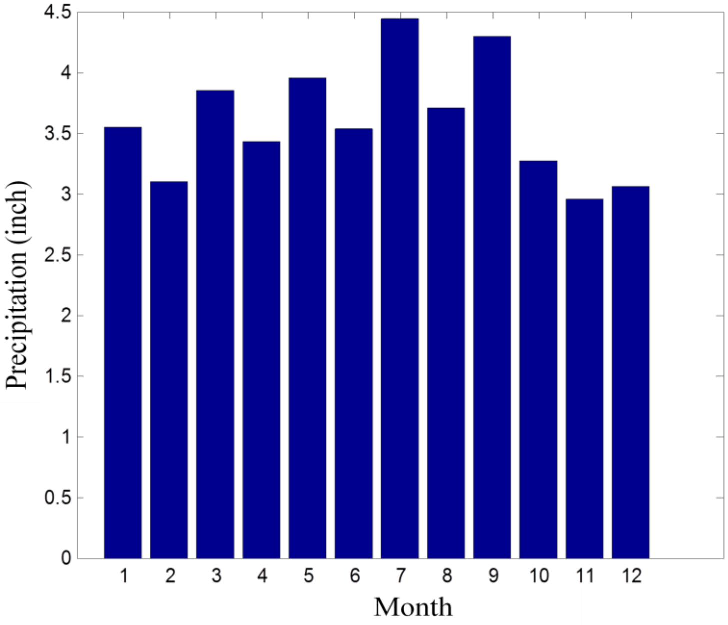


Figure 5.2 Monthly Mean Precipitation in City of Greensboro, NC

limitation of criterion 1 listed above. Secondly, the initial weights in matrices *W* and *V* are generated within the interval, (0,1), by a uniform random generator. The tansig function is applied as the activation function in the processing elements with α=1. The learning rate *β* is fixed as 0.25.

Figure 5.3 displayed the test performances of the initial screen of 41sub-BPNs with 10 to 50 hidden neurons in the hidden layer, evaluated by mean square errors between the predicted outputs and desired outputs of different sub-network testing. The

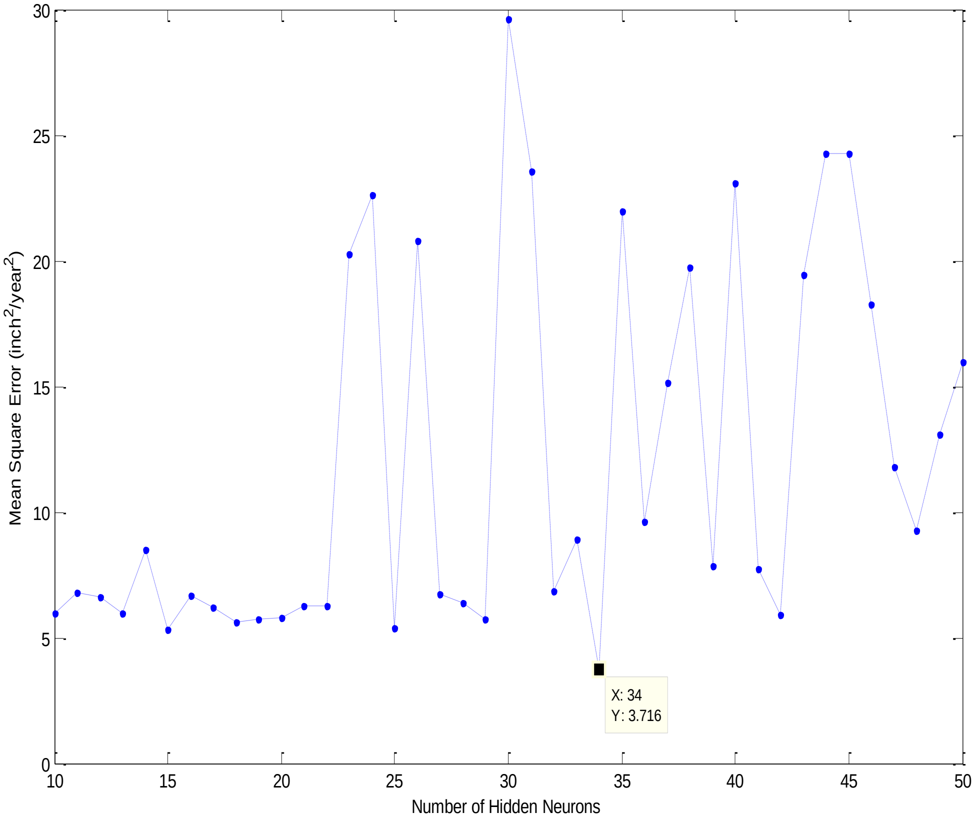


Figure 5.3 Performances of the Initial Screen of 41Sub-BPNs with 10 to 50 Hidden Neurons in the BPN Training Procedure of Leachate Flow Rate Modeling

x-coordinate represented the sub-networks with different hidden neurons from 10 to 50, and the y-coordinate represented their related mean square errors, evaluated by Equation 5.1. The little text block indicated the BPN with 34 hidden neurons in the hidden layer has the minimum testing mean square error 3.716 inch^2/year^2.

Figure 5.4 shows their correlation coefficients (*R*s), evaluated by equation 5.2, which measure the strength and the direction of a linear relationship between the network outputs and the desired outputs. The x-coordinate represented the sub-networks with different hidden neurons from 10 to 50, and the y-coordinate represented their related

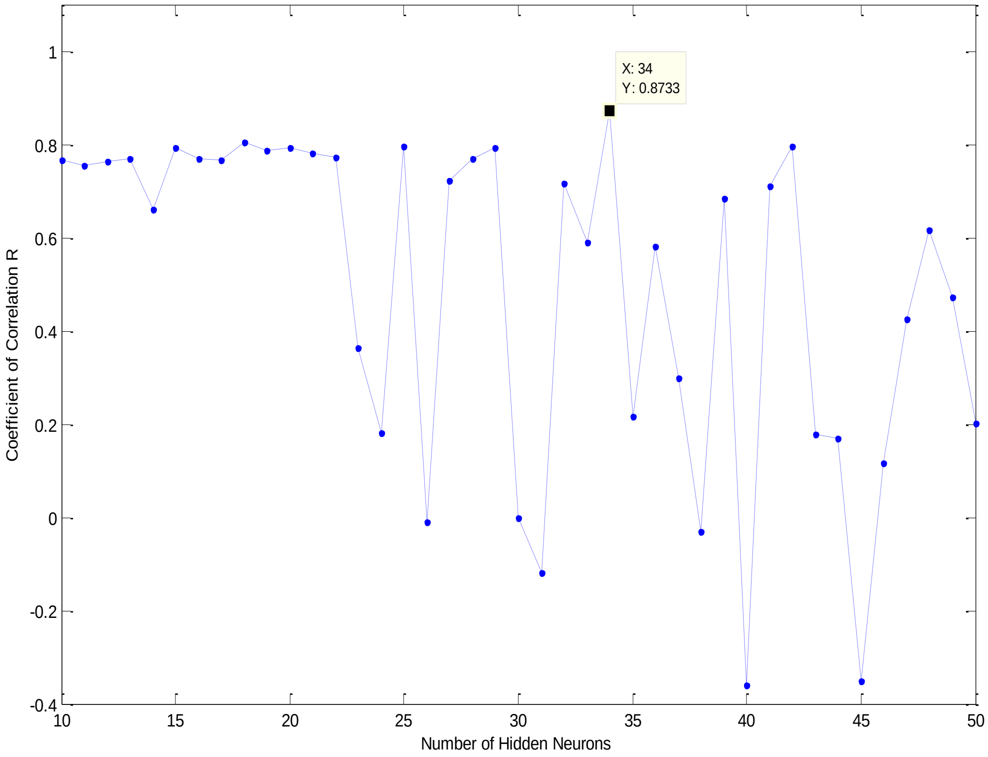


Figure 5.4 Correlation Coefficients of the Initial Screen of 41Sub-BPNs with 10 to 50 Hidden Neurons in the BPN Training Procedure of Leachate flow Rate Modeling

correlation coefficients. The little text block indicated the BPN with 34 hidden neurons has the maximum testing correlation coefficients 0.873.

Based on Figures 5.3 and 5.4, the best candidate will be the back-propagation neural network with 34 hidden neurons, which has the minimum mean square error 3.716 and the maximum linear correlation coefficient 0.873 among all 41 sub-BPNs. Figure 5.3 and 5.4 also implicated that increasing the number of hidden neurons will not improve the network performance directly. It is possible that there is a better solution when more than 50 hidden neurons are applied in the hidden layer, but it will enlarge the size of network, create more connections between each layer, increase the network training time and consume huge computation capacity. The candidate BPN with 34 hidden neurons will be initialized and retrained without the limitation of maximum training epochs. At 9704th training epoch, the validation check failed which means the mean square error of testing results kept decreasing until it reached the 9704th epoch. The local gradient δ at the output layer was decreasing to 0.001. As a result, the best normalized validation (testing) performance of the network (mean square error) is 0.009 at epoch 9703. These facts are demonstrated in Figures 5.5 and 5.6.

