DISCHARGE PREDICTIONS USING ANN IN SLOPING RECTANGULAR CHANNELS WITH FREE OVERFALL

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAYRULLAH UĞRAŞ ÖZTÜRK

IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

OCTOBER 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof.Dr. Erdal Çokça
Head of Department

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

Assist. Prof. Dr. Şahnaz TİĞREK
Co-Supervisor

Prof. Dr. Metin GER
Supervisor

Examining Committee Members

Prof. Dr. A.Melih YANMAZ	(METU,CE)	
Prof. Dr. Metin GER	(METU,CE)	
Assist. Prof. Dr. Şahnaz TİĞREK	(METU,CE)	
Assoc. Prof. Dr. Can BALAS	(Gazi U,CE)	
Dr. Yurdagül KAYATÜRK	(DSİ)	

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Hayrullah Uğraş ÖZTÜRK

Signature :

ABSTRACT

DISCHARGE PREDICTIONS USING ANN IN SLOPING RECTANGULAR CHANNELS WITH FREE OVERFALL

ÖZTÜRK, Hayrullah Uğraş

M.Sc., Department of Civil Engineering

Supervisor: Prof. Dr. Metin GER

Co-Supervisor: Assist. Prof. Dr. Şahnaz Tiğrek

October 2005, 45 pages

In recent years, artificial neural networks (ANNs) have been applied to estimate in many areas of hydrology and hydraulic engineering. In this thesis, multilayered feedforward backpropagation algorithm was used to establish for the prediction of unit discharge q ($m^3/s/m$) in a rectangular free overfall. Researchers' experimental data were used to train and validate the network with high reliability. First, an appropriate ANN model has been established by considering determination of hidden layer and node numbers related to training function and training epoch number. Then by applying sensitivity analysis, parameters involved in and their effectiveness relatively has been determined in the phenomenon. In the scope of the thesis, there are two case studies. In the first case study, ANN models reliability has been investigated according to the training data clustered and the results are given by comparing to regression analysis. In the second case, ANN models' ability in establishing relations with different data clusters is investigated and effectiveness of ANN is scrutinized.

Key Words: Artificial Neural Networks (ANN), Brink Depth, Free Overfall, Discharge Measurement

ÖZ

SERBEST DÜŞÜLÜ EĞİMLİ DİKDÖRTGEN KESİTLİ KANALLARDA YSA İLE DEBİ TAHMİNİ

ÖZTÜRK, Hayrullah Uğraş
Yüksek Lisans, İnşaat Mühendisliği Bölümü
Danışman: Prof. Dr. Metin GER
Yardımcı Danışman: Yard. Doç. Dr. Şahnaz Tiğrek
Ekim 2005, 45 sayfa

Son yıllarda, yapay sinir ağları (YSA) hidroloji ve hidrolik mühendisliğinde etkili tahminler yapmakta oldukça geniş bir alanda kullanılmıştır. Bu tezde de, birim debi tahmininde, dikdörtgen kesitli serbest düşülü eğimli kanallarda, çok katmanlı, ileri beslemeli, geri yayınım algoritmalı bir YSA mimarisi kullanılmıştır. Çeşitli araştırmacılar tarafından toplanan veriler ağın eğitiminde ve testinde kullanılmıştır. En uygun modelin oluşturulmasında, eğitim fonksiyonu ve iterasyon sayısına bağlı olarak değişen, gizli katman ve nöron sayıları hesaplanmıştır. Daha sonra duyarlılık analizi uygulanarak, olaydaki girdi parametrelerin etkinlik dereceleri araştırılmıştır. Tezin kapsamı dahilinde iki örnek çalışma yapılmıştır. YSA modellerinin farklı eğitim kümelerine göre güvenirliliği test edilmiş, sonuçlar regresyon analizi sonuçlarıyla karşılaştırılmıştır. İkincisinde ise, oluşturulan YSA modellerinin farklı veri kümeleriyle ilişkiler kurabilme yeteneği incelenmiş ve YSA'nın etkinliği irdelenmiştir.

Anahtar Kelimeler: Yapay sinir ağları (YSA), Düşü Akım Derinliği, Serbest Düşü, Debi Ölçümü.

ACKNOWLEDGMENTS

This study was suggested and has been completed under the supervision of Prof. Dr. A. Metin GER in Hydromechanics Laboratory of Civil Engineering Department at the Middle East Technical University in Ankara, Turkey.

The author is indebted to Dr. Şahnaz TİĞREK for her helpful guidance and precious suggestions throughout this study.

The author would also like to thank to Prof. Dr. A. Metin GER for her supervision and supportive suggestions throughout this study.

Special thanks go to İhsan Kutlu and Emrah Doğan for their kind assist and support.

Special thanks also go to my family for support and encouraging me with endless patience and love throughout this period.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

LIST OF SYMBOLS

b	channel width
E	error
f	an activation function
g	the acceleration due to gravity
i	inner layer
j	hidden layer
k	output layer
m	total number of inputs to the node j
n	Manning's roughness coefficient
N	number of patterns
p	index number for pattern
q	discharge per unit width
Q	discharge
q_x	number of computed unit discharge in threshold statistics
r^2	root mean square
Sj	weighted sum
S_0	bed slope
u_i	output from the node i in the previous layer
w	weight function
w_{ij}	weight between nodes i and j
w_{oj}	bias function
y_c	critical depth

List of Symbols

y_e	brink depth
y_o	upstream normal depth
y_j	output from node j
μ	viscosity of water
ρ	density of water
t_i	target value for i^{th} pattern
o_i	output of the i^{th} pattern
R^2	correlation coefficient
t_{mean}	mean target value

ABBREVIATIONS

TS	Threshold Statistics
MSE	mean sum of squares
ANN	Artificial Neural Network
ANNs	Artificial Neural Networks

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

The overfall refers to the downstream portion of a rectangular channel, horizontal or sloping, terminating abruptly at its lower end. If it is not submerged by the tail water, it is referred to as the free overfall (Rajaratnam and Muralidhar, 1976).

The free overfall has a distinct importance in hydraulic engineering; it forms the starting point in computations of the surface in a gradually varied flow such as the discharge spills into an open reservoir at the downstream end. Typical free overfall and the parameters are illustrated in Figure 1.1.

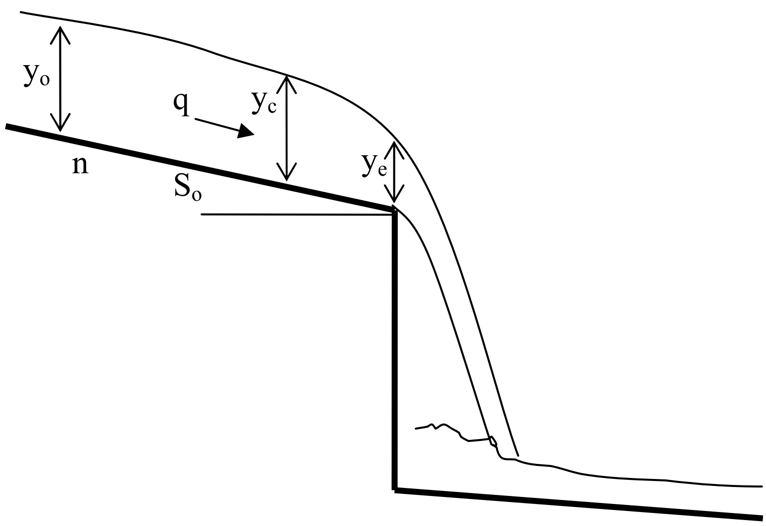

Figure 1.1. Typical free overfall and parameters involved in.

In this Figure, S_o (m/m) is the bed slope, n is Manning's roughness coefficient, y_e (m) is the brink depth, y_o (m) is the normal depth, y_c (m) is the critical depth, q (m^3/s/m) is the unit discharge of a flume of width b(m).

The study of a free overfall is also important because it can be used as a discharge-measuring device. The problem of the free overfall as a discharge measuring device has been investigated for 70 years and the end depth discharge relationship has been extensively studied by carrying out the theoretical and experimental studies by establishing a relationship between the critical depth, y_c and the brink depth (end depth), y_e. Rouse (1936) was the first to point out the possibility of using the free overfall as a flow meter, which needs no calibration (Firat, 2004).

Theoretical approaches are studied by many researchers. But these theoretical solutions on free overfall based on some sort of the assumptions. Even though giving promising answers, inadequacies of the studies lead researchers to pursue working on the issue. Detailed information can be found in Özsaraç (2001).

1.2 Scope of the Thesis

In the present study, it is tried to obtain an approximation with the experimental data in the literature by using artificial neural networks (ANN).

In the literature there are many attempts to obtain a relation between brink depth and discharge in channels (having cross sections rectangular, trapezoidal, triangular, circular and parabolic). Studies to find such a relation can be classified as; theoretical approaches and experimental approaches.

As for the experimental studies on this field, they are studied by numerous researchers. Among them, Rajaratnam and Muralidhar (1976), Ferro (1999), Davis et al. (1998), Turan (2002), Fırat (2004), and Kutlu (2005) have conducted quite a few numbers of experiments. For the experimental studies, most of the time to collect data on each variable could not be obtained due to the physical restrictions in the laboratory environment. For this reason, collected data are not sufficient to provide information out of data range. So the need for another method is warranted to complete missing data. At this point, soft computational methods may be useful because of their high ability to establish firm relations between input and output parameters.

In this study an approach has been suggested to relate discharge Q to brink depth y_e, and the other effective parameters (bed slope, S_o, Manning's roughness coefficient, n, and channel width, b), by using one of the soft computation methods, called Artificial Neural Networks (ANN).

Raikar et al. (2004) presents the application of ANN to determine the end depth ratio for a smooth inverted semicircular channel based on data of Dey (2003). In their study, discharge related to the slope and brink depth. However, in this study an analysis will be demonstrated for rectangular free overfall. Also, surface roughness and channel width will be included.

In the literature, as a soft computational method, ANNs have not yet been used as a predictive tool in rectangular free overfall.

CHAPTER 2

IMPORTANT ASPECTS OF ANN MODELLING

There are no fixed rules for developing an ANN, even though a general framework can be followed based on previous successful applications in engineering. Some issues that typically arise while developing an ANN are briefly described in this section.

2.1 Components of ANNs

Artificial Neural Networks (ANNs) consist of large number of processing elements with their interconnections. ANNs are basically parallel computing systems similar to biological neural networks. They can be characterized by three components:

- Nodes
- weights (connection strength)
- An activation (transfer) function

In ANNs architecture, there are layers and nodes at each layer. Each node at input and inner layers receives input values, processes and passes to the next layer. This process is conducted by weights. Weight is the connection strength between two nodes. The most commonly used ANN is the three-layer feed-forward ANN.

2.2 Three-Layered Feed-forward ANN

A typical three-layer feed-forward ANN, consists of a layer of input nodes, a single layer of hidden nodes, and a layer of output nodes, as shown in Figure 2.1. In the Figure, i, j, k denote nodes inner layer, hidden layer and output layer, respectively. w is the weight of the nodes. Subscripts specify the connections between the nodes. For example, w_{ij} is the weight between nodes i and j. The term "feed-forward" means that a node connection only exists from a node in the input layer to other nodes in the hidden layer or from a node in the hidden layer to nodes in the output layer; and the nodes within a layer are not interconnected to each other.

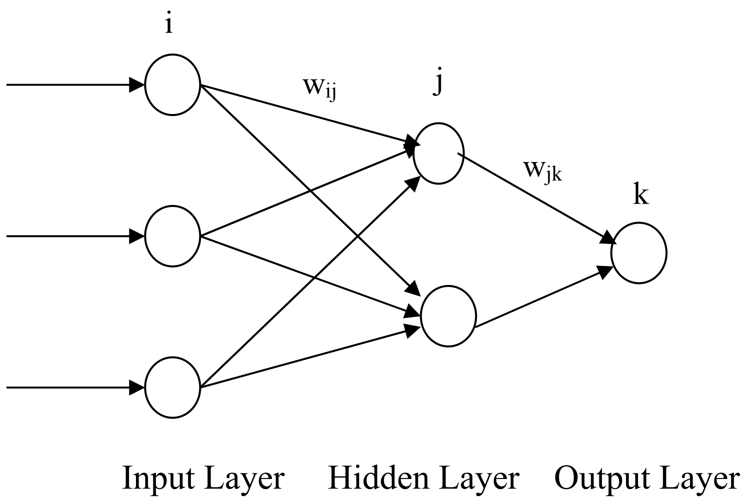

Figure 2.1. A typical three-layer feed forward ANN

Each node in the input layer receives an input variable and passes it to the nodes in the hidden layer. In addition, a bias node, which is also a weight with a fixed input, 1.0, is usually added to the input layer and to the hidden layer. The nodes in the hidden layer and in the output layer are nonlinear nodes meaning the weights multiplied by inputs. A typical node in the hidden layer or in the output layer is shown in Figure 2.2. Here w_{oj} is the bias function.