Figure 5.7 shows the normalized linear regression plots of leachate flow rate modeling by using BPN. The upper left plot indicates the linear regression model in the training section with R = 0.952, and $Y_{net\_train} \approx 0.84 \cdot Y_{train} + 0.066$. The upper right plot and lower left plot are the same, because the validation and test samples are identical. The *Rs* = 0.853, and the linear regression model can be represented as $Y_{net\_test} \approx 0.74 \cdot Y_{test} + 0.11$. The lower right plot is a summary of three previous cases, the overall

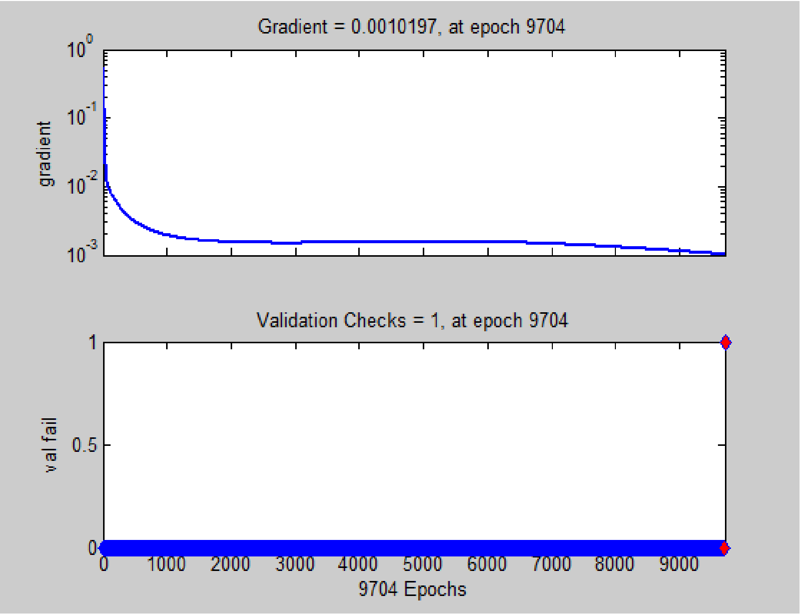


Figure 5.5 Training State Plots in the BPN Training Procedure of Leachate Flow Rate Modeling

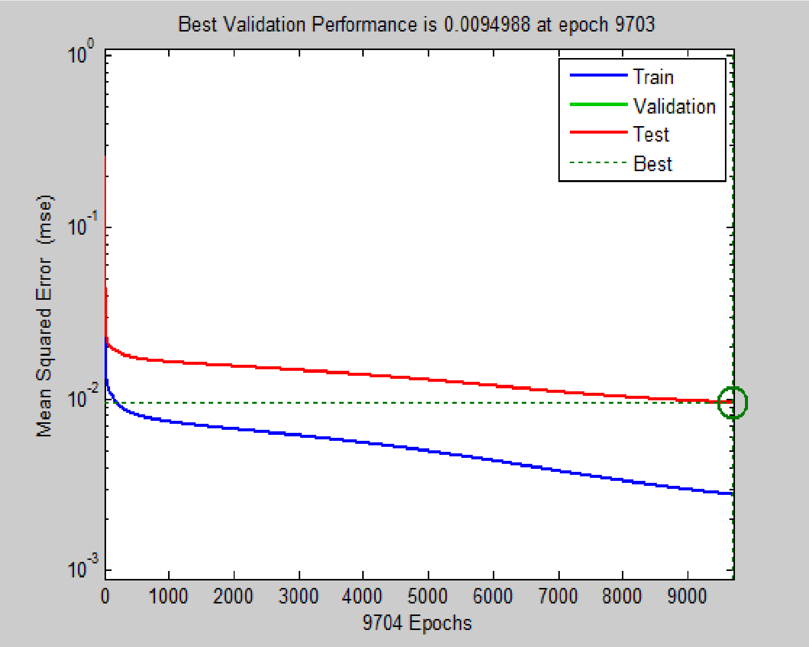


Figure 5.6 Performance Plot in the BPN Training Procedure of Leachate Flow Rate Modeling

correlation coefficient is 0.920, and $Y_{net\_all} \approx 0.79 \cdot Y_{all} + 0.086$. After de-normalization, the testing network outputs were re-scaled, and the testing regression model was changed to $Y_{net\_test} \approx 0.74 \cdot Y_{test} + 2.1$, as shown in figure 5.8, but the coefficient of correlation is same as the one before it was re-scaled.

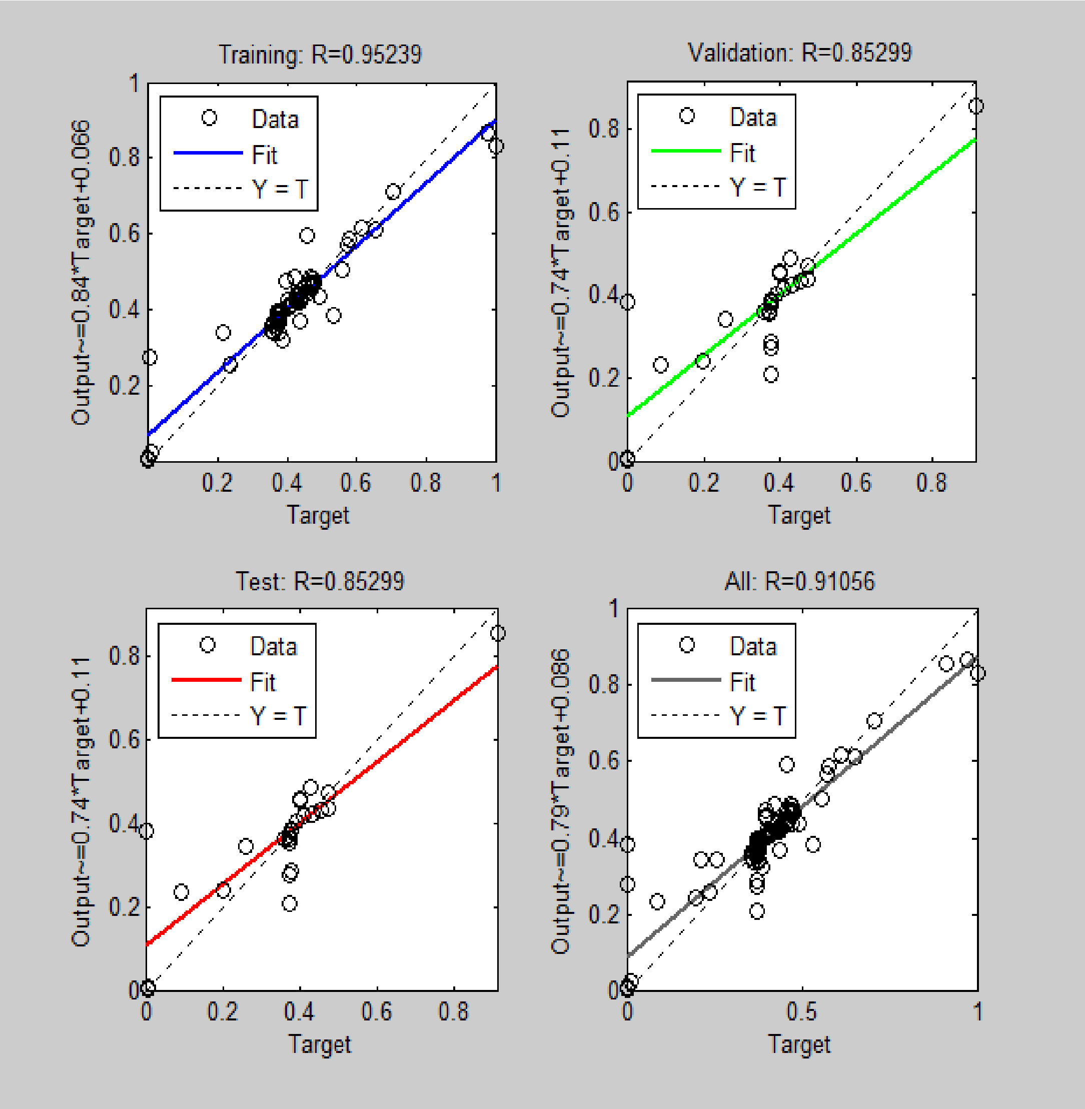


Figure 5.7 Regression Plots of Leachate Flow Rate Modeling by using BPN

Figure 5.9 depicts the test performance of the back propagation neural network in original scale. The dash line curve with circle marker represented the desired leachate flow rate and the dash line curve with square marker represented the BPN predicted leachate flow rate. The dash line curve with triangle marker represented the error calculated by desired leachate flow rate minus BPN predicted leachate flow rate. The x coordinator represented 25 testing samples, and the y coordinator represented the related leachate flow rate. The model successfully predicted the peak and valley values of the leachate flow rate within ±2 in/year. The largest error happened at 17th testing sample may due to the similar sample or samples in the training section less excited or no similar sample or samples were trained in the training section. As a result, the network did not learn such information contained in 17th testing sample.

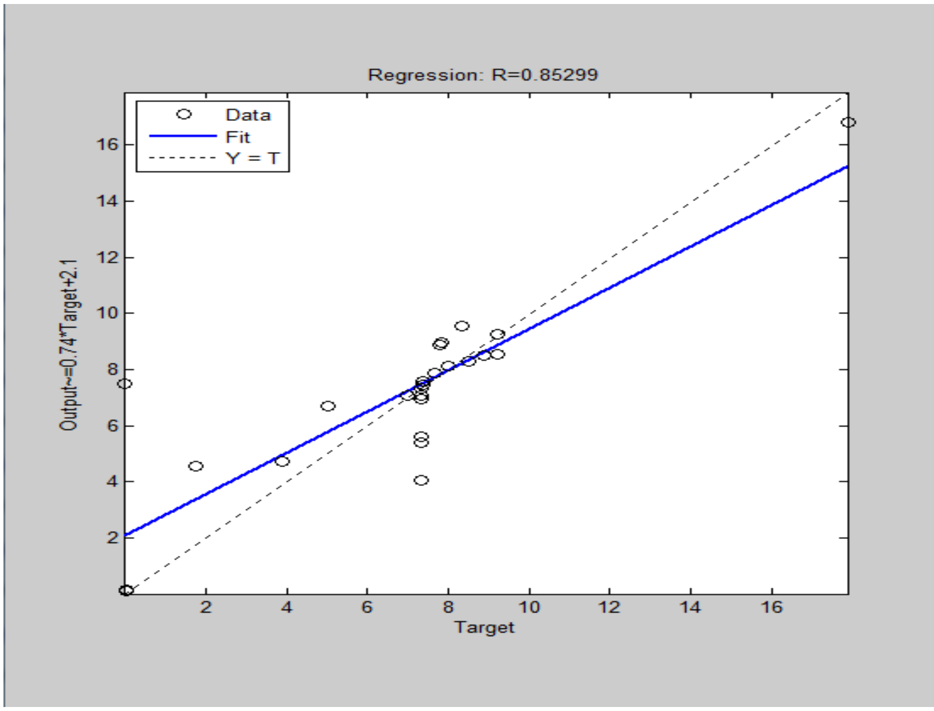


Figure 5.8 Testing Regression Plots of Leachate Flow Rate Modeling by using BPN

Finally the coefficient of determination R^2 is applied to evaluate the performance of the linear regression. Because R of the test section is 0.853, $R^2 = 0.728$, which means 72.8% of the total variation in the desired test output can be explained by the linear relationship between the desired test output and the BPN test output ($Y_{net\_test} \approx 0.74 \cdot Y_{test} + 2.1$), the other 27.2% of the total variation of the desired test output remains unexplained. The mean square error of the final BPN leachate flow rate modeling is 3.644inch^2/year^2.