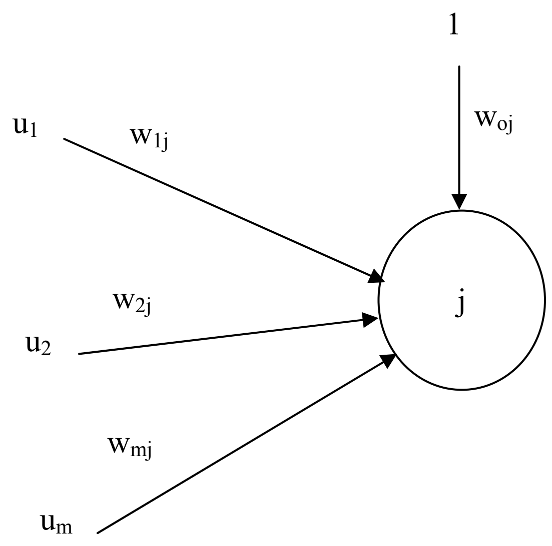

Figure 2.2. A typical node in the hidden layer or the output layer

The input to the node j, is designated as S_j. S_j is the weighted sum of all the incoming inputs, which are the outputs from the nodes in the previous layer. It can be mathematically represented as;

$$S_j = \sum_{i=0}^{m} w_{ij} u_i \tag{2.1}$$

The parameters in this equation are as follows:
w_{ij} is the weight between nodes i and j;
u_i is the output from the node i in the previous layer;
m is the total number of inputs to the node j;
y_j which is the output from node j, is calculated using an activation function.
Activation function determines the response of a node to the total input it receives.
The sigmoid function is the most commonly used one (ASCE 2000a), given as;

$$y_j = f(S_j) = \frac{1}{1+\exp(-S_j)} = \mathrm{logsig}(S_j) \tag{2.2}$$

A logsig transfer function is given in Figure 2.3.

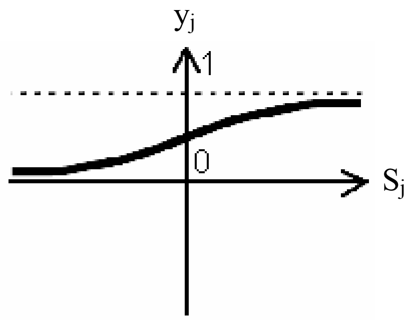

Figure 2.3. Logsig transfer function

Sigmoid functions are used to bound the outputs of the weighted sum of all the incoming inputs S_j. Whatever the output of S_j becomes, the result will be limited to [0, 1] interval by sigmoid function in a nonlinear manner. Since, it is easy to take derivative of sigmoid function; it is more popular than any other functions.

2.3 Backpropagation of Neural Networks

Backpropagation models, in a feedforward architecture, contain three components. They are an input layer, an output layer and at least one hidden layer. All those layers are fully connected to each other as shown in Figure 2.4 a-b.

In backpropagation algorithm there are two main steps. The first step is a forward pass, which is also called as activation phase. In that step, inputs are processed to reach the output layer through the network. After the error is computed, a second step starts backward through the network, which is also called as error backpropagation.

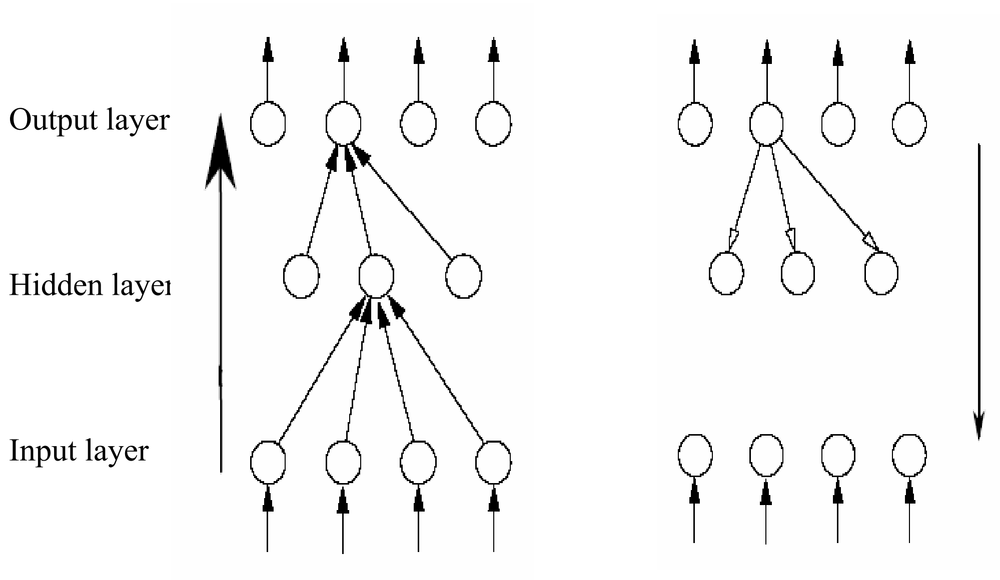

Figure2.4 Backpropagation of Neural Networks

a) Feedforward of inputs b) Backpropagation of output error

There are three types of learning in ANNs. These are supervised, reinforced and unsupervised learning. In supervised learning, true answer is given to the system. In reinforced learning, only, whether the answer is of the system produced is told as true or not. In unsupervised learning, correct or wrong answers are not differentiated.

Backpropagation process is conducted by supervised learning. Because, the output of the system delivered is compared to the exact values. Most of the backpropagation models also employ a delta learning rule, which requires the continual backpropagation of an error term from the output layer back to the input layer. The delta rule is one of the most commonly used learning rules. For a given input vector, the output vector is compared to the correct answer. Then, the weights are adjusted to reduce this difference. This procedure is applied until the difference between the actual and predicted outputs is less than preassigned value of maximum error. For ANN architectures with no hidden units, it is guaranteed to reach the global minima, since error space will always be a parabolic shape. But, as in our case, there will be at least one hidden layer. As shown in Figure 2.5, there is no one minimum point. In this case, generalized delta rule is used to adjust the weights.

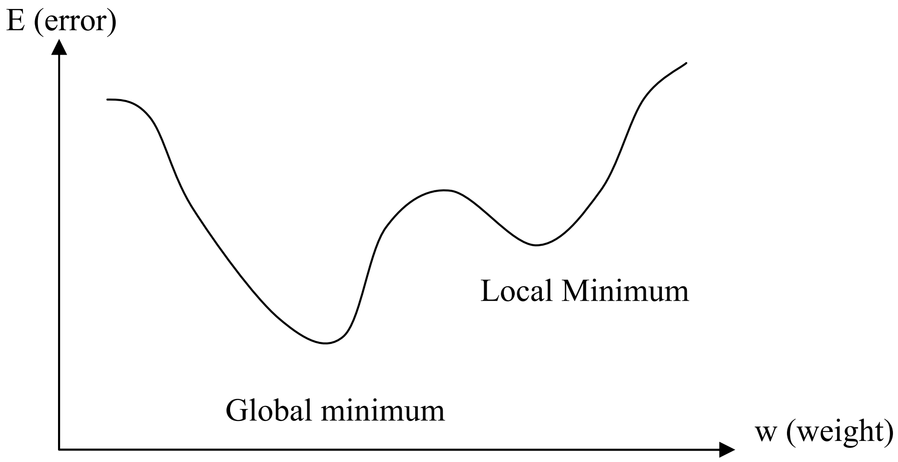

Figure 2.5. A typical weight (w) versus Error (E) graph for an ANN architecture with hidden layer(s).

During the training phase, an error value, usually mean square error (MSE) is calculated between the desired output and the actual output. The MSE is then propagated backwards to the input layer and the connection weights between the layers are readjusted. After the weights have been adjusted and the hidden layer nodes have generated an output result, the error value is again re-determined. If the error has not reached, which is usually defined by a particular iteration number, the error will then again be propagated backwards to the input layer. This procedure continues until the model has finally reached to the predetermined tolerance limit.

The weights in backpropagation algorithm are adjusted according to the direction in which the performance function, in this study MSE, decreases rapidly. Although the function decreases most rapidly along the steepest descent direction (negative of the gradient), it may not produce the fastest convergence. A search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions (Matlab 7.0, 2004).

Before the training phase begins, parameters involved in the phenomenon must be defined. In addition to declaring the total number of input nodes, the number of hidden layer nodes and the total number of iteration (epoch) must also be declared. Since most of the time, the performance of the network is affected by the number of the nodes in a hidden layer. To avoid failing to reach convergence, it is recommended the total number of hidden layer nodes be at least three times the total number of input layer nodes.

First of all, the input values are presented to the network and a desired output is determined. Next, the total number of hidden layer numbers and the hidden layer nodes are specified, as well as the number of iterations of the network. Initialization of training phase, the weights are given in a random manner to be used in an activation function. Since backpropagation models are sensitive to initial weight values, the weight values are randomized in order to ensure that the delta learning rule does not reach convergence prematurely (Merwin, 2004). In Figure 2.6, a typical backpropagation flow chart is given.

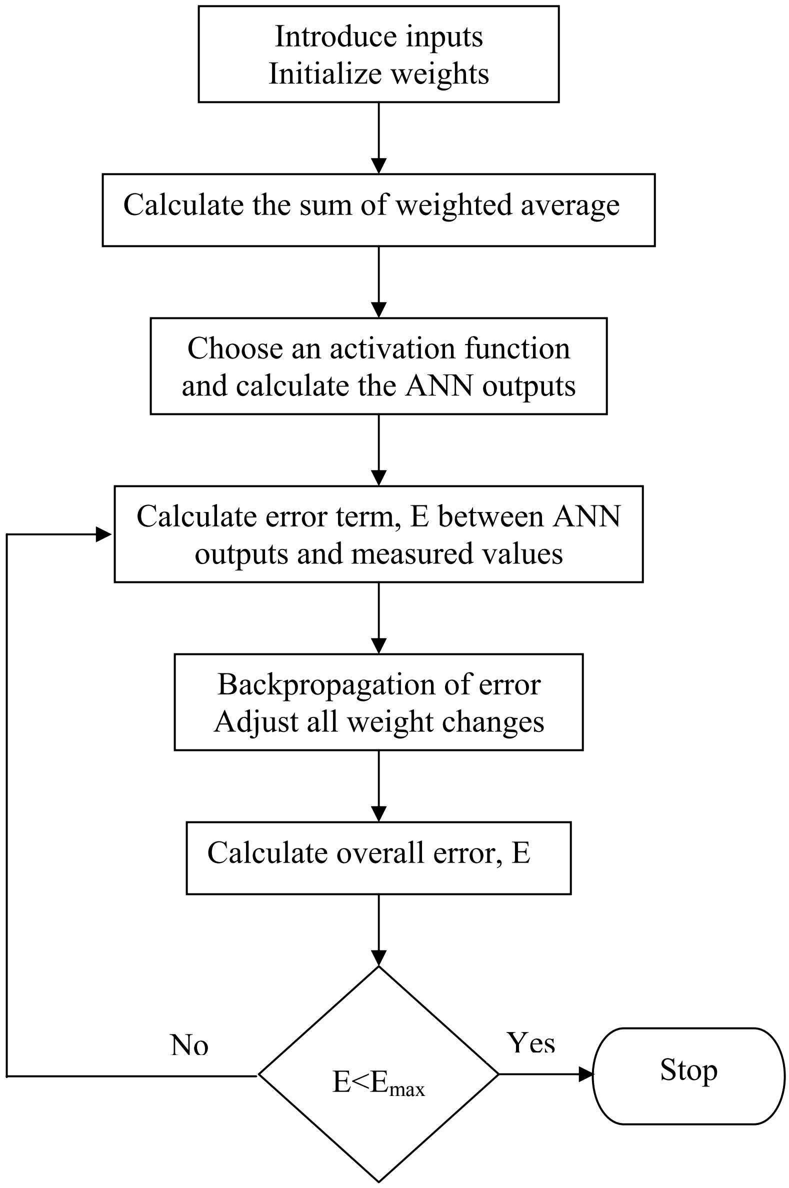

Figure 2.6. Flowchart of backpropagation algorithm.

2.4 Literature Survey

Attempts to simulate the human brain date back to 1930s. In 1986, on Artificial Neural Network, there has been a tremendous growth in its computational mechanism by Rumerhalt et al. Within the last decade, ANN has become a powerful computation tool due to the development of more sophisticated algorithms. After these improvements, they have been applied to many branches of the science.

The first article on a civil/structural engineering application of neural networks was published by Adeli and Yeh (1989). Since then, a large number of articles have been published on civil engineering applications of neural networks (Adeli, 2001).

As an example, Raikar et al. (2004) work briefed below. They applied ANN to determine the end-depth-ratio (EDR) for a smooth inverted semi-circular channel for subcritical and supercritical flows. In their study, four-layered feed forward ANN model is used to analyze the experimental data. The sigmoidal function is used as an activation function. In the hidden layer, they have tried 3,5, and 7 nodes to find out optimum ANN architecture. They have divided their data set to 50:50 and 65:35 as training and test data keeping Manning's roughness coefficient constant and with three different diameters of the channel cross section. They have concluded that the results of the ANN gives close answers to the target values in the study of Dey (2003). For subcritical flow, the EDR has been predicted. For supercritical flow an empirical relationship is given for discharge calculation.

2.5 Performance Functions

A performance function is used in ANNs to evaluate the effectiveness of the output of the network. The typical performance function that is used for training feedforward neural networks is the mean sum of squares (MSE) of the network errors.

$$MSE = \frac{1}{N}\sum_{p=1}^{N}\left(t_p - o_p\right)^2 \tag{2.3}$$

where, N is the number of the patterns; t_p is the target value for the p^{th} pattern; o_p is the output of the p^{th} pattern which is produced by ANNs.

The performance control of the ANN outputs was evaluated by estimating the correlation coefficient (R^2) which is defined as:

$$R^2 = \frac{\sum_{p=1}^{N}(t_p - t_{mean})^2 - \sum_{p=1}^{N}(t_p - o_p)^2}{\sum_{p=1}^{N}(t_p - o_p)^2} \qquad (2.4)$$

where, t_{mean} is the mean target value.

Threshold Statistics (TS) for a level of x% is a measure of the consistency in forecasting errors from a particular model. The TS are represented as TS_x and expressed as a percentage. This criterion can be expressed for different levels of absolute relative error from the model. It is defined as:

$$TS_x = \frac{o_x}{N} \cdot 100 \qquad (2.5)$$

where, o_x; is the number of computed values (out of N total computed) for which absolute relative error is less than x% from the model (Doğan et al., 2005).

2.6 Developing an ANN model

There are no fixed rules for developing an ANN, even though a general framework can be followed based on previous successful applications in engineering. Some issues that typically arise while developing an ANN are briefly described in this section.

2.6.1 Selection of Input and Output Variables

In order to use ANN structures effectively, input variables in the phenomenon must be selected with great care. This highly depends on the better understanding of the problem. In a firm ANN architecture, in order to prevent confusing training process key variables must be introduced and unnecessary variables must be avoided. For this purpose, a sensitivity analysis can be used to find out the key parameters. Also sensitivity analysis can be useful to determine the relative importance of the parameters when sufficient data are available (ASCE 2000a).

2.6.2 Sensitivity Analysis

The sensitivity analysis is used to determine the effect of changes and to determine relative importance or effectiveness of a variable on the output. The input variables that do not have a significant effect on the performance of an ANN can be excluded from the input variables, resulting in a more compact network. Then, it becomes necessary to work on methods like sensitivity analysis to make ANNs work effectively (ASCE, 2000).

2.6.3 Designing an ANN

This important step involves the determination of the ANN architecture and selection of a training algorithm. An optimal architecture may be considered the one giving the best performance in terms of error minimization, while retaining a simple and compact structure. There is no specific information for determination of such an optimal ANN architecture. Often, more than one ANN can generate similar results. The numbers of input and output nodes are problem dependent. The flexibility lies in selecting the number of hidden layers and in assigning the number of nodes to each of these layers as well as in the number of iteration. A trial-and-error procedure is generally applied to decide on the optimal architecture.

Feedforward neural networks depend on two main factors. Feedforward neural networks can be used without needing any explicit mathematical equation relating inputs and outputs. This shows the computational superiority of ANNs. Also, feedforward network with a hidden layer or hidden layers without considering the number of sigmoidal hidden nodes can approximate any continuous function (ASCE, 2000a). This feature of ANNs points the high capacity in establishing relations between inputs and outputs.

The number of hidden layer nodes significantly influences the performance of a network. Both too few and too many nodes in the hidden layer lead the system to poor performance. If there are too few nodes in the hidden layer, the problem of underfitting, if there are too many nodes in the hidden layer the problem of overfitting is encountered. Thus, optimum number of nodes in the hidden layer must be selected. This process will be explained in Chapter 3.

2.6.4 Training and Testing

Training and testing concept can be understood as a calibration process. The available data set is generally grouped into two parts, one for training and the other for testing.

The purpose of training is to determine the set of connection weights that cause the ANN to estimate outputs within the given tolerance limits to target values. The data set reserved for training is used for this purpose. This grouping of the complete data to be employed for training should contain sufficient patterns so that the network can learn the underlying relationship between input and output variables adequately. That is why the training part generally consists of a large percentage the data available. In the literature, there is no specific rule while grouping total data into training and test divisions.

The weights are assigned small random values initially. During training, these are adjusted based on the error, or the difference between ANN output and the target responses. This adjustment continues until a weight space is found, which results in the smallest overall prediction error. However, there is the danger of overtraining a network in this fashion, which is known as overfitting. This happens when the network parameters are too fine-tuned to the training data set. It is as if the network, in the process of trying to ‘‘learn’’ the underlying rule, has started trying to fit the noise component of the data as well. In other words, overtraining results in a network that memorizes the individual examples, rather than trends in the data set as a whole. When this happens, the network performs very well over the data set used for training, but shows poor predictive capabilities when supplied with data other than the training patterns. This case can be thought as “memorization” rather than “learning”. To prevent this kind of overfitting, testing procedure is usually recommended. The goal of this procedure is to stop training when the network begins to overtrain. The second part of the data is reserved for this purpose. In Figure 2.7a-b this situation is shown.

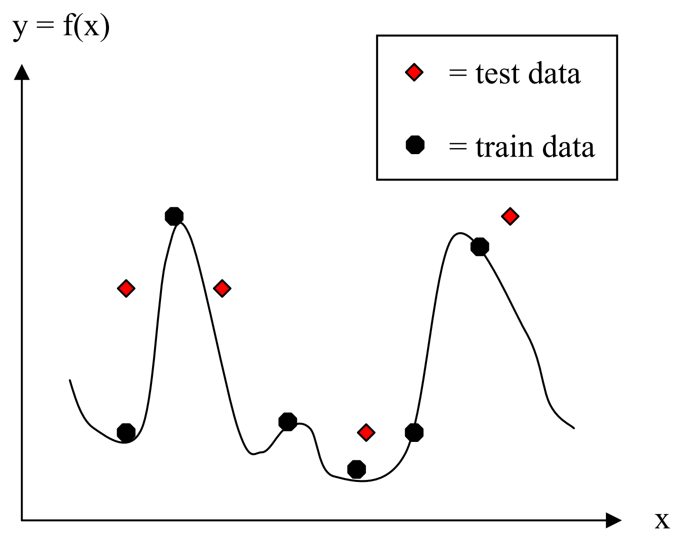

Figure 2.7a An ANN architecture poorly predictive.

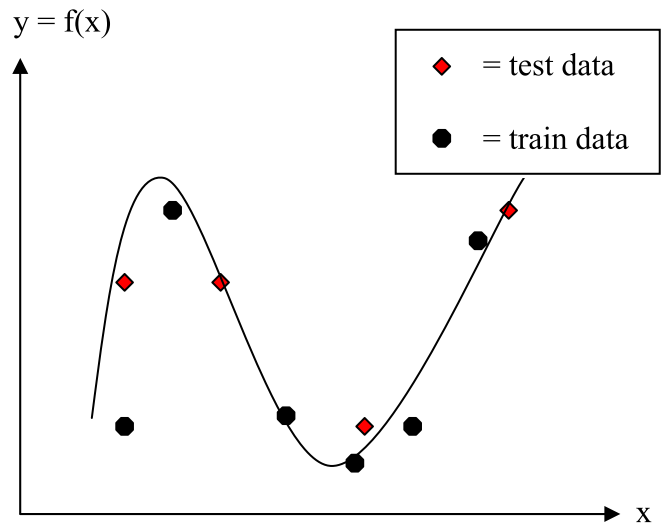

Figure 2.7b. An ANN architecture that have high generalization capabilities.

Although the curve fits almost exactly through the training data in Figure 2.7a, giving almost zero error, the test data are poorly predictive. The ANN architecture has generalized poorly. But in Figure2.7b, a firm ANN architecture has been established. Although the error term is larger for training part, this architecture has high generalization capabilities. If we use more hidden layers, ANN architecture will not be able to understand the underlying trend in the phenomenon. So this case must be avoided.

After the adjustment of network parameters with each epoch, the network is used to find the error for this data set. Initially, errors for both the training and test data sets decrease. After an optimal amount of training has been achieved, the errors for the training set continue to decrease, but those associated with the test data set begin to rise as shown in Figure 2.8. This is an indication that further training will likely result in the network overfitting the training data. The process of training is stopped at this time, and the current set of weights is assumed to be the optimal values. The network is ready for use as a predictive tool. If the available data set is too small for partitioning, the simplest way to prevent overtraining is to stop training when the mean square error stops to decrease significantly.

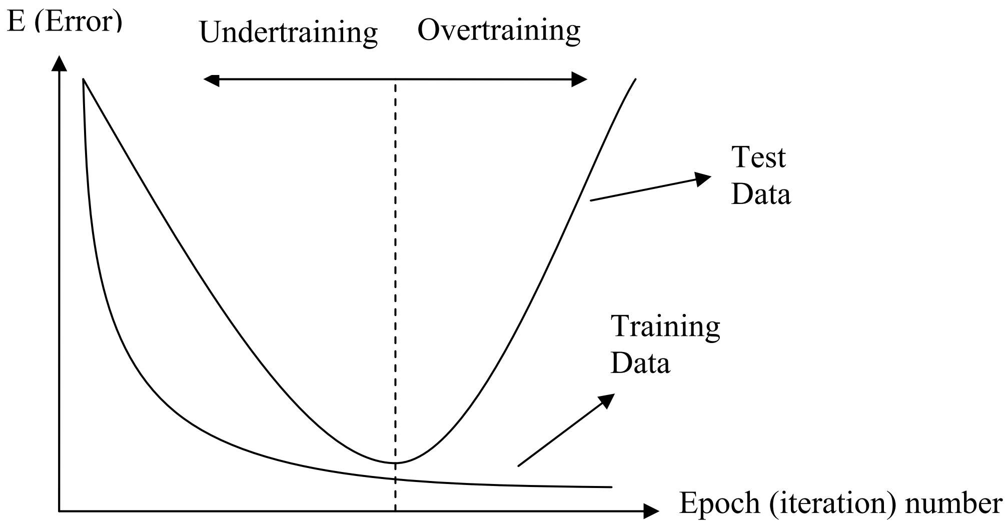

Figure 2.8 Error distributions according to test and training data.

2.6.5 Strength and Limitations

ANNs become popular because of numerous reasons. They can be summarized as follows:

1. ANNs can utilize a relationship between input and output variables without needing explicit mathematical consideration.

2. Even though the training sets include some errors due to the measurements or inadequacy of the measuring devices, they can work well.

3. ANNs can be adapted easily for different circumstances easily.

4. Once key input variables are introduced to the system, ANNs are easy to use and modify for a particular problem.

In general, the mathematical expression obtained using experimental observations are limited by the range of the data. Furthermore, these expressions are also limited by the assumption made in their construction. That makes ANNs seem more attractive. Because, ANNs can be used without having to solve complicated partial differential equations. Furthermore, there is no need to make assumptions between input and output.

The presence of noise in the inputs and outputs is handled by an ANN without severe loss of accuracy because of distributed processing within the network. This, along with the nonlinear nature of the activation function, truly enhances the generalizing capabilities of ANNs and makes them desirable for a large class of problems.

The ANN approach does not require a prior defined functional structure. An ANN adapts its structure in the modeling process to the available data set. A good ANN structure should be able to work generally. Thus it would produce result beyond the available data set which used for model development. Once the structure of an ANN is determined, the ANN becomes a parametric model, thus the number of parameters (weights) is fixed (Jia, 2004).

CHAPTER 3

PROCEDURE AND DISCUSSIONS

3.1 Application of ANN Model by Using Matlab Neural Network Toolbox

In order to apply neural networks to our study, Matlab Neural Network Toolbox is used. Architecture of the network and the parameters selected are given in this section.

First, a new network is created. Secondly, in the *Create New Network window*, the *Network Type* is set to feedforward backpropagation. The input ranges can be set by entering numbers in that field, but it is easier to get them by importing from a specified file.

In this study, the scaled conjugate gradient backpropagation function (*trainscg*), a logsig transfer function and a *learngdm* adaptation learning function are used in the architecture of the ANN. In the Create New Network menu, Transfer function and Learning function are chosen respectively. As default parameter mean square errors (MSE) is used as performance function. The *Create New Network* window, then, looks like as depicted in Figure 3.1.

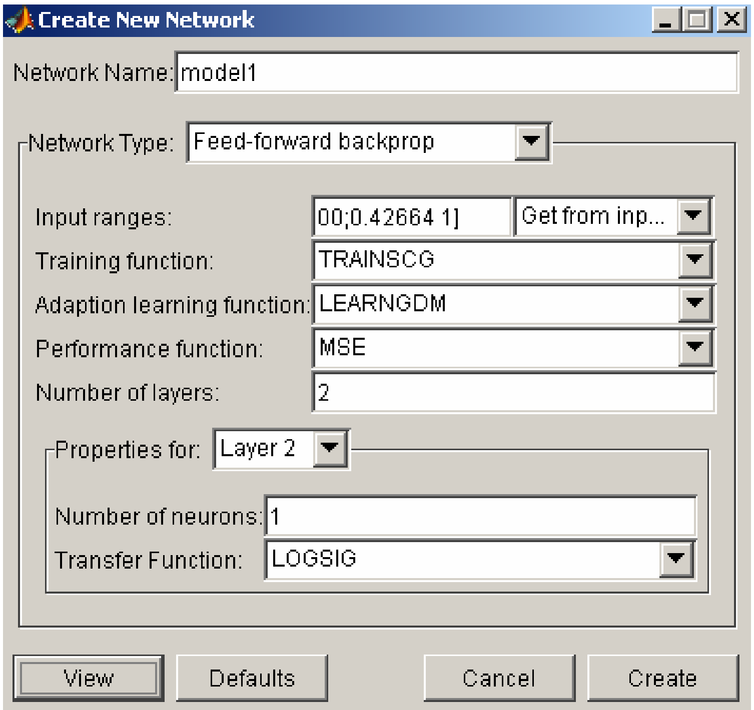

Figure 3.1 Creating New Network

Next, by clicking on the View button the network is viewed as depicted in Figure 3.2. This Figure shows that a network is created such that it consists of an input layer with four elements, a hidden layer with fourteen nodes and an output layer with a single output. Data importing window is illustrated in Figure 3.3.