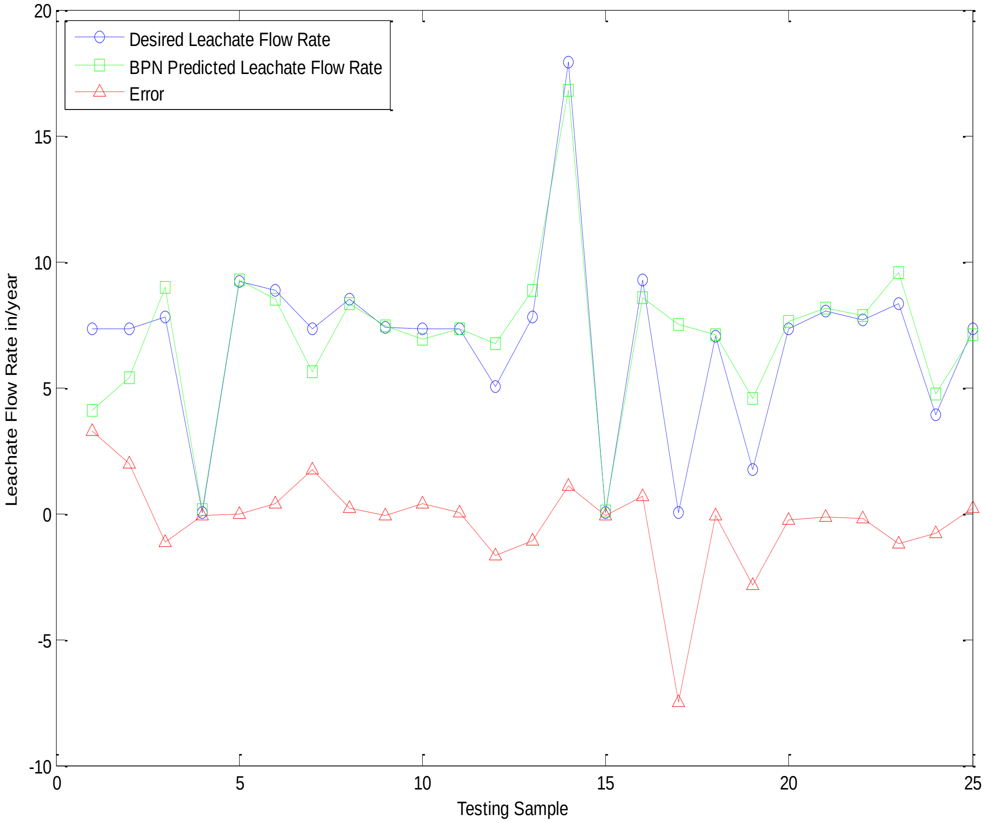


Figure 5.9 De-normalized BPN Test Performance, Desired Leachate Flow Rate vs. BPN Predicted Leachate Flow Rate

5.2.4 RBFGRNN Model Description and Results in Case 1

The structure of RBFGRNN is different from that of BPN, as well as the learning algorithm. A RBFGRNN with defined centers and spread is a one-pass network, which means there is no iterative weight updating or calculations. As mentioned in Chapter 4, the iterative process created for RBFGRNN is only aim to locate the optimal centers and spread. Definitely, testing performance decides the generalization ability of the proposed network and evaluates how well the network is learning the information given by the training samples. In this section, the training performance will be ignored and the testing performance will be amplified. Figure 5.10 shows the testing performances (mean square error) of a cluster of sub radial basis functional generalized regression neural networks. These subnets are varied by different number of centers and spread values. The mean square errors of different subnets are represented by different colors. Red color indicates high mean square error and blue indicates low mean square error. Based on recorded RBFGRNN testing performances, the RBFGRNN with 24 centers and *spread=1*has the best testing performance MSE = 2.430 inch^2/year^2.

Figure 5.11 shows the plot of linear regression model of the desired leachate flow rate and RBFGRNN predicted leachate flow rate in testing section. The circle represented point which is in the form of ($Y_{desired,i}, Y_{predicted,i}$), where $i \in [1, 25]$. The dash line represented $Y_{desired,i} = Y_{predicted,i}$, and the blue line represented the fitting curve, which is $Y_{predicted,i} \approx Y_{desired,i} + 1.3$with the correlation coefficient R = 0.907.

Figure 5.12 depicts the test performance of the radial basis functional generalized regression neural network. The dash line curve with circle marker represented the desired

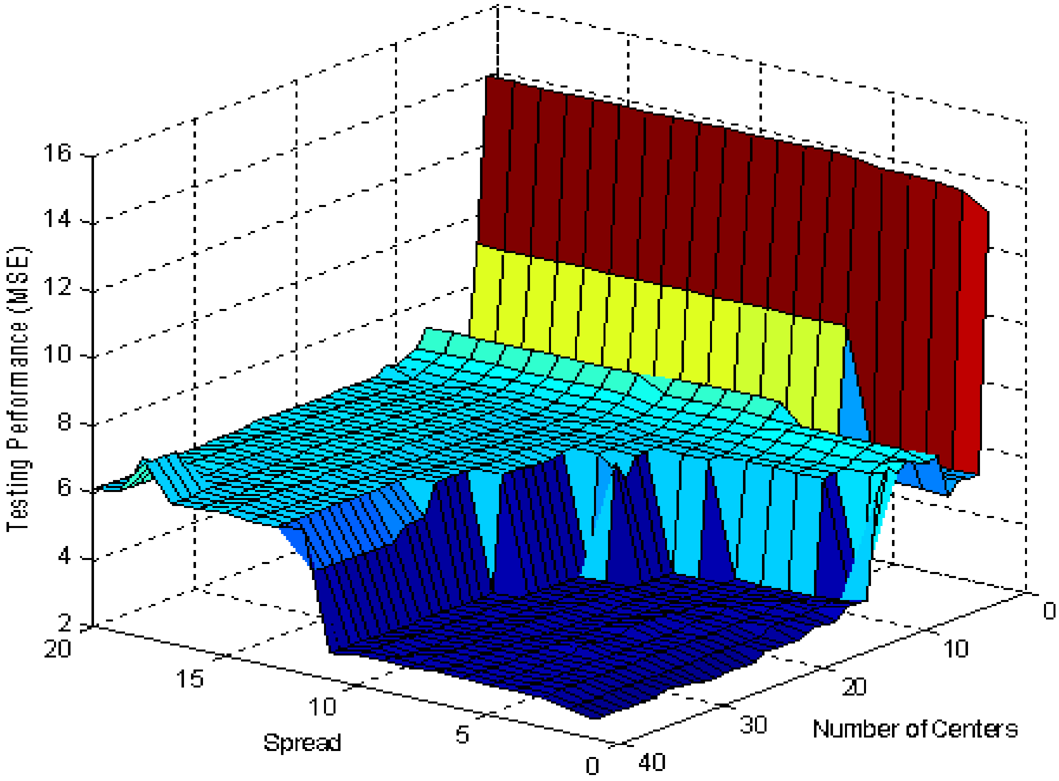


Figure 5.10 3D Plot of RBFGRNN Testing Performance with Different Spread and Centers in the Leachate Flow Rate Modeling

leachate flow rate and the dash line curve with square marker represented the RBFGRNN predicted leachate flow rate. The dash line curve with triangle marker represented the error calculated by desired leachate flow rate minus RBFGRNN predicted leachate flow rate. The x coordinator represented 25 testing samples, and the y coordinator represented the related leachate flow rate. The model successfully predicted the peak and valley values of the leachate flow rate within ±1 in/year. The largest error happened at 17th testing sample may due to the similar sample or samples in the training section less excited or no similar sample or samples were trained in the training section. As a result, the network did not learn such information contained in 17th testing sample. The mean square error between the desired leachate flow rate and the RBFGRNN predicted leachate flow rate is 2.430inch^2/year^2. The coefficient of determination R^2 is equal to 0.823, which means 82.3% of the total variation in the desired test output can be explained by the linear relationship between the desired test output and the RBFGRNN predicted output ($Y_{net\_test} \approx 0.85 \times Y_{test} + 1.3$).

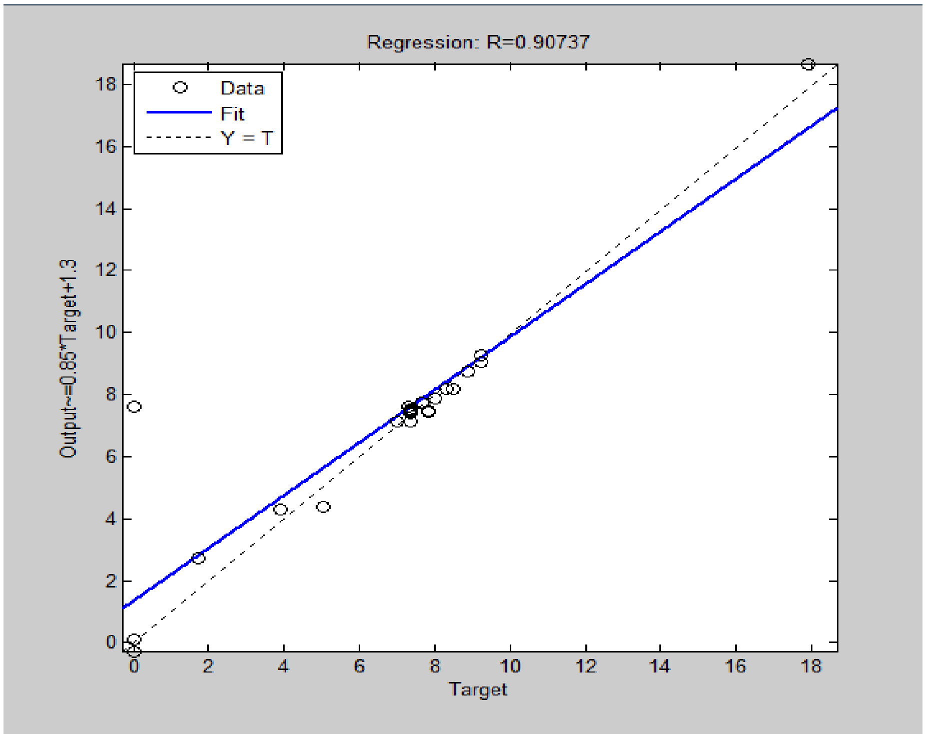


Figure 5.11 Regression Plots in the RBFGRNN Testing Procedure of Leachate Flow Rate Modeling