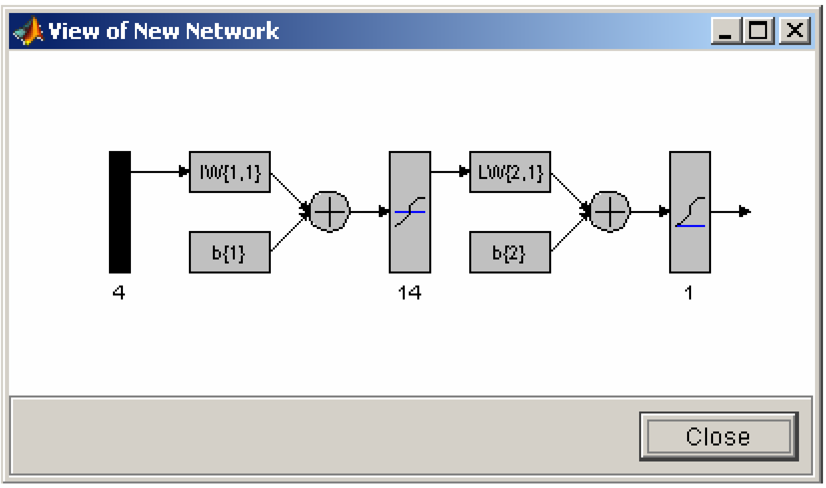

Figure 3.2 View of network

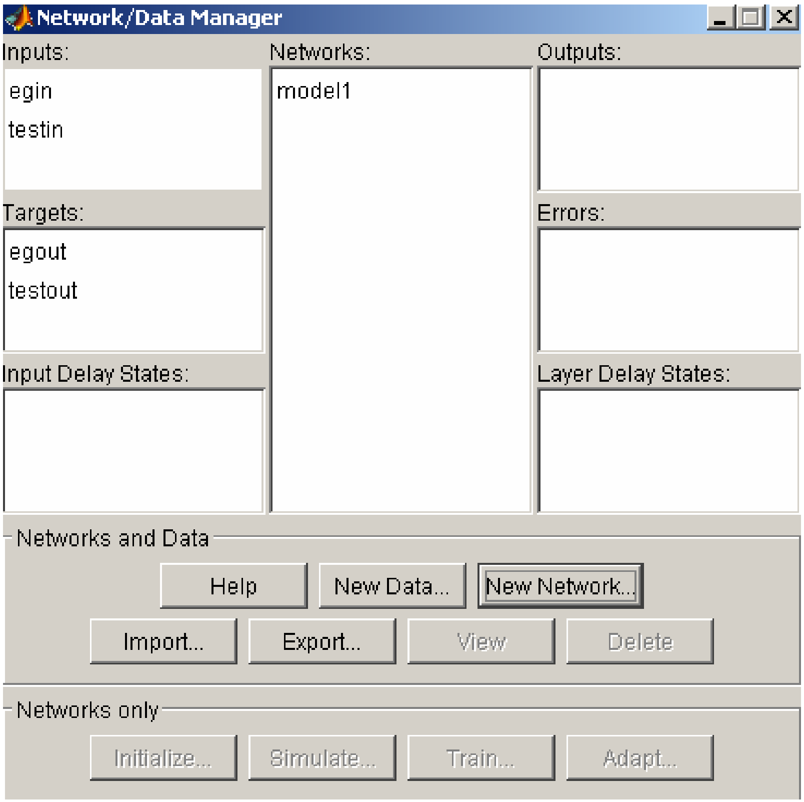

Figure 3.3 Network data manager

To train the network, the train button on *network data manager* window is clicked. First, on this window outputs are specified by clicking on the left tab Training Info and selecting “egin” (training inputs; user defined) from the pop-down list of inputs and “egout” (training inputs) from the pull-down list of targets. *The Network: model1* window looks like as shown in Figure 3.4.

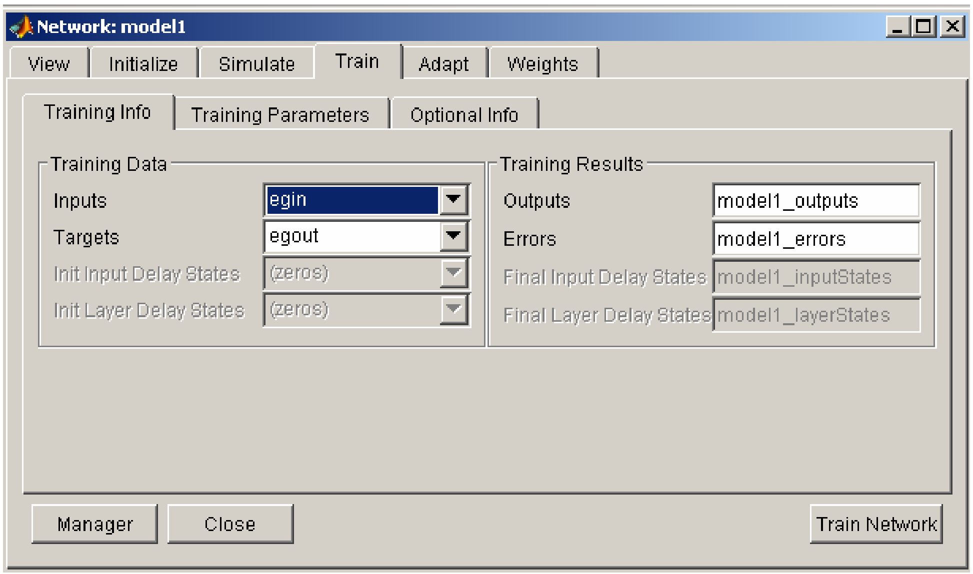

Figure 3.4. Training info

There are eight training parameters associated with *trainscg*: *epochs, show, goal, time, min_grad, max_fail, sigma* and *lambda* as shown in Figure 3.5. The training status is displayed for every iteration (epoch) selected of the algorithm. The other parameters determine the time to terminate. The training stops;

i) if the number of iterations exceeds epochs,

ii) if the performance function drops below *goal*,

iii) if the magnitude of the gradient is less than *min_grad*, or

iv) if the training time is longer than time seconds.

Max_fail is related with the maximum validation failures. Test vectors are used as a further check that the network is generalizing well, but do not have any effect on training. The parameter sigma determines the change in the weight for the second derivative approximation (Mathworks Inc., 2004).

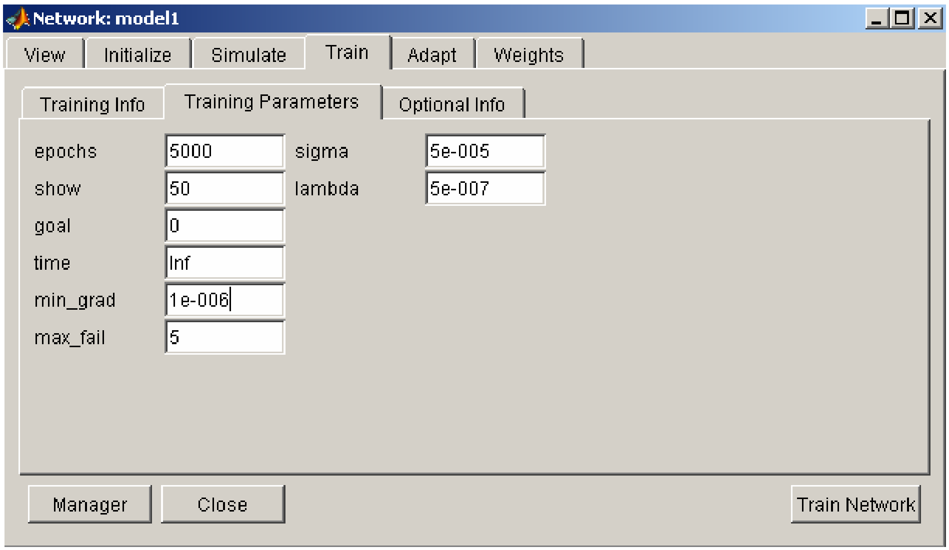

Figure 3.5 Training control parameters.

3.2 Overfitting

One of the problems that occur during neural network training is called overfitting. Overfitting is suggested when the error on the training set is driven to a very small value, while for the test data presented to the network the error is large. That means the network has memorized the training examples, but it has not learned to generalize a new situation. In order to prevent overfitting training data, appreciate epoch number, number of hidden layers and node number of hidden layer must be chosen by trial and error process.

Networks are sensitive to the number of nodes in their hidden layers. Too few nodes can lead to underfitting and too many nodes can result in overfitting. In order to reach an optimum amount of hidden layer nodes, from 1 to 20 nodes are tested. The results are shown in Figure 3.6. Up to 6 nodes there are excessive fluctuations suggesting that number nodes should be within the interval of 6 through 20. Within this range, 14 nodes corresponding to the first smallest difference between training and test MSE values is seemingly the best choice. Although a similar MSE value occurs for the 20th node, for the sake of avoiding additional computational time a hidden layer with 14 nodes is selected.

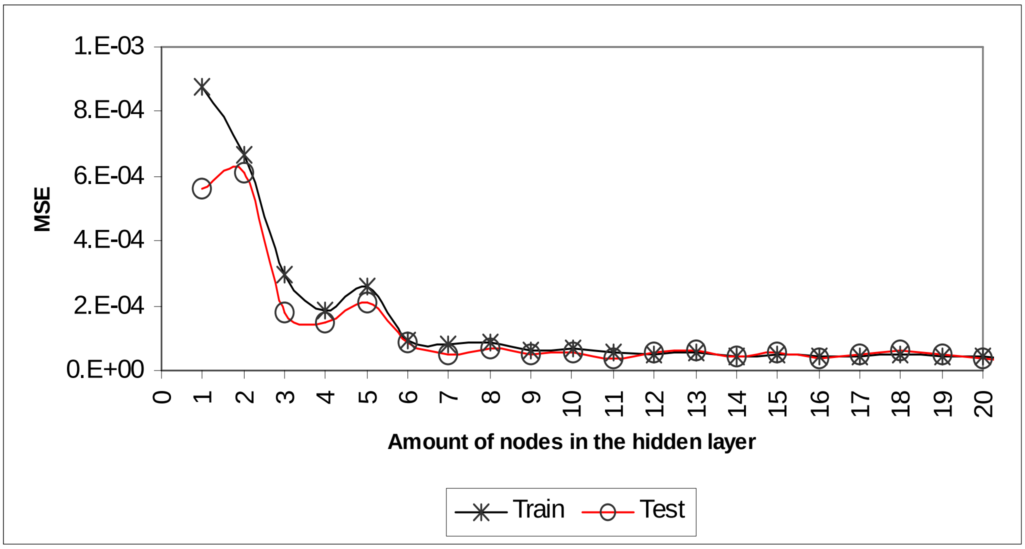

Figure 3.6.Distribution of MSE values of train and test data according to amount of nodes.

Networks also are sensitive to the number of hidden layers. In this study, ANN architectures with only one and two hidden layers are tested. Since, three or more hidden layered systems are known to cause unnecessary computational overload. The variation of MSE for one and two hidden layers obtained are presented in Figure 3.7 a,b. As it is seen, an ANN architecture with one hidden layer turns out to be a more stable design.