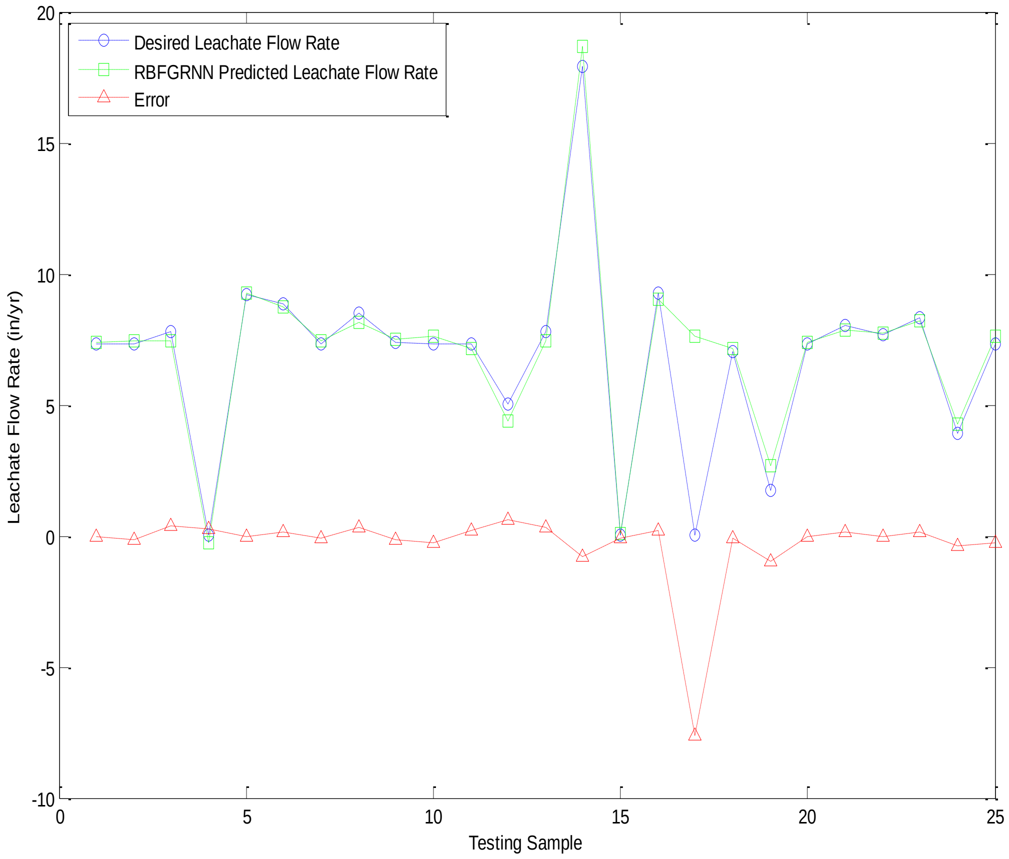


Figure 5.12 De-normalized test performance, Desired Leachate Flow Rate vs. RBFGRNN predicted Leachate Flow Rate

5.2.5 Case Study 1 Summary

In this landfill leachate flow rate modeling case, BPN and RBFGRNN are applied. Table 5.3 shows a performance summary of two networks. Compared with the BPN, RBFGRNN performed better evaluated by lower mean square error, higher coefficient of correlation, and higher coefficient of determination. The R^2=0.823 in the RBFGRNN modeling stated that 82.3% of the total variation in the desired test output can be explained by the linear relationship between the desired test output and the RBFGRNN predicted output.

Table 5.3 Testing Performances of Two Neural Network Applications in Case 1

Networks	MSE $inch^2/year^2$	Coefficient of Correlation	Coefficient of Determination
RBFGRNN	2.430	0.907	0.823
BPN	3.644	0.853	0.728

5.3 Case 2: Total Phosphorus Concentration Prediction in Te-Chi Reservoir, Taiwan

5.3.1 Background Information

Nutrients are important because they are required for growth of the microorganisms used in wastewater treatment processes and because, if not removed, they can lead to excess algal growth, particularly in lakes. The principal external sources of nutrient inputs are: municipal wastes; industrial wastes; agriculture runoff; forest runoff; urban and suburban runoff; and atmospheric fallout (Ray, 1994). Phosphorus, the primary controllable nutrient load, is one of the key elements necessary for growth of plants and animals and in lake ecosystems it tends to be the growth switch. The presence of phosphorus is often scarce in the well-oxygenated lake waters and importantly, the low levels of phosphorus limit the production of freshwater systems. Phosphates are not toxic to people or animals unless they are present in very high levels.

Phosphate supports and excites the growth of plankton and aquatic plants, which provide food for larger organisms, including: zooplankton, fish, humans, and other mammals. Plankton represents the lowest level of the food chain. Initially, this increased productivity will cause an increase in the fish population and overall biological diversity of the system. But as the phosphate loading continues and there is a build-up of phosphate in the lake or surface water ecosystem, the aging process of lake or surface water ecosystem will be accelerated. The overproduction of lake or water body can lead to an imbalance in the nutrient cycling process. Eutrophication is enhanced production of primary producers resulting in reduced stability of the ecosystem. Phosphate has been shown to be the main cause of eutrophication over the past 30 years. This aging process can result in large fluctuations in the lake water quality and trophic status and in some cases periodic blooms of cyanobacteria. Figure 5.13 displays the green algae booming in Dian Chi Lake, Yunnan, China, 2007. The picture is cited from the China Economic Net. According to the report from China News Net, the causation of the continuous green algae booming is the water contained the phosphorus from the life waste water, agricultural chemicals flows into the lake, and the high temperature.

Based on the negative side affection of massive green algae outbreak, it is significant to build an accurate total phosphate (TP) concentration prediction model. The main factors which appear to determine the development of plank-tonic populations are light, temperature, pH, nutrient concentrations and the presence of organic solutes.

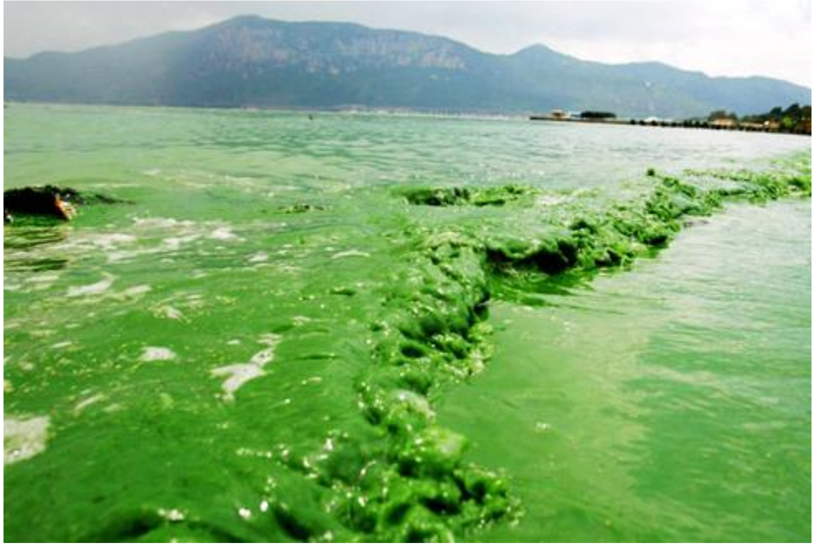


Figure 5.13 Green Algae Blooming in Dianchi Lake, Yunnan, China, 2007

In this case study, BPN and RBFGRNN will be applied in TP concentration prediction modeling, based on the historical water quality information of Te-Chi Reservoir and downstream of Ta-Chia Creek in central Taiwan.

5.3.2 Study Area Profile

The Te-Chi Reservoir is located in the downstream of Ta-Chia Creek in central Taiwan as shown in Figure 5.14, captured from Google map. It is the fourth largest (in terms of storage volume) reservoir in Taiwan with a maximum water surface area of 4.54 km^3 and initial design storage volume of about 232×106m^3. The annual inflow is about 1.2×109m^3, about five times the reservoir volume, but over three-fourth comes during the wet season. The watershed area is 592km^2. The watershed altitude varies from 3884 meter (highest mountain) to 1408m (normal water level)—a drop of over 2400 m. The slope of main branches in this field is mostly over 50% and the average slope usually exceeds 30%. The Environmental Protection Administration of Executive Yuan, R. O. C. has established five sampling stations in this reservoir area, as shown in Figure 5.15, captured from Google map.

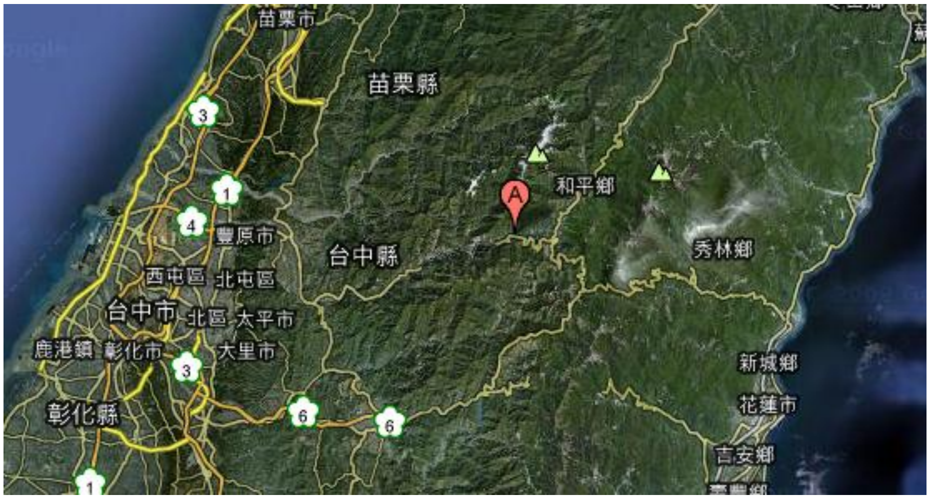


Figure 5.14 Location of the Te-Chi Reservoir

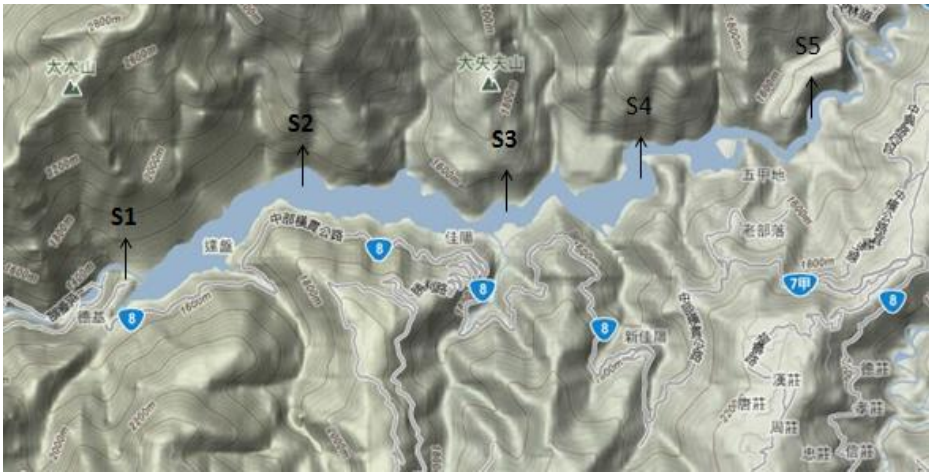


Figure 5.15 Sampling Stations in the Reservoir Area

5.3.3 Input Features Selection

Kuo, et al. (2007) performed a pre-screen of the potential input features through trial and error processes, and selected the PO_4 and Suspended Solid (*SS*) as the variables of their Total Phosphorus neural network model . In 2008,J. Możejko and R. Gniot selected 14 observations as their input features, which include Water Temperature, Air Temperature, *pH* Value, Total Kjeldahl Nitrogen, Nitrate-N ($N\text{-}NO_3$, $N\text{-}NO_2$), Total Phosphorus, Orthophosphate, Dissolved Oxygen, Biochemical Oxygen Demand, Chemical Oxygen Demand, Sulphate Concentration, Chloride Concentration, and Total Suspended Concentration (Możejko and Gniot, 2008). In this case study, a combination of these features mentioned above will be used as the input variables based on the available recorded historical observations and they are listed in the Table 5.4 as well as the output.

The samples used in this case study are recorded from 5 stations from December 1993 to August 2010, and downloaded from the website of Environmental Protection Administration of Executive Yuan, R. O. C. There are 52 qualified samples will be used in the artificial neural network TP concentration modeling.

5.3.4 BPN Model Description and Results in Case 2

As mentioned in Chapter 3, there are a number of key parameters in the back propagation neural network with one hidden layer and one output layer. First of all is the size of hidden layer that implicates the number of hidden neurons in the hidden layer. However, there is no efficient approach to determine the optimal number of hidden neurons. Hence, the back-propagation model will vary the number of hidden neurons from 10 to 50, called initial screen. The network with minimum mean square error and coefficient of determination will be chosen as the candidate for further experiment.

Table 5.4 Neural Network Input and Output Variables of Total Phosphorus Case

Division	Variable	Units
Input	Water Temperature	^{0}C
	Air Temperature	^{0}C
	Suspended Solid	mg/L
	Nitrate-N ($N\text{-}NO_3$)	mg/L
	Nitrite-N ($N\text{-}NO_2$)	mg/L
	Chemical Oxygen Demand	mg/L
	Dissolved Oxygen	mg/L
	Total Kjeldahl Nitrogen	mg/L
	Orthophosphate	μg/L
Output	Total Phosphorus	μg/L

The stopping criteria in the initial screen are:

1. Maximum number of epochs to train is 5000.
2. Performance goal (mean square error of the training result) is 0.
3. Minimum performance gradient is 1×10^{-10}.
4. Maximum validation failures equal to 1.

At each training epoch, the testing samples will be applied to validate the network training performance. If the mean square error is increased, the validation fails. It prevents the network over training. Once the candidate is chosen after the initial screen, this network will initialized and pass through the training process again without the limitation of criterion 1 listed above. Secondly, the initial weights in matrices *W* and *V* are generated within the interval, (0, 1), by a uniform random generator. The tansig function is applied as the activation function in the processing elements with α=1. The learning rate β is fixed as 0.25.

Figure 5.15 displayed the test performances of the initial screen of 41sub-BPNs with 10 to 50 hidden neurons in the hidden layer, evaluated by mean square errors between the predicted outputs and desired outputs of different sub-network testing. The x-coordinate represented the sub-networks with different hidden neurons from 10 to 50, and the y-coordinate represented their related mean square errors. The little text block indicated the BPN with 29 hidden neurons in the hidden layer has the minimum testing mean square error 9.265μg^2/L^2.

Figure 5.16 shows their correlation coefficients (*R*s), evaluated by Equation 5.2, which measure the strength and the direction of a linear relationship between the network outputs and the desired outputs. The value of *R* is such that $-1 \leq R \leq +1$. An *R* value of exactly +1 indicates a perfect positive fit, and an *R* value of exactly -1 indicates a perfect negative fit. If there is no linear correlation or a weak linear correlation, *R* is close to 0. The x-coordinate represented the sub-networks with different hidden neurons from 10 to 50, and the y-coordinate represented their related correlation coefficients. The little text block indicated the BPN with 29 hidden neurons has the maximum testing correlation coefficients 0.994.

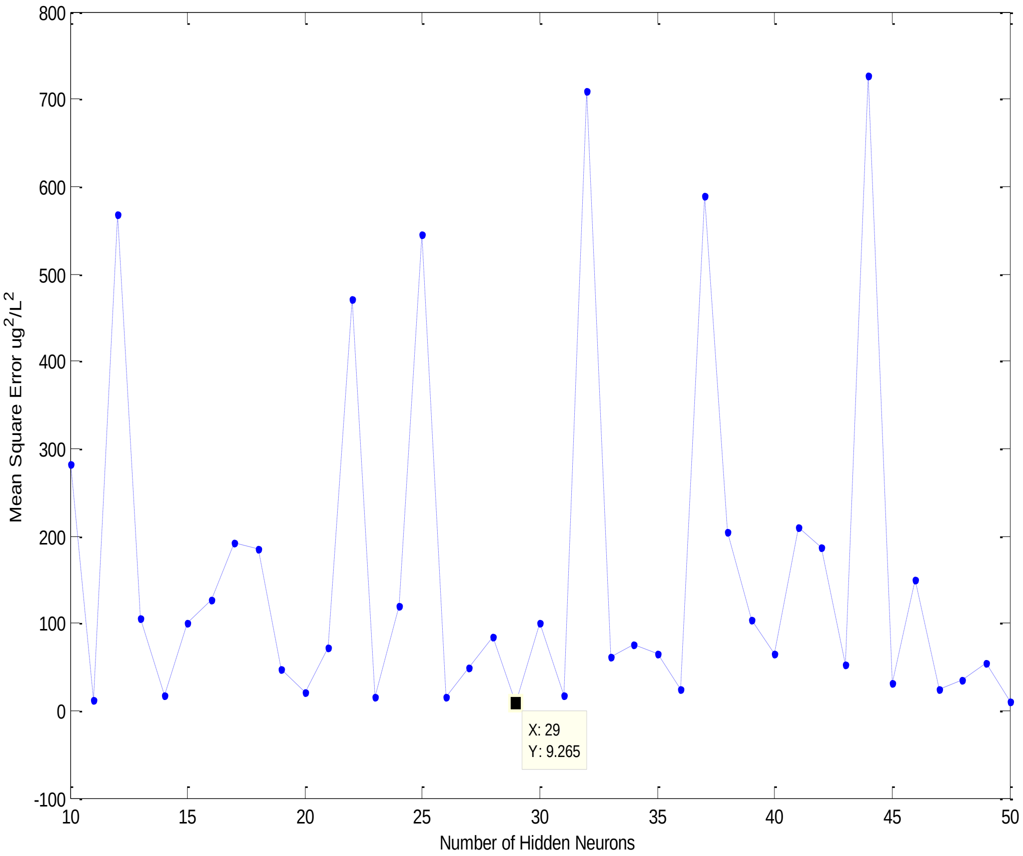


Figure 5.16 Testing Performances of the Initial Screen of 41sub-BPNs with 10 to 50 Hidden Neurons in the BPN Training Procedure of TP Concentration Modeling

Based on Figures 5.16 and 5.17, the best candidate will be the back-propagation neural network with 29 hidden neurons, which has the minimum mean square error 9.265 μg^2/L^2 and the maximum linear correlation coefficient 0.994 among all 41 sub-BPNs. Figure 5.16 and 5.17 also implicated that increasing the number of hidden neurons will not improve the network performance directly. It is possible that there is a better solution when more than 50 hidden neurons are applied in the hidden layer, but it will enlarge the

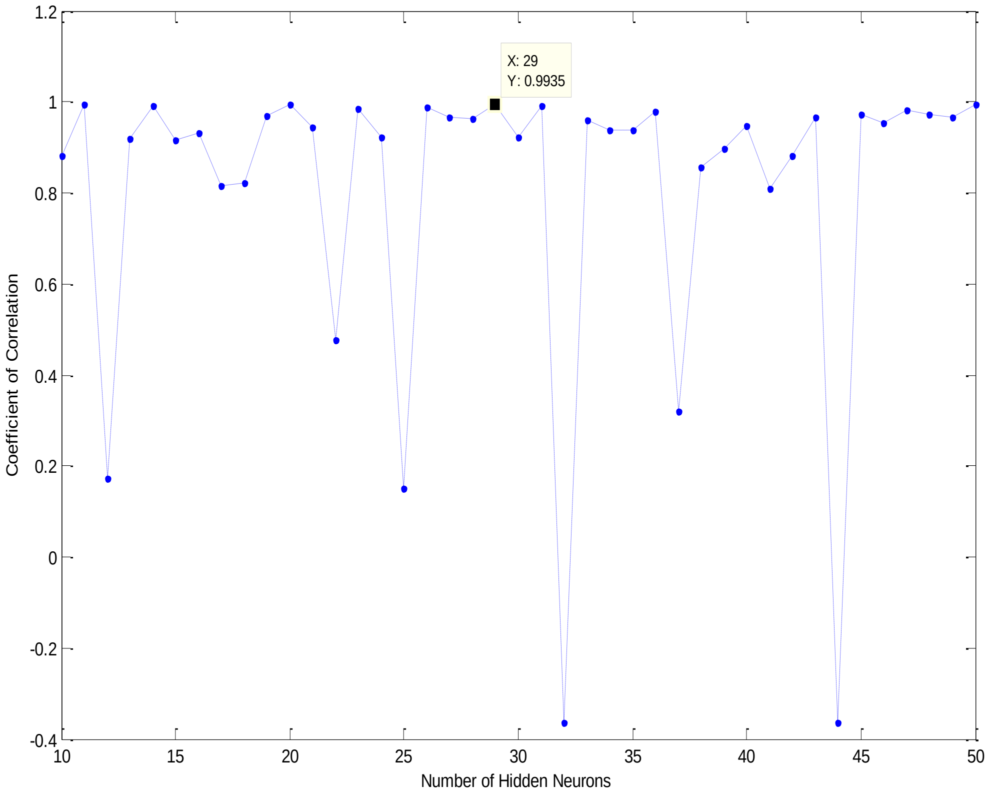


Figure 5.17 Correlation Coefficients of the Initial Screen of 41sub-BPNs with 10 to 50 Hidden Neurons in the BPN Training Procedure of TP Concentration Modeling

size of network, create more connections between each layer, increase the network training time and consume huge computation capacity. The candidate BPN with 29 hidden neurons will be initialized and retrained without the limitation of maximum training epochs. At 10102th training epoch, the validation check failed which means that the mean square error of testing results kept decreasing until reached the 10102th epoch. The local gradient δ at the output layer was decreasing to 2.313×10^{-4}. As a result, the best normalized validation (testing) performance of network (mean square error) is 5.502×10^{-4}at 10101th epoch. These facts are demonstrated in Figure 5.18 and 5.19.