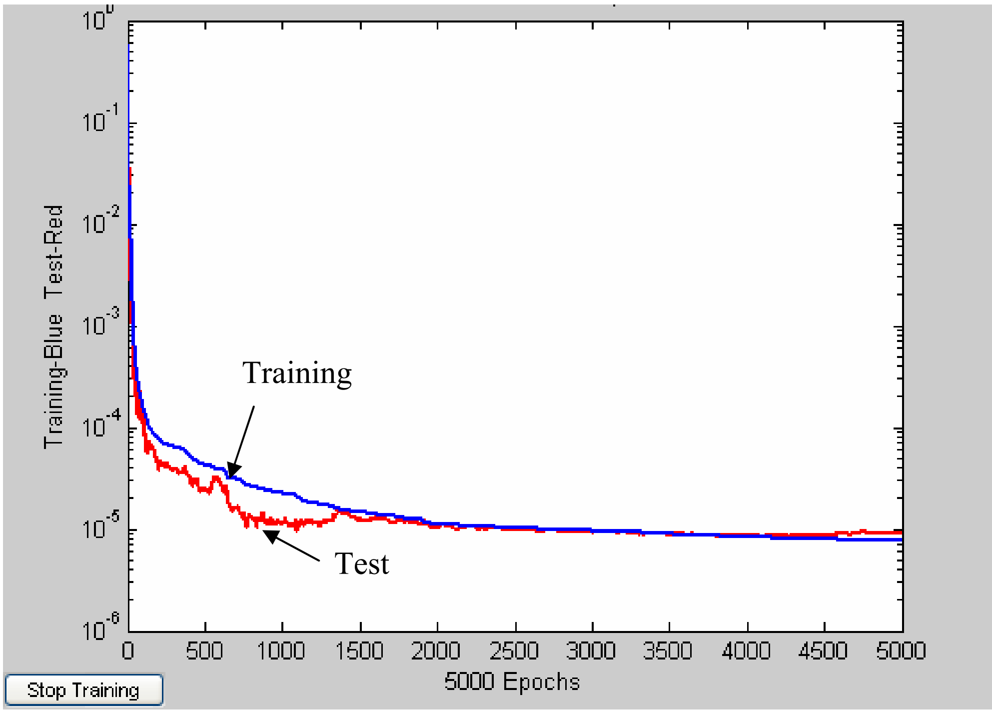

Figure 3.7 a. ANN system with one hidden layer

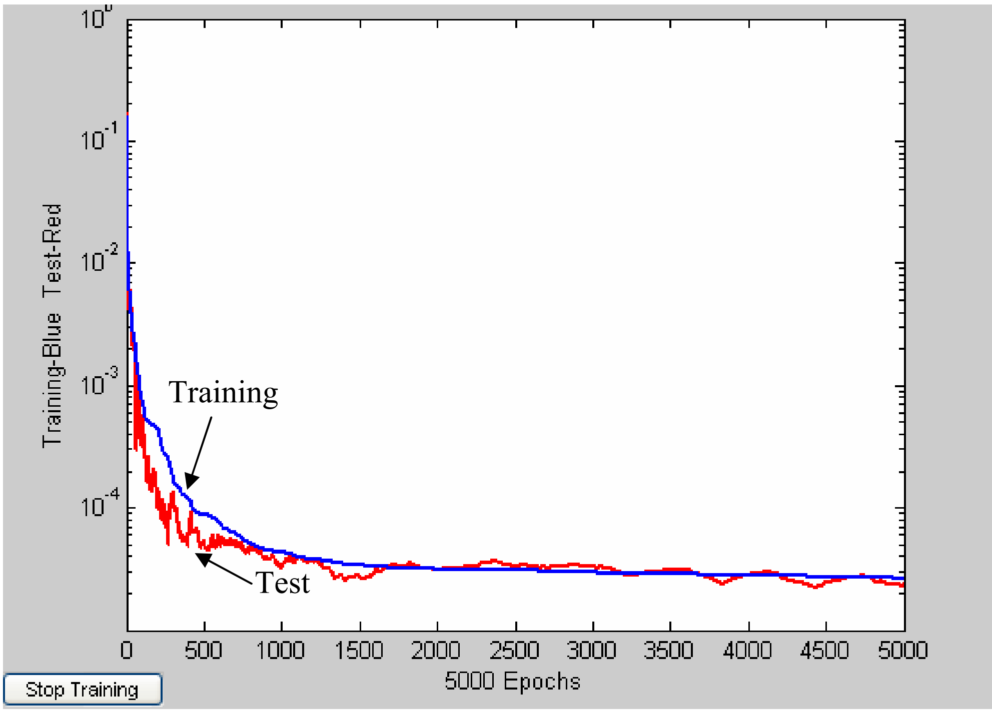

Figure 3.7 b. ANN system with two hidden layers

Thus, for this study an ANN architecture of 5000 epochs with one hidden layer having 14 nodes have been adopted. If the epoch number exceeds 5000, deviations and thus overfitting begins as illustrated in Figure 3.7.a.

3.3 Sensitivity Analysis

In this study, a sensitivity analysis has been conducted to determine the degree of effectiveness of a variable by using performance functions. In this analysis, the data which are given in the appendix are composed of studies of Rajaratnam and Muralidhar (1976), Ferro (1999), Davis et al. (1998), Turan (2002), Fırat (2004), and Kutlu (2005). Parameters in the phenomenon are given in Table 3.1.

Table 3.1. Parameters in the phenomenon

Parameters	Definition
q	Discharge per Unit Depth
y_e	Brink Depth
b	Channel Width
S_o	Channel Bed Slope
n	Manning' s Roughness Coefficient
μ	Viscosity of water
ρ	Density of water
g	The acceleration due to gravity

Unit discharge q ($m^3/s/m$), depends on the other parameters and can be given in this form: $q = f(y_e, y_o, b, S_o, g, \mu, \rho, n)$ (Fırat, 2004).

In terms of availability of the data, properties of the liquid (water) and the gravitational acceleration is constant. Thus, rest of the study effectiveness of brink depth y_e, channel bed slope S_o, channel width b and Manning's roughness coefficient n, are used as the independent parameters over unit discharge to establish an ANN architecture.

Sensitivity analysis correlation coefficients (R^2) of the parameters involved in the phenomenon is given in Figure 3.8 a-d respectively. The most effective parameter is determined as y_e, the brink depth, among the set of the variables that also include S_o, n and b.

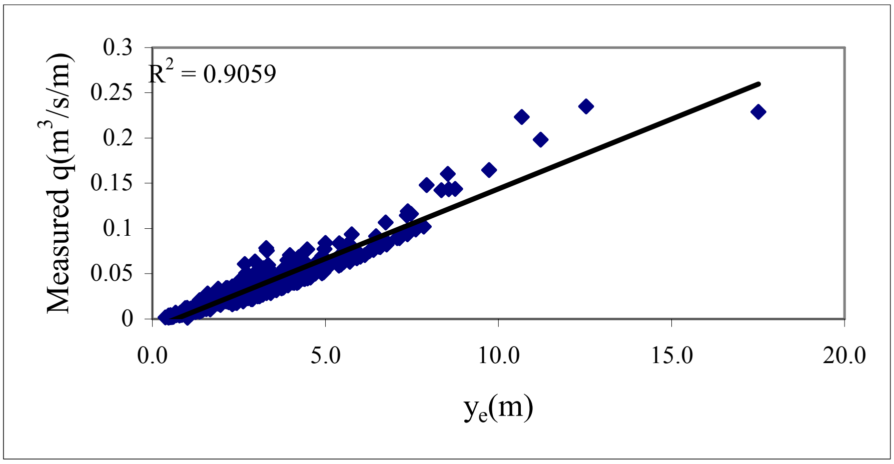

Figure 3.8 a. Variation of q with y_e

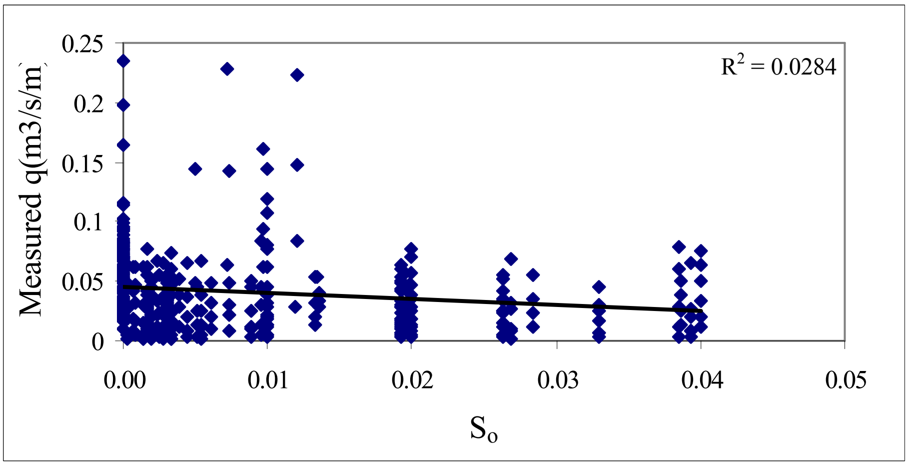

Figure 3.8 b. Variation of q with S_o

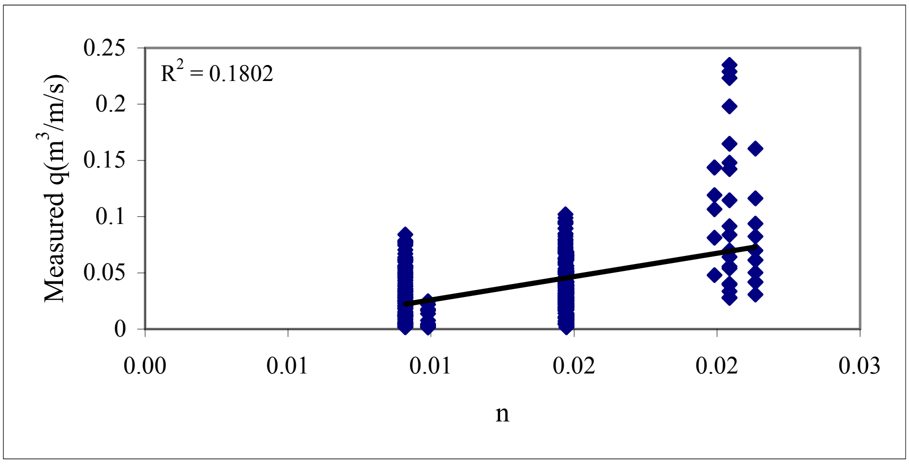

Figure 3.8 c. Variation of q with n

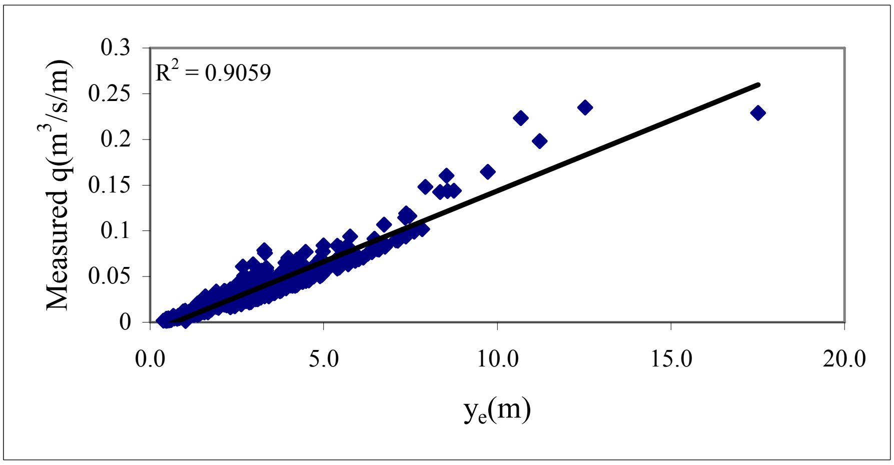

Figure 3.8 d. Variation of q with b

Figure 3.8 a-d. Sensitivity analysis and correlation coefficient (R^2) of the parameters involved in the phenomenon.

Thus, y_e is used as the common parameter for the rest of the sensitivity analysis. Performance evaluation of all possible combination of variables such that each and every combination includes y_e, were also investigated. The findings are listed in Table 3.2. As shown in this table, the MSE value becomes smaller when y_e is used in combination with other variables in a group.

Table 3.2. Performance evaluation of the effective parameters for sensitivity analysis.

Performance	y_e	y_e+b	y_e+S_o	y_e+n	y_e+S_o+n	y_e+S_o+b	y_e+n+b	y_e+b+n+S_o
MSE	0.0011	3.26E-04	1.93E-04	2.32E-04	8.70E-05	1.24E-04	2.36E-04	5.63E-05
R^2	0.941	0.984	0.998	0.990	0.997	0.996	0.990	0.997
TS_{10}(%)	36.364	54.545	72.727	45.455	81.818	72.727	45.455	81.818

Performance of the groups of two and three with the smallest MSE values, that is to say with the best performance values, and the performance of the group that includes all four variables is listed in Table 3.3 and the respective MSE values are depicted in Figure 3.9.

Table 3.3 Best group performances according to number of parameters.

Performance	y_e	y_e+S_o	y_e+S_o+n	y_e+S_o+n+b
MSE	1.10E-03	1.93E-04	8.70E-05	5.63E-05
R^2	0.941	0.998	0.997	0.997
TS_{10}(%)	36.364	72.727	81.818	81.818

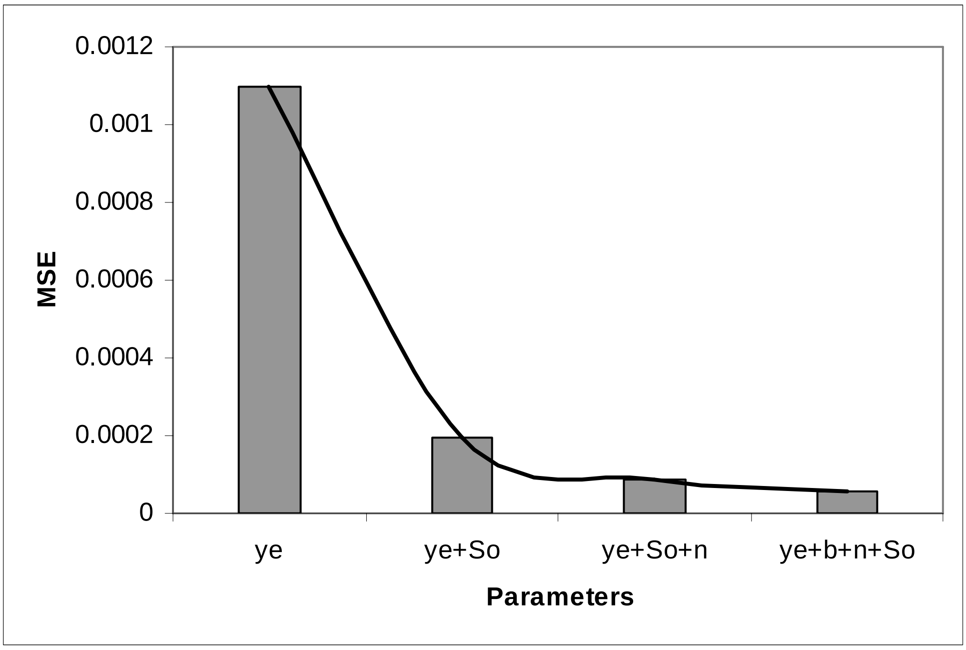

Figure 3.9. MSE performances according to number of parameters.

Based on the findings, as depicted in Figure 3.9, MSE values decreases as the number of variables in the group increases. Furthermore, it can also discerned that the relative increase in the performance due to inclusion of S_o is larger than the contribution of n and b to that effect.

CHAPTER 4

IMPLEMENTATION OF ANN ON FREE OVERFALL

4.1 Implementation of ANN on Free Overfall

Data ranges of researchers used in the study are listed in Table 4.1. All the data used in the study is made available in the Appendix.

Table 4.1 Data ranges of researchers.

Researchers	Slope Range S_o	Roughness Range n	Width Range b (m)	Brink Depth Range y_e(m)	Unit Discharge Range $q(m^3/s/m)$
Kutlu (2005)	0.00063~0.0387	0.014700	1	0.0046~0.0503	0.002~0.061
Fırat (2004)	0.0003~0.0394	0.0091 and 0.0147	1	0.0038~0.0545	0.002~0.084
Turan (2002)	0.0017~0.04	0.009104	1	0.0101~0.0581	0.012~0.078
Ferro (1999)	0	0.014700	0.05~0.3	0.0167~0.0784	0.011~0.102
Rajaratnam and Muralidhar (1976)	0~0.0136	0.0199~0.0213	0.46	0.0183~0.1751	0.028~0.235
Davis et al. (1998)	0.0033~0.02	0.0099 and 0.0147	0.295	0.005~0.0365	0.001~0.046

4.2 Case Study 1

In the implementation of ANN, train groups corresponding to test data are presented below. Test data sets are formed by the randomly chosen 11 data from each researchers' own data cluster. Those data sets are tested twice. In the first step, test data are tested with another researchers' data in training group without any data having their own cluster. In the second step, training group is composed by adding its own cluster excluding test data. Same procedure is followed for the others. Inputs to program, performance values of mean unit discharges (measured, ANN and regression analysis) are presented in Table 4.2.

The Regression analysis outputs mentioned above are obtained by knowing the relation $y_e/y_c = f(\sqrt{S_o}/n)$ from previous studies. Extensive information can be found in Fırat's study (2004). From this relation an experimental equation can be obtained. By replacing y_c by $\sqrt[3]{\frac{q^2}{g}}$, a unit discharge prediction equation is obtained. This procedure is applied in accordance with the implementation of ANN.

The available data clusters which are summarized in Table 4.1 are named after the researcher such that "(K,k)" in for Kutlu, "(E,e)" for Firat, "(T,t)" for Turan, "(F,f)", for Ferro, "(R,r)" for Rajaratnam and Muralidhar and "(A,a)" for Davis et al..

In the above naming convention, upper case implies that the set includes all the data in the respective cluster while lower case implies that only the randomly selected data from the respective cluster is included.

Table 4.2. Performance evaluation of clustered data sets.

Step No	Total Groups		ANN		Regression Analysis	
	Test	Train Group	MSE	R^2	MSE	R^2
1	r	K	1.19E-03	0.8145	1.24E-03	0.9854
2	r	K + (R-r)	5.70E-04	0.9885	1.00E-03	0.9862
3	t	K + R	5.46E-01	0.4029	8.36E-05	0.9371
4	t	K + R + (T-t)	1.52E-04	0.9830	1.24E-04	0.9817
5	a	K + R + T	5.65E-01	0.4868	4.57E-05	0.9602
6	a	K+ R + T + (A-a)	4.24E-04	0.8858	4.99E-05	0.9583
7	e	K + R + T + A	2.04E-04	0.9712	8.36E-05	0.9583
8	e	K + R + T + A+ (E-e)	5.69E-05	0.9925	8.02E-05	0.9676
9	f	K + R + T + A + E	9.98E-03	0.6573	1.13E-03	0.9357
10	f	K + R + T + A + E+ (F-f)	5.76E-05	0.9945	8.29E-04	0.9565

Considering the results as listed in Table 4.2, ANN returns a closer approximation comparing to regression analysis if the same cluster excluding the test data is introduced into the analysis. Yet, if only the train data are used, that is excluding the test data of the same cluster, results becomes unreliable as compared that of regression analysis. This unreliability becomes more apparent for Davis et al.'s (1998) and Turan's (2002) data. For ANN to be able to make robust relations, data distribution of the train group must be regular and close the test group's values. For example, a teacher should not ask a question from a topic that was never explained to the students. If it happens, percentage of answering correctly will not be high. Due to the results observed, ANN's cannot predict its out of range. In other words, ANN cannot extrapolate in long ranges.

4.3 Case Study 2

In the second part, ANN models' ability in establishing relations with different data clusters is studied. As mentioned above, all data clusters used during this study consists of different researchers' data. In this part, one of those six researchers' data cluster was introduced as test and remaining 5 researchers' data clusters were used to compose training data. Using this approach six different models are constructed and the results obtained using ANN and regression analysis are given in Table 4.3 and depicted in Figure 4.1 for companion purposes.

Table 4.3 Result of ANN and regression analysis of each researcher

	ANN		Regression	
	MSE	R^2	MSE	R^2
Davis et al.	1.09E-04	0.835627	6.36E-05	0.950243
Rajaratnam	2.63E-03	0.746602	8.66E-04	0.868204
Ferro	7.64E-04	0.555433	2.76E-05	0.993902
Kutlu	4.73E-06	0.992277	5.89E-05	0.980425
Turan	2.04E-05	0.980816	7.88E-05	0.850099
Fırat	1.01E-05	0.983723	1.15E-04	0.956305

Examining the findings, one may discern that Kutlu’s (2005), Turan’s (2002) and Fırat’s (2004) data clusters turn out a better estimate as compared to the rest of the data clusters. This may be attributed to the fact that they have used the same flume in gathering data. In other words, the majority of the clusters are originated from the same flume inducing a bias towards themselves as far as learning is concerned.

Furthermore the size of the data in the combined clusters of Kutlu (2005), Turan (2002), and Fırat (2004) is larger than that of the rest. This may also induce some unfavorable bias for Davis et al.’s (1998), Ferro’s (1999) and Rajaratnam’s (1976) data during the training.

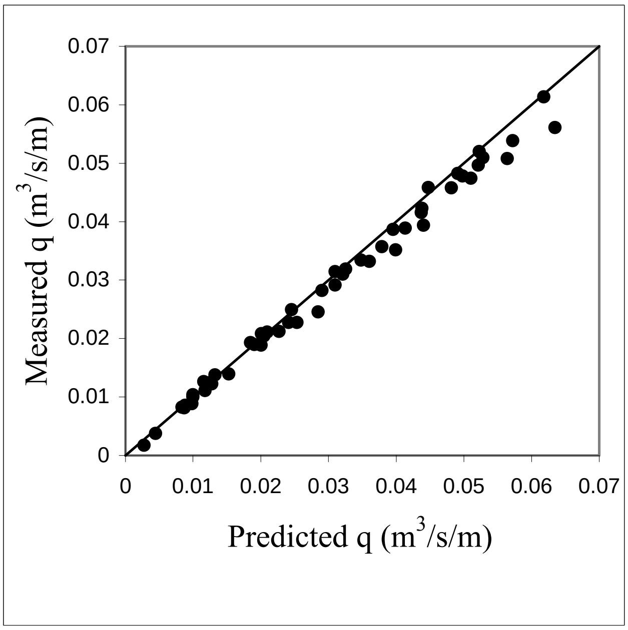

Figure 4.1 a. q_{mean} of ANN versus $q_{measured}$ for Kutlu (2005).

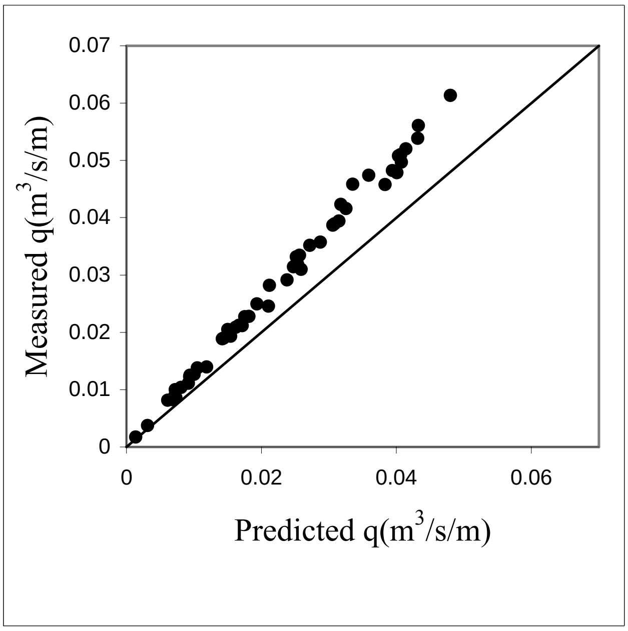

Figure 4.1 b. $q_{predicted}$ of Regression analysis versus $q_{measured}$ for Kutlu (2005)

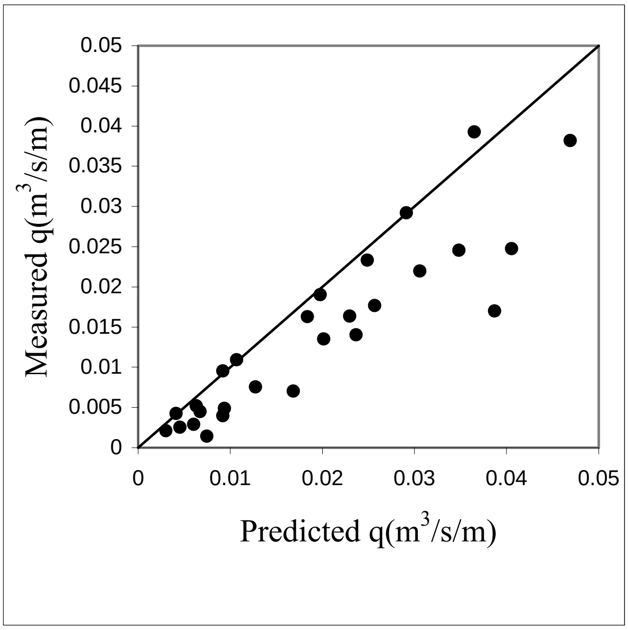

Figure 4.1 c. $q_{predicted}$ of ANN versus $q_{measured}$ for Davis et al. (1998)

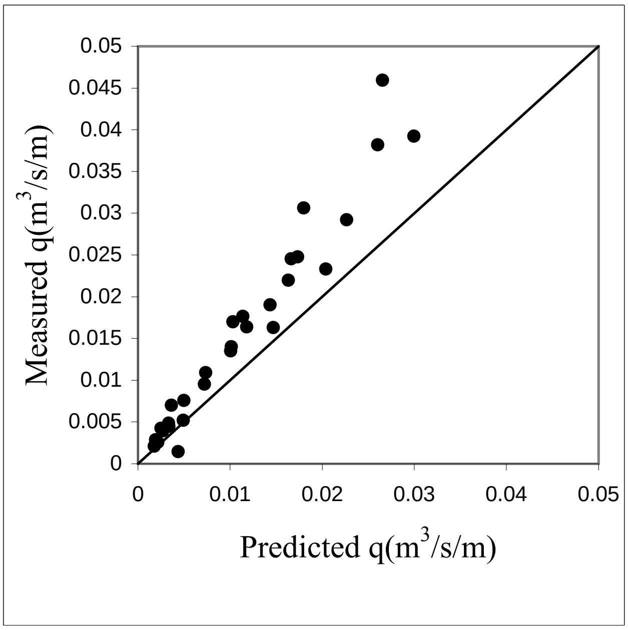

Figure 4.1 d . $q_{predicted}$ of Regression analysis versus $q_{measured}$ for Davis et al. (1998)

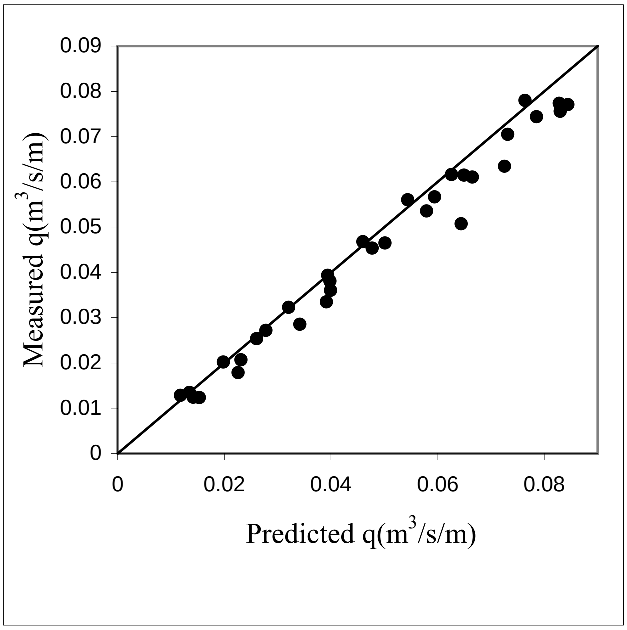

Figure 4.1 e. $q_{predicted}$ of ANN versus $q_{measured}$ for Turan (2002)

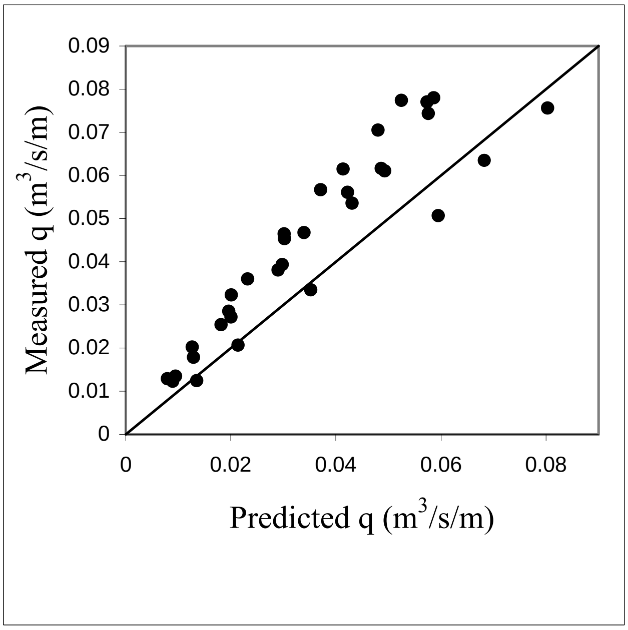

Figure 4.1 f . $q_{predicted}$ of Regression analysis versus $q_{measured}$ for Turan (2002)