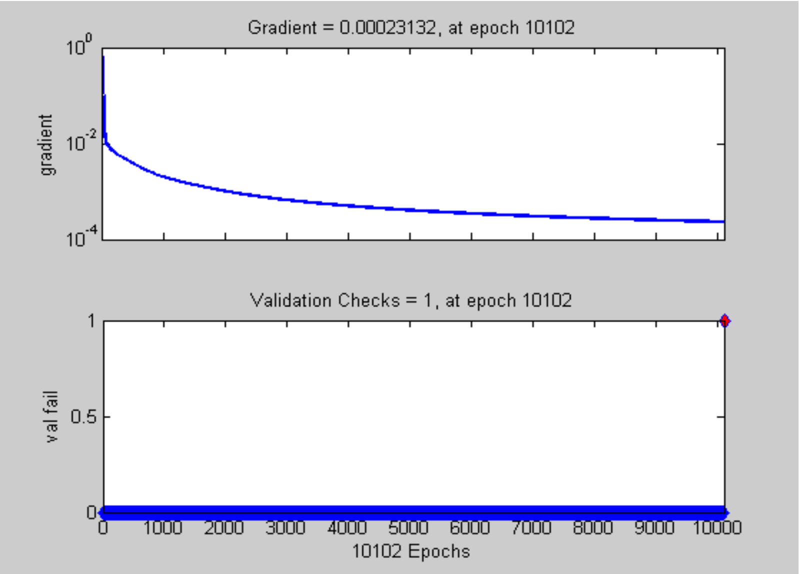


Figure 5.18 Training State Plots in the BPN Training Procedure of TP Concentration Modeling

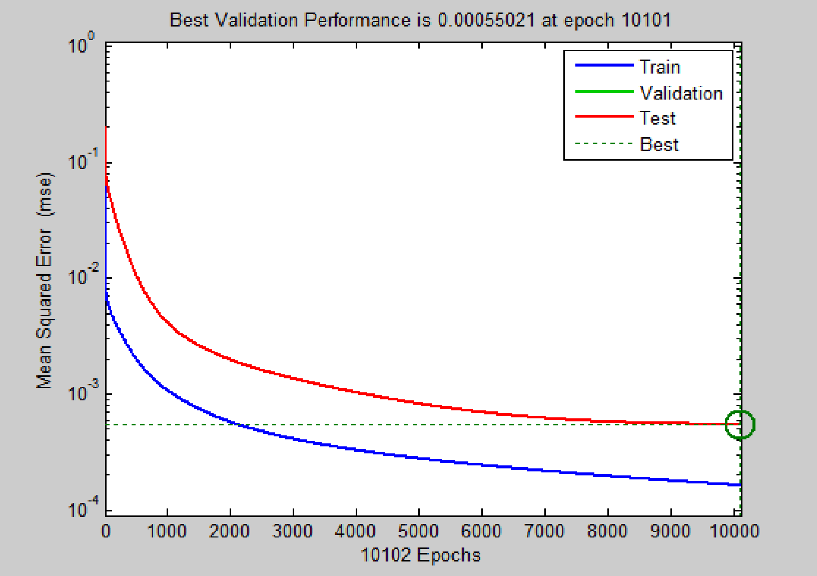


Figure 5.19 Performance Plot in the BPN Training Procedure of TP Concentration Modeling

Figure 5.20 shows the normalized linear regression plots of leachate flow rate modeling by using BPN. The upper left plot indicates the linear regression model in the training section with R = 0.989, and $Y_{net\_trian} \approx 0.95 \cdot Y_{trian} + 0.0064$. The upper right plot and lower left plot are same, because the validation and test samples are identical. The *Rs* = 0.996, and the linear regression model can be represented as $Y_{test} \approx 0.96 \cdot Y_{net\_test}$ +0.014. The lower right plot is a summary of three previous cases, the overall correlation coefficient is 0.995, and $Y_{net\_all} \approx 0.96 \cdot Y_{all} + 0.0087$.

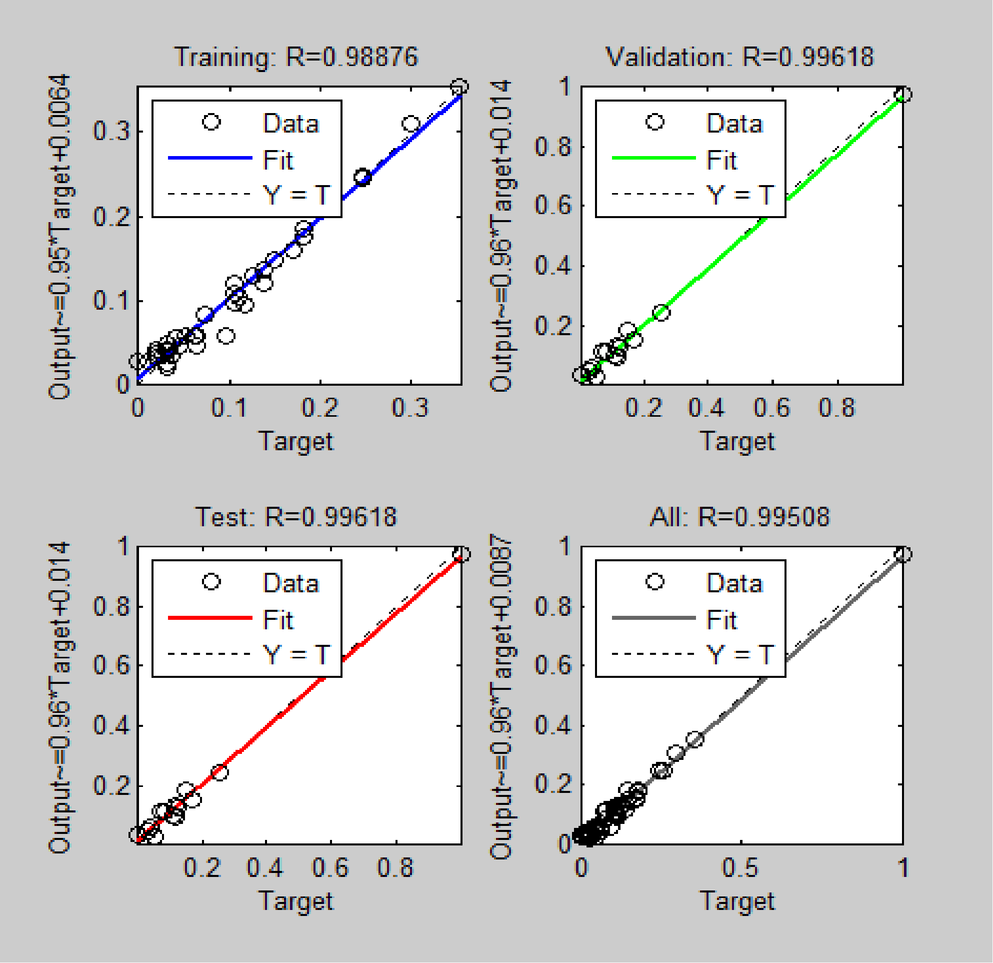


Figure 5.20 Regression Plots of TP Concentration Modeling by using BPN

After de-normalization, the testing network outputs were re-scaled, and the testing regression model was changed to $Y_{net\_test} \approx 0.96 \cdot Y_{test} + 1.5$, as shown in Figure 5.21, but the coefficient of correlation is same as the one before re-scaled.

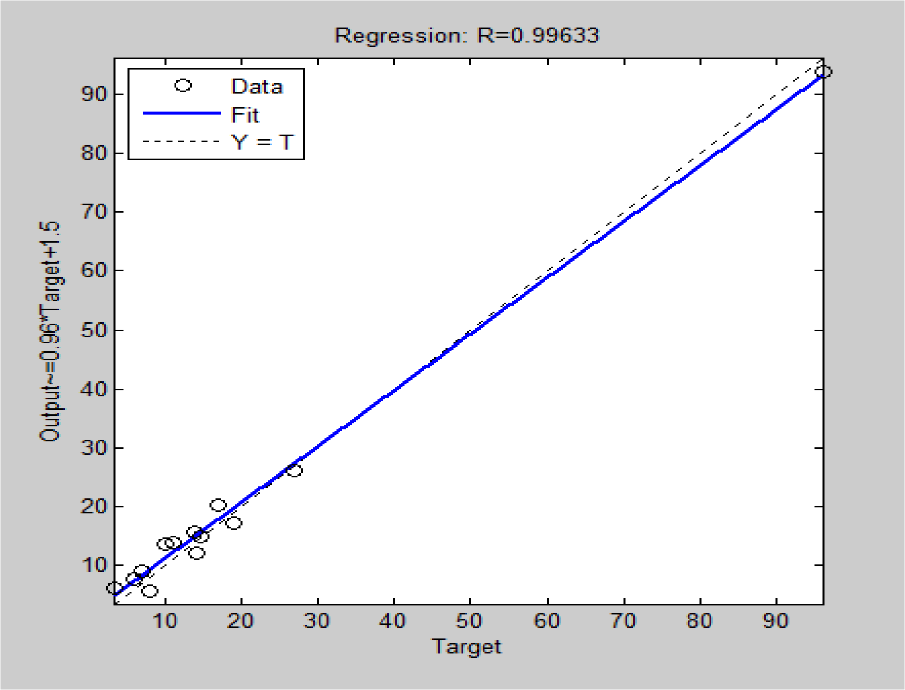


Figure 5.21 Testing Regression Plots of TP Modeling by using BPN

Figure 5.22 depicts the test performance of the back propagation neural network in original scale. The dash line curve with circle marker represents the desired TP concentration and the dash line curve with square marker represents the BPN predicted TP concentration. The dash line curve with triangle marker represents the error calculated by the desired TP concentration minus the BPN predicted TP concentration. The x coordinator represents 13 testing samples, and the y coordinator represents the total phosphorus concentration. The model successfully predicted the peak and valley values of the leachate flow rate within ±2 μg/L. The largest error happened at 12th testing sample may due to the similar sample or samples in the training section less excited or no similar sample or samples were trained in the training section. As a result, the network did not learn such information contained in 12th testing sample.