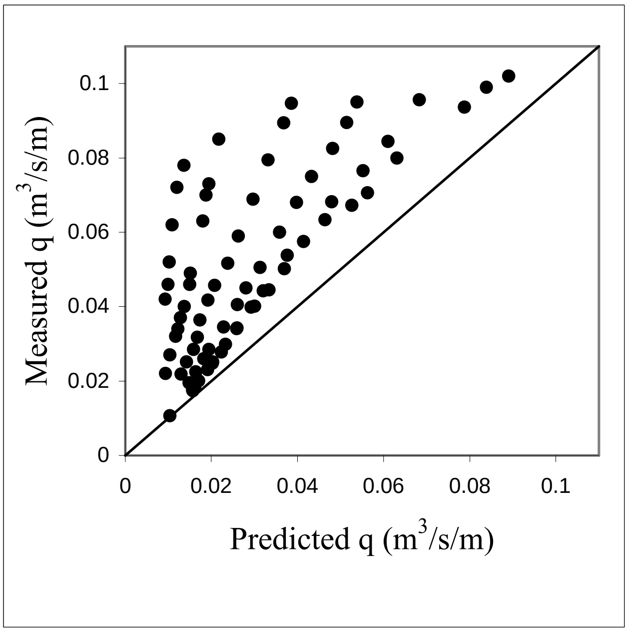

Figure 4.1 g. $q_{predicted}$ of ANN versus $q_{measured}$ for Ferro (1999)

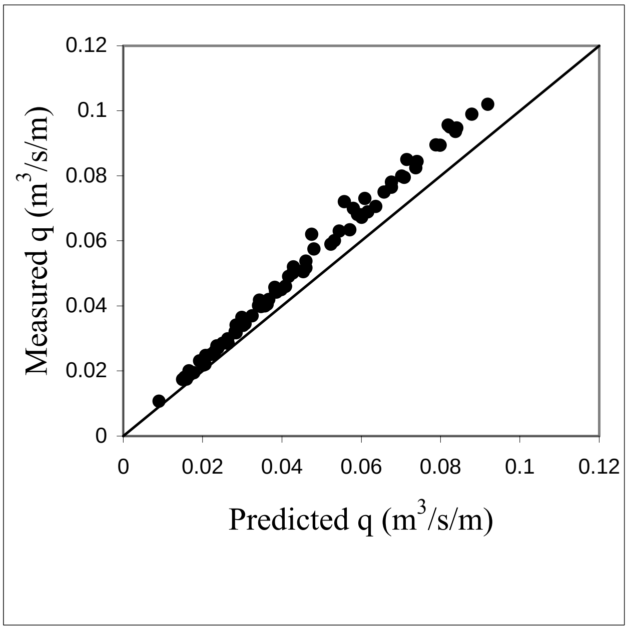

Figure 4.1 h. $q_{predicted}$ of Regression analysis versus $q_{measured}$ for Ferro (1999)

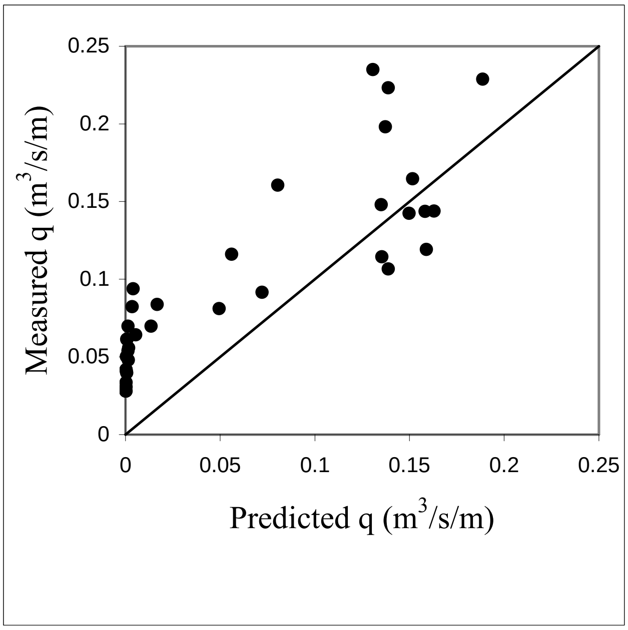

Figure 4.1 i. $q_{predicted}$ of ANN versus $q_{measured}$ for Rajaratnam and Muralidhar (1976)

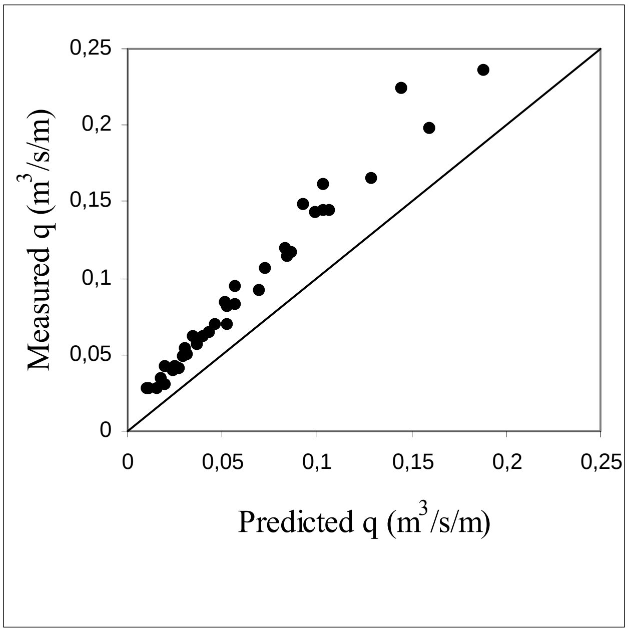

Figure 4.1 j. $q_{predicted}$ of Regression analysis versus $q_{measured}$ for Rajaratnam and Muralidhar (1976)

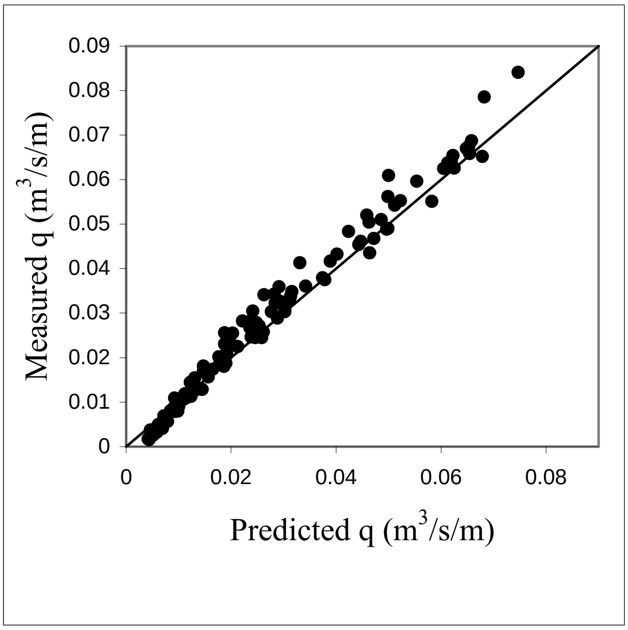

Figure 4.1 k. $q_{predicted}$ of ANN versus $q_{measured}$ for Fırat (2004)

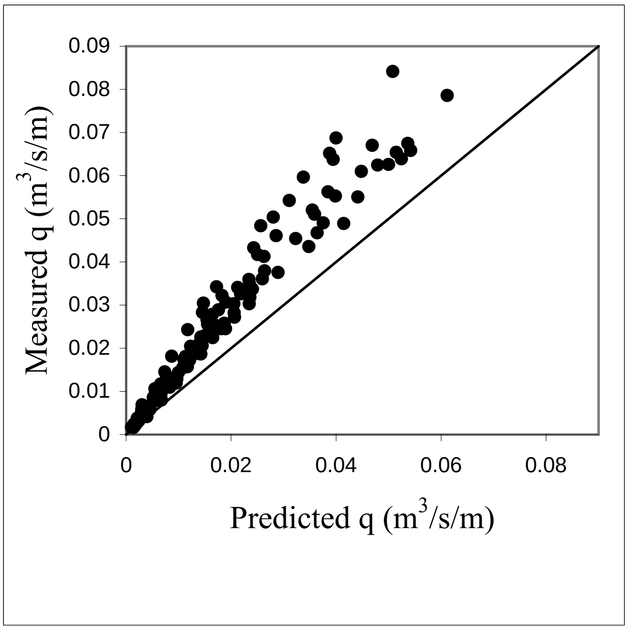

Figure 4.1 l. $q_{predicted}$ of Regression analysis versus $q_{measured}$ for Fırat (2004)

Figure 4.1 a-l Comparison of predicted ANN unit discharges and regression analysis with measured unit discharge values.

CHAPTER 5

CONCLUSION

There are several factors that affect the duration of the training process in ANN development. To obtain a good statistical model of the given problem using multilayer feed-forward networks, the most important thing is to properly define the input and output nodes so that they can adequately reflect the nature of the problem. The next important parameters include the number of hidden layers and nodes and the number of training iterations. In the absence of specific guidelines, obtaining an optimum ANN optimum architecture is a trial error procedure.

In order to construct a robust ANN architecture in this study, after overfitting analysis, 5000 epochs with one hidden layer having 14 nodes are selected. That is, if the epoch number exceeds this value, deviations and thus overfitting begins.

In the present study, sensitivity analysis has been conducted to determine the degree of effectiveness of the variables by using performance functions. According to sensitivity analysis, correlation coefficients (R^2), the most effective parameter is determined as y_e, the brink depth, among the set of variables that also include S_o, slope of the flume, n, Manning's roughness coefficient, and b, the flume width.

It can also be derived from the results that the relative effectiveness in increasing the performance, of the remaining parameters (S_o, n and b), S_o is more effective than n and n is more effective than b.

In the scope of the thesis, there are two case studies. In the first case study, ANN models reliability has been investigated according to the training data clustered and the results are given by comparing to regression analysis. In the second case, ANN models' ability in establishing relations with different data clusters is studied. Based on the findings, it is concluded that, ANNs are highly sensitive to training data group. Furthermore, even though a firm ANN model has been fixed, it is once more confirmed that ANNs must be used with caution for the range outside the based on which they are established.

REFERENCES

Adeli, H., Yeh, C.(1989) : Perceptron Learning in Engineering Design, Microcomp. In Civ. Eng"., 4, pp. 247-256,

Adeli, H., (2001), ''Neural Networks in Civil Engineering: 1989-2000'' *Computer-Aided Civil and Infrastructure Eng.,* 16 (2001) 126-142

Davis, A. C., Ellett, B. G. S., Jacob, R. P., "Flow measurements in sloping channels with rectangular free overfall", J. Hydraulic Eng., ASCE, 1998, Vol. 124, No. 7, July

American Society of Civil Engineers (ASCE) Task Committee on Application of artificial Neural Networks in Hydrology. (2000a). ''Artificial neural networks in hydrology. I: Preliminary concepts.'' *J. Hydrologic Eng.,* 5(2), 115-123.

American Society of Civil Engineers (ASCE) Task Committee on Application of artificial Neural Networks in Hydrology. (2000b). ''Artificial neural networks in hydrology. II: Hydrologic applications.'' *J. Hydrologic Eng.,* 5(2), 124-137.

S. Dey, "Free overfall in inverted semicircular channel", *J. Hydrologic Eng.,* ASCE 129 (6) (2003) 438–447.

Dogan, E., Sasal, M., and Isık S. 2005. Suspended Sediment Load Estimation in Lower Sakarya River by Using Soft Computational Methods, CMMSE 2005 Alicante, Spain, June, 27-30,pp. 395-406

Ferro, V., "Theoretical end-depth-discharge relationship for free overfall", J. Irrigation and Drainage Div., ASCE, 1999, Vol. 125, No. 1

Fırat, C. E., *Effect of Roughness on Flow Measurements in Sloping Rectangular Free Overfall*, *M.s. Thesis*, Middle East Technical University, Ankara, 2004.

Guan, J., *Predicting Neural Networks Behaviors from Basic Operations on a Parallel Computers, M.s. Thesis*, The University of Guelph, Chicago, 2003.

Kutlu, İ., *Scrutinizing Rectangular Free overfall* , *M.s. Thesis*, Middle East Technical University, Ankara, 2005.

Jia, Y., *Robust Optimization for Total Maximum daily load Allocations, Ph.d. Thesis, University of Virginia,* Virginia, 2004

Mathworks Inc., (2004), Matlab 7.0 Neural Network Toolbox, Users Guide Releases Notes, Version 4.0.1, Mathworks Inc.

Merwin, D., A., *A Neural Network Approach to Solving Areal Interpolation Problems, Ph.d. Thesis, University of Connecticut*, Connecticut, 2004.

Özsaraç, D., *Potential flow solution for the Free Overfall*, *M.s. Thesis*, Middle East Technical University, Ankara, November 2001.

Raikar, R.V., Kumar, D. N. and Dey S., “End depth computation in inverted semicircular channels using ANNs”, Flow Measurement and Instrumentation, 15 (2004) 285–293

Rajaratnam, N. and Muralidhar, D., “Roughness Effect of Rectangular Free Overfall”, J. of Hydraulic Division, 1976, Vol. 102 No. HY5.