Finally the coefficient of determination R^2 is applied to evaluate the performance of the linear regression. Because R of the test section is 0.996, R^2= 0.992, which means99.2% of the total variation in the desired test output can be explained by the linear relationship between the desired test output and the BPN test output ($Y_{net\_test} \approx 0.96 \cdot Y_{test}$ +1.5), The other 0.8% of the total variation in of the desired test output remains unexplained. The mean square error of the final BPN total phosphorus concentration modeling is 5.075μg^2/L^2.

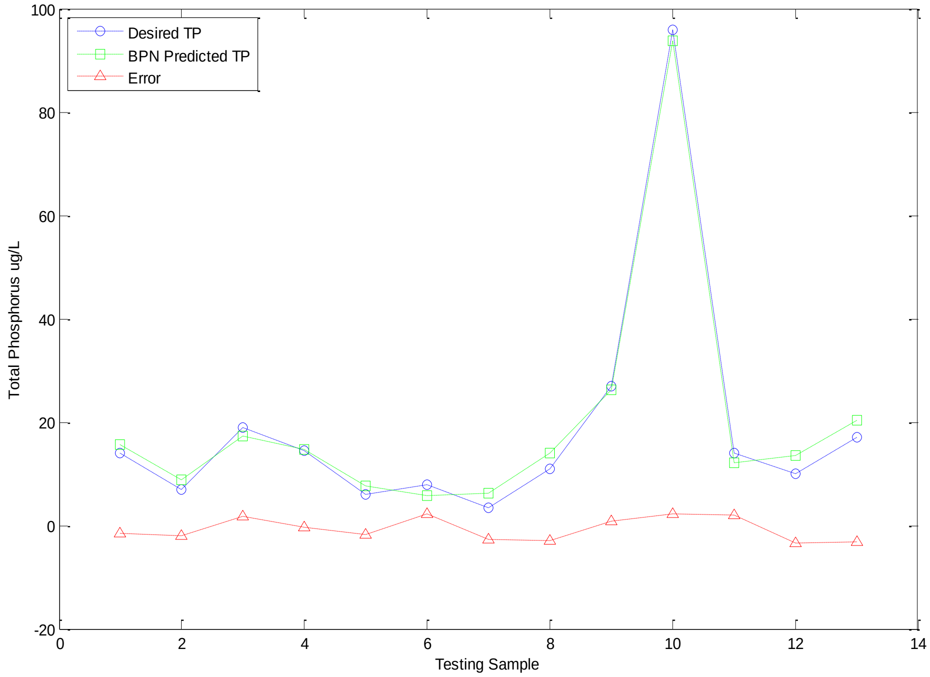


Figure 5.22 De-normalized Test Performance, Desired TP vs. BPN Predicted TP

5.3.5 RBFGRNN Model Description and Results in Case 2

The structure of RBFGRNN is different from that of BPN, as well as the learning algorithm. A RBFGRNN with defined centers and spread is a one-pass network, which means there is no iterative weight updating or calculations. As mentioned in chapter 4, the iterative process created for RBFGRNN is only aim to locate the optimal centers and spread. Definitely, testing performance decides the generalization ability of the proposed network and evaluates how well the network is learning the information given by the training samples. In this section, the training performance will be ignored and the testing performance will be amplified. Figure 5.23 shows the testing performances (mean square error) of a cluster of sub radial basis functional generalized regression neural networks. These subnets are varied by different number of centers and spread values. The mean square errors of different subnets are represented by different colors. The red color indicates high mean square error and the blue color indicate low mean square error. Based on recorded RBFGRNN testing performances, the RBFGRNN with 16 centers and *spread*=41 has the best testing performance MSE= 1.091×10^{-7}μg^2/L^2.

Figure 5.24 shows the linear regression plots. It indicates the linear regression model in the testing section with correlation coefficient R = 1, and $Y_{net\_test} \approx Y_{test}$ +0.0073. Figure 5.25 depicts the test performance of the radial basis functional generalized regression neural network. The dash line curve with circle marker represented the desired TP concentration and the dash line curve with square marker represented the RBFGRNN predicted TP concentration. The dash line curve with triangle marker represented the error calculated by desired TP concentration minus RBFGRNN

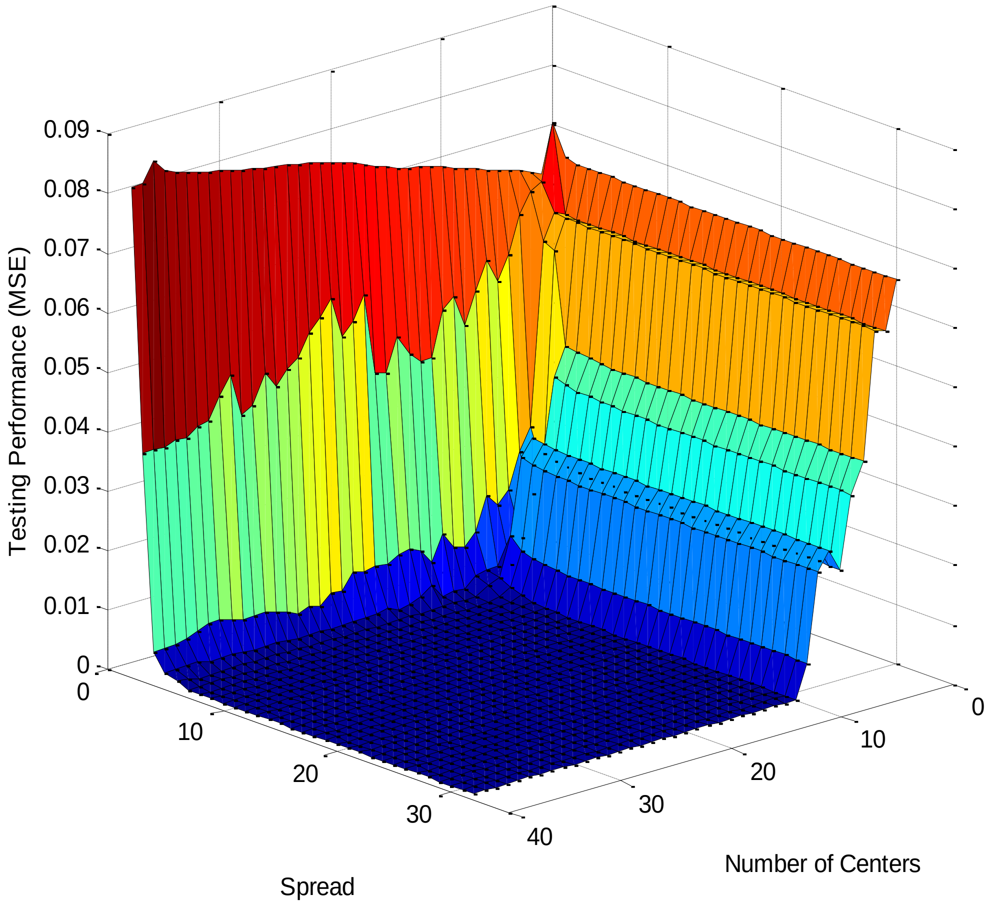


Figure 5.23 3D plot of RBFGRNN Testing Performance with Different Spreads and Centers in the TP Concentration Modeling

predicted TP concentration. The x coordinator represented 13 testing samples, and the y coordinator represented the TP concentration. The model successfully predicted the peak and valley values of the TP concentration.

The mean square error between the desired TP and the RBFGRNN predicted TP is$1.091\times10^{-7}\mu g^2/L^2$. The coefficient of determination R^2 is equal to 1, which means 100% of the total variation in the desired test output can be explained by the linear relationship between the desired test output and the RBFGRNN predicted output ($Y_{net\_test} \approx Y_{test}$ +0.0073).

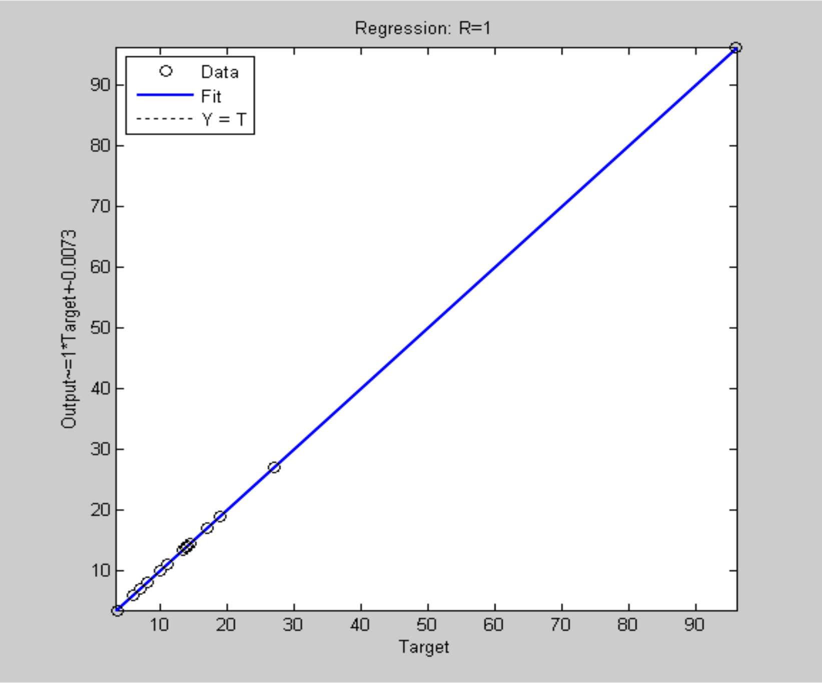


Figure 5.24 Linear Regression Plots in the RBFGRNN Testing Procedure of TP Concentration Modeling

5.3.6 Case Study 2 Summary

In this total phosphorus concentration modeling case, BPN and RBFGRNN are applied. Table 5.5 shows a performance summary of two networks. Compared with the BPN, RBFGRNN performed better evaluated by a lower mean square error, higher coefficient of correlation, and higher coefficient of determination. The R^2=1 in the RBFGRNN modeling stated that 100% of the total variation in the desired test output can be explained by the linear relationship between the desired test output and the RBFGRNN predicted output. In a short word, it works perfectly.

Table 5.5 Testing Performances of Two Neural Network Applications in Case 2

Networks	MSE μg^2/L^2	Coefficient of Correlation	Coefficient of Determination
RBFGRNN	1.091 ×10^{-7}	1	1
BPN	5.075	0.996	0.992

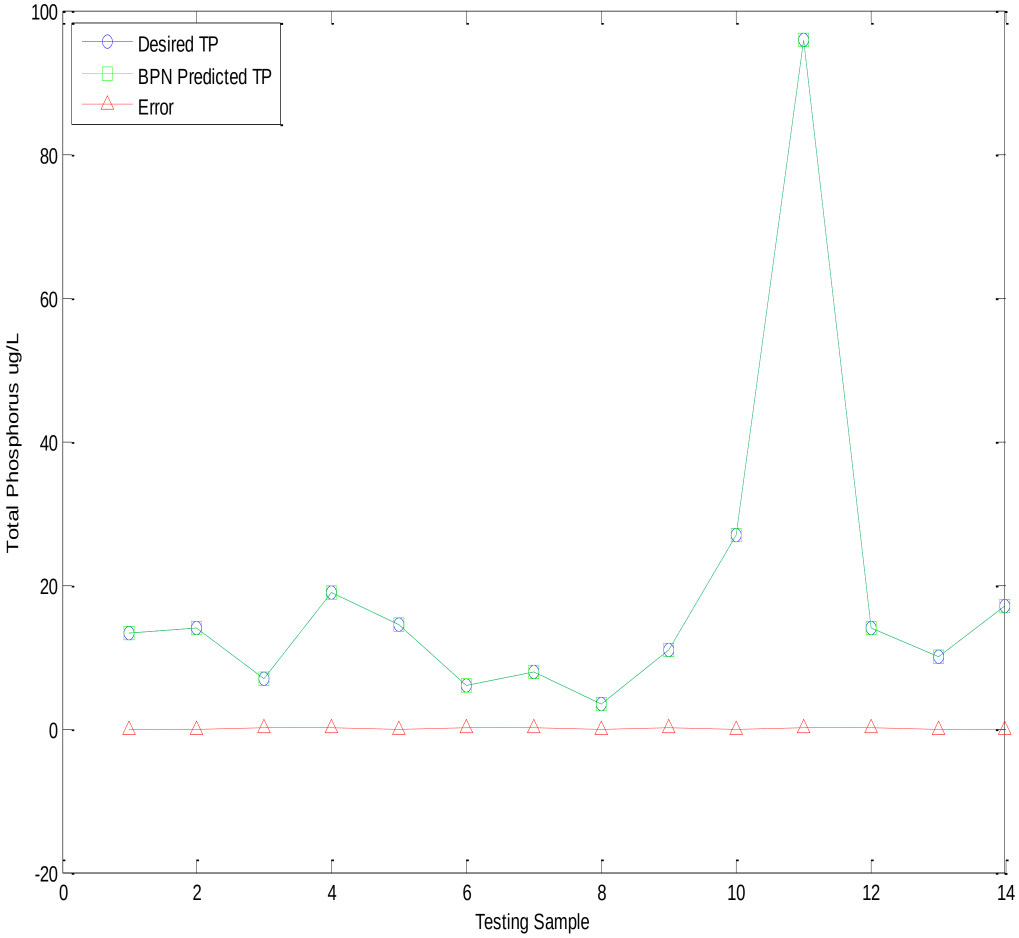


Figure 5.25 De-normalized RBFGENN Test Performance, Desired TP vs. RBFGRNN Predicted TP

CHAPTER 6
CONCLUSION

In this study, two different artificial neural networks, BPN and RBFGRNN, were applied to modeling two different environmental engineering systems synchronously. Based on the testing performances displayed in Table 5.3 and 5.5, the results of artificial neural networks applied in modeling of total phosphorus concentration is better than those of landfill leachate flow rate modeling. The major causations are concluded as following:

1. In the data collection section, the samples used in TP modeling are the real observations, recorded by 5 sampling stations in Te-Chi reservoir area; the samples used in landfill leachate flow modeling are generated by HELP model under randomly adjusting the values of 5 features and fixed others, which caused the difficulties to capture the universal underlying patterns in the Greensboro area.
2. In the data randomization, the patterns of samples used in testing section of TP modeling are well captured in the training section TP modeling, compared with the landfill leachate flow modeling. In other words, all of the special events are experienced or learned in the training section. It implicates that the variance of the samples used in TP modeling is smaller than that of samples used in leachate flow rate modeling. This issue may be solved by enlarging the size of data set.

In general, the major portion of the test error is caused by the unknown features which affect the corresponding environmental systems. However, environmental engineering systems are complex and associated with different biological, chemical, and other processes, so it is difficult to find out all of the features as the input elements for the network training. In case 1, there is 27.2% of the total variation in the desired test output cannot be explained by the linear relationship between the desired test output and the BPN test output and 17.7% of the total variation in the desired test output cannot be explained by the linear relationship between the desired test output and the RBFGRNN predicted output. Using different network is capable of reducing the prediction errors, but it can't overcome the lack of knowledge of the systems. In case 2, the testing results are much better, only 0.8% of the total variation in the desired test output remains unexplained by the linear relationship between the desired test output and the BPN test output, and 0% of the total variation in the desired test output remains unexplained by the linear relationship between the desired test output and the RBFGRNN test output. It proves that the 9 features selected as the input elements of two neural networks can fully represent the cause-and-effect of total phosphorus concentration in the Te-Chi Reservoir.

During implementing the RBFGRNN in study cases, the proposed supervised center selection method offers a large convenience for seeking the centers and spread which are needed in the Gaussian displacement functions. Compared with conventional unsupervised clustering method, the supervised center selection method reunited the center selection portion with RBFGRNN, and the next center and new spread are only decided by the network performance based on current centers and spread.

During implementing the BPN in study cases, it is important to find near optimal number of hidden neurons. As shown in Figure 5.3 and 5.16, the performances of different number of hidden neurons applied are dynamic. Even adding another hidden neuron in the hidden layer, the performance will upgrade or downgrade a lot in some scenarios. In this research, a trial-and-error process was applied to find the near optimal configuration of the BPN.

In this research, both of two networks performed successfully in modeling the environmental engineering systems. It verified the potential of artificial neural network methods in the application of complex systems, especially the environmental engineering systems.

REFERENCE

Burke H. B., Rosen D. B., and Goodman P. H.,1994, “Comparing Artificial Neural Networks to Other Statistical Methods for Medical Outcome Prediction Neural Networks”, IEEE World Congress on Computational Intelligence., Issue Date: 27 Jun-2 Jul 1994, Volume: 4 on page(s): 2213.

Chang S. Y., and Wang Y., 2009, “Prediction of Leachate Flow-Rate in a MSW Landfill Site Using Neural Network Method”, Journal of Solid Waste Technology and Management Volume 35, No. 2, May 2009.

Ferhat K., and Bestamin O., 2006, “NN-LEAP: A Neural Network-based Model for Controlling Leachate Flow-rate In a Municipal Solid Waste Landfill Site”, Environmental Modeling & Software Volume 21, Issue 8, page(s): 1190-1197, August 2006.

Goebel K., and Saha B., 2007, “Estimating Remaining Useful Life using Data-driven Techniques”, accepted for poster presentation at Data Mining in Aeronautics, Science, and Exploration Systems 2007 Conference, Mountain View, CA, June 2007.

Hill M. C., and Tiedeman C. R., 2007, “Effective Groundwater Model Calibration: With Analysis of Data, Sensitivities, Predictions, and Uncertainty”, Wiley and Sons, page: 464.

Khoa N. L. D., Sakakikara K., and Nishikawa, 2006, “I.: Stock price forecasting using back propagation neural networks with time and profit based adjusted weight factors”, in: SICE-ICASE International Joint Conference 2006, Busan, pp. 5484-5488.

Kramer A. H., and Sangiovanni-Vincentelli A., 1989, “Efficient Parallel Learning Algorithms for Neural Networks”, Advances in Neural Information Processing Systems 1 (Denver 1988) D. S. Touretzky, Editor, 40-48. Morgan Kaufmann, San Mateo.

Kuo J. T., Hsieh M. H., Lung W. S., and She N., 2007, “Using artificial neural network for reservoir eutrophication prediction”, Ecological Modeling, Volume 200, Issues 1-2, Pages 171-177, 10 January 2007.

Mennon A., Mehrotra K., Mohan C. K., and Ranka S., 1996, “Characterization of a class of sigmoid functions with application to neural networks”, *Neural Networks,* vol. 9, pp 819-835.

Możejko J., and Gniot R., 2008, “Application of Neural Networks for the Prediction of Total Phosphorus Concentrations in Surface Waters”, Polish J. of Environ. Stud. Vol. 17, No. 3, Pages: 363-368.

Nadaraya E. A., 1965, “On Nonparametric Estimates of Density Functions and Regression Curves”, Theory Appl. Probability 10, page(s): 186–190.

Parker D. B., 1987, “Optimal algorithms for adaptive networks: Second order back propagation, second order direct propagation, and second order Hebbian learning,” IEEE 1st International Conference on Neuron Network, vol.2, pp.593-600, San Diego, CA.

Werbos P., 1974, “Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences”, PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA, November 1974.

Ray B. T., 1994, “Environmental Engineering”, PWS Publishing Company. ISBN 0-534-20652 2 Pages: 236-237.

Rumelhart D. E., Hinton G. E., and Williams R. J., 1986, “Learning Representations by Back-Propagating errors”, NATURE, Volume 323, Issue 9, Page 533-536, October 1986.

Schwabacher M., 2005, “A Survey of Data-Driven Prognostics.” AIAA Infotech @Aerospace Conference.

Haykin S., 1998, “Neural Network: A Comprehensive (2nd Edition)”, Prentice Hall ISBN-10:0132733501, ISBN-13: 978-0132733502.

Specht D. F., 1991, “A General Regression Neural Network”, IEEE Transaction on Neural Networks, Vol.2, No. 6, pages: 568-576.

Tu J. V., 1996, “Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes.” Journal of Clinical Epidemiology, Volume 49, Issue 11, Pages 1225–1231, November 1996.

Zimmerman D. A., Marsily G., Gotway C. A., Marietta M. G., Axness C. L., Beauheim R. L., Bras R. L., Carrera J., Dagan G., Davies P. B., Gallegos D. P., Galli A., G ómez-Hern ández J., Grindrod P., Gutjahr A. L., Kitanidis P. K., Lavenue A. M., McLaughlin D., Neuman S. P., RamaRao B. S., Ravenne C., and Rubin Y., 1998, “A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow”, Water Resource Research, Vol. 34, No. 6, Page 1373.