Rouse, H., “Discharge characteristics of the free overfall”, Civil Engineering, 1936, Vol. 6, No. 4

Rumelhart, D.E., Hinton G.E. & Williams, R.J. *Learning internal representation by error backpropagation,* in Parallel Distributed Processing (D.E.Rumelhart and J.L.McClleland eds.), MIT Press, pp. 318-362, 1986

Sahin, E., Neurocomputing (lecture notes) 2005, METU Computer Engineering Department (CENG 569)

Turan, Ç., *Experimental Studies on Free Overfall*, *M.s. Thesis*, Middle East Technical University, Ankara, 2002.

APPENDIX-A

Table A.1. Researchers' data used in the study

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Kutlu	1	0.00063	1.270	0.01474	0.00814
Kutlu	1	0.00063	2.237	0.01474	0.01888
Kutlu	1	0.00063	3.817	0.01474	0.04230
Kutlu	1	0.00063	5.027	0.01474	0.06136
Kutlu	1	0.00166	1.397	0.01474	0.00888
Kutlu	1	0.00166	1.410	0.01474	0.00998
Kutlu	1	0.00166	2.885	0.01474	0.02821
Kutlu	1	0.00166	3.840	0.01474	0.04157
Kutlu	1	0.00188	0.463	0.01474	0.00175
Kutlu	1	0.00188	2.290	0.01474	0.02046
Kutlu	1	0.00188	3.270	0.01474	0.03342
Kutlu	1	0.00188	3.910	0.01474	0.04581
Kutlu	1	0.00194	1.675	0.01474	0.01228
Kutlu	1	0.00194	2.455	0.01474	0.02120
Kutlu	1	0.00194	3.695	0.01474	0.03890
Kutlu	1	0.00194	4.440	0.01474	0.05100
Kutlu	1	0.00300	1.665	0.01474	0.01248
Kutlu	1	0.00300	2.205	0.01474	0.01897
Kutlu	1	0.00300	3.495	0.01474	0.03573
Kutlu	1	0.00300	4.410	0.01474	0.04968
Kutlu	1	0.00388	1.660	0.01474	0.01240
Kutlu	1	0.00388	3.060	0.01474	0.02916
Kutlu	1	0.00388	3.615	0.01474	0.03866
Kutlu	1	0.00388	4.425	0.01474	0.05200
Kutlu	1	0.00506	0.785	0.01474	0.00376
Kutlu	1	0.00506	1.760	0.01474	0.01381
Kutlu	1	0.00506	2.640	0.01474	0.02492
Kutlu	1	0.00506	4.165	0.01474	0.04578
Kutlu	1	0.00613	1.465	0.01474	0.01041
Kutlu	1	0.00613	2.330	0.01474	0.02086
Kutlu	1	0.00613	3.085	0.01474	0.03143
Kutlu	1	0.00613	4.210	0.01474	0.04824
Kutlu	1	0.00728	1.360	0.01474	0.00861

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Kutlu	1	0.00728	2.400	0.01474	0.02113
Kutlu	1	0.00728	3.155	0.01474	0.03100
Kutlu	1	0.00728	4.220	0.01474	0.04782
Kutlu	1	0.01325	1.610	0.01474	0.01265
Kutlu	1	0.01325	2.140	0.01474	0.01932
Kutlu	1	0.01325	2.980	0.01474	0.03189
Kutlu	1	0.01325	4.245	0.01474	0.05384
Kutlu	1	0.01978	1.180	0.01474	0.00826
Kutlu	1	0.01978	2.225	0.01474	0.02274
Kutlu	1	0.01978	2.830	0.01474	0.03317
Kutlu	1	0.01978	3.585	0.01474	0.04741
Kutlu	1	0.02838	1.360	0.01474	0.01112
Kutlu	1	0.02838	2.135	0.01474	0.02275
Kutlu	1	0.02838	2.795	0.01474	0.03517
Kutlu	1	0.02838	3.810	0.01474	0.05612
Kutlu	1	0.03869	1.490	0.01474	0.01395
Kutlu	1	0.03869	2.180	0.01474	0.02456
Kutlu	1	0.03869	2.850	0.01474	0.03938
Kutlu	1	0.03869	3.360	0.01474	0.05081
Fırat	1	0.00280	3.590	0.01474	0.03753
Fırat	1	0.00280	2.707	0.01474	0.02450
Fırat	1	0.00280	0.470	0.01474	0.00170
Fırat	1	0.00280	5.450	0.01474	0.06585
Fırat	1	0.00280	1.567	0.01474	0.01095
Fırat	1	0.00280	1.037	0.01474	0.00576
Fırat	1	0.00450	1.370	0.01474	0.00798
Fırat	1	0.00450	5.287	0.01474	0.06543
Fırat	1	0.00450	0.607	0.01474	0.00264
Fırat	1	0.00450	3.350	0.01474	0.03608
Fırat	1	0.00450	1.183	0.01474	0.00804
Fırat	1	0.00450	2.277	0.01474	0.02060
Fırat	1	0.01930	1.847	0.01474	0.01740
Fırat	1	0.01930	4.320	0.01474	0.06376
Fırat	1	0.01930	0.590	0.01474	0.00262
Fırat	1	0.01930	3.193	0.01474	0.04168
Fırat	1	0.01930	1.510	0.01474	0.01311
Fırat	1	0.01930	2.413	0.01474	0.02784

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Fırat	1	0.03940	1.817	0.01474	0.02047
Fırat	1	0.03940	0.617	0.01474	0.00321
Fırat	1	0.03940	1.083	0.01474	0.00866
Fırat	1	0.03940	3.903	0.01474	0.06519
Fırat	1	0.03940	2.157	0.01474	0.02710
Fırat	1	0.03940	1.022	0.01474	0.00860
Fırat	1	0.00880	3.880	0.01474	0.04542
Fırat	1	0.00880	1.420	0.01474	0.01057
Fırat	1	0.00880	4.163	0.01474	0.05101
Fırat	1	0.00880	0.603	0.01474	0.00289
Fırat	1	0.00880	2.697	0.01474	0.02578
Fırat	1	0.00880	1.603	0.01474	0.01247
Fırat	1	0.00230	2.460	0.01474	0.02246
Fırat	1	0.00230	0.813	0.01474	0.00455
Fırat	1	0.00230	1.043	0.01474	0.00661
Fırat	1	0.00230	5.397	0.01474	0.06746
Fırat	1	0.00230	1.547	0.01474	0.01185
Fırat	1	0.00230	3.163	0.01474	0.03373
Fırat	1	0.00080	1.910	0.01474	0.01696
Fırat	1	0.00080	0.860	0.01474	0.00476
Fırat	1	0.00080	4.940	0.01474	0.06248
Fırat	1	0.00080	1.977	0.01474	0.01724
Fırat	1	0.00080	4.110	0.01474	0.04673
Fırat	1	0.00080	3.080	0.01474	0.03175
Fırat	1	0.02690	1.203	0.01474	0.01014
Fırat	1	0.02690	4.233	0.01474	0.06872
Fırat	1	0.02690	0.480	0.01474	0.00246
Fırat	1	0.02690	2.513	0.01474	0.03223
Fırat	1	0.02690	1.093	0.01474	0.00825
Fırat	1	0.02690	2.243	0.01474	0.02669
Fırat	1	0.03850	0.957	0.00910	0.01195
Fırat	1	0.03850	1.597	0.00910	0.02824
Fırat	1	0.03850	2.677	0.00910	0.06094
Fırat	1	0.03850	0.530	0.00910	0.00412
Fırat	1	0.03850	3.293	0.00910	0.07860
Fırat	1	0.02630	2.170	0.00910	0.03410
Fırat	1	0.02630	1.380	0.00910	0.01544

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Fırat	1	0.02630	2.310	0.00910	0.03594
Fırat	1	0.02630	1.790	0.00910	0.02468
Fırat	1	0.02630	1.850	0.00910	0.02550
Fırat	1	0.02630	0.690	0.00910	0.00481
Fırat	1	0.02630	1.760	0.00910	0.02305
Fırat	1	0.02630	1.050	0.00910	0.01094
Fırat	1	0.02630	2.495	0.00910	0.04129
Fırat	1	0.02630	3.220	0.00910	0.05620
Fırat	1	0.02630	3.050	0.00910	0.05197
Fırat	1	0.02630	1.760	0.00910	0.02555
Fırat	1	0.02630	0.470	0.00910	0.00377
Fırat	1	0.02630	1.310	0.00910	0.01440
Fırat	1	0.02630	0.700	0.00910	0.00496
Fırat	1	0.01920	2.167	0.00910	0.02891
Fırat	1	0.01920	1.617	0.00910	0.01805
Fırat	1	0.01920	1.357	0.00910	0.01812
Fırat	1	0.01920	2.787	0.00910	0.04834
Fırat	1	0.01920	3.347	0.00910	0.05963
Fırat	1	0.01920	2.957	0.00910	0.05039
Fırat	1	0.01920	1.927	0.00910	0.03044
Fırat	1	0.01920	1.217	0.00910	0.01453
Fırat	1	0.01920	0.667	0.00910	0.00588
Fırat	1	0.01920	2.137	0.00910	0.03425
Fırat	1	0.01920	1.657	0.00910	0.02428
Fırat	1	0.01920	0.677	0.00910	0.00690
Fırat	1	0.01920	0.997	0.00910	0.01056
Fırat	1	0.01920	1.913	0.00910	0.02832
Fırat	1	0.01920	3.167	0.00910	0.05421
Fırat	1	0.01920	0.717	0.00910	0.00544
Fırat	1	0.01920	2.690	0.00910	0.04323
Fırat	1	0.01920	1.123	0.00910	0.01171
Fırat	1	0.00960	2.600	0.00910	0.03066
Fırat	1	0.00960	5.000	0.00910	0.08412
Fırat	1	0.00960	1.410	0.00910	0.01274
Fırat	1	0.00960	3.410	0.00910	0.04610
Fırat	1	0.00960	0.850	0.00910	0.00571
Fırat	1	0.00540	3.353	0.00910	0.03794

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Fırat	1	0.00540	0.383	0.00910	0.00170
Fırat	1	0.00540	4.920	0.00910	0.06705
Fırat	1	0.00540	2.610	0.00910	0.02448
Fırat	1	0.00540	1.723	0.00910	0.01286
Fırat	1	0.00540	1.513	0.00910	0.01125
Fırat	1	0.00540	0.760	0.00910	0.00483
Fırat	1	0.00300	1.600	0.00910	0.01185
Fırat	1	0.00300	2.100	0.00910	0.02020
Fırat	1	0.00300	4.080	0.00910	0.04352
Fırat	1	0.00300	4.290	0.00910	0.04901
Fırat	1	0.00300	2.590	0.00910	0.02463
Fırat	1	0.00300	1.250	0.00910	0.00792
Fırat	1	0.00300	3.110	0.00910	0.03296
Fırat	1	0.00300	2.220	0.00910	0.02106
Fırat	1	0.00300	2.870	0.00910	0.03031
Fırat	1	0.00300	2.250	0.00910	0.02260
Fırat	1	0.00300	2.210	0.00910	0.01875
Fırat	1	0.00300	2.990	0.00910	0.03261
Fırat	1	0.00300	4.780	0.00910	0.05506
Fırat	1	0.00300	3.140	0.00910	0.03482
Fırat	1	0.00250	2.330	0.00910	0.02287
Fırat	1	0.00250	1.350	0.00910	0.00908
Fırat	1	0.00250	1.300	0.00910	0.00894
Fırat	1	0.00250	0.603	0.00910	0.00294
Fırat	1	0.00250	4.470	0.00910	0.05524
Fırat	1	0.00140	3.130	0.00910	0.03029
Fırat	1	0.00140	0.927	0.00910	0.00514
Fırat	1	0.00140	5.177	0.00910	0.06258
Fırat	1	0.00140	0.483	0.00910	0.00161
Fırat	1	0.00140	1.970	0.00910	0.01567
Fırat	1	0.00140	0.517	0.00910	0.00206
Fırat	1	0.00140	0.580	0.00910	0.00275
Fırat	1	0.00030	2.200	0.00910	0.01865
Fırat	1	0.00030	0.517	0.00910	0.00198
Fırat	1	0.00030	5.243	0.00910	0.06390
Fırat	1	0.00030	4.483	0.00910	0.04887
Fırat	1	0.00030	2.820	0.00910	0.02718

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Fırat	1	0.00030	0.840	0.00910	0.00448
Turan	1	0.04000	1.010	0.00910	0.01245
Turan	1	0.04000	1.370	0.00910	0.02070
Turan	1	0.04000	1.910	0.00910	0.03348
Turan	1	0.04000	2.710	0.00910	0.05069
Turan	1	0.04000	2.970	0.00910	0.06345
Turan	1	0.04000	3.310	0.00910	0.07560
Turan	1	0.02000	1.300	0.00910	0.01233
Turan	1	0.02000	1.660	0.00910	0.01787
Turan	1	0.02000	2.190	0.00910	0.02852
Turan	1	0.02000	2.450	0.00910	0.03601
Turan	1	0.02000	2.920	0.00910	0.04647
Turan	1	0.02000	3.350	0.00910	0.05666
Turan	1	0.02000	3.980	0.00910	0.07053
Turan	1	0.02000	4.480	0.00910	0.07704
Turan	1	0.01000	1.410	0.00910	0.01286
Turan	1	0.01000	1.930	0.00910	0.02020
Turan	1	0.01000	2.630	0.00910	0.03226
Turan	1	0.01000	3.450	0.00910	0.04533
Turan	1	0.01000	4.250	0.00910	0.06145
Turan	1	0.01000	4.980	0.00910	0.07738
Turan	1	0.00333	2.640	0.00910	0.02537
Turan	1	0.00333	3.610	0.00910	0.03810
Turan	1	0.00333	4.010	0.00910	0.04676
Turan	1	0.00333	4.700	0.00910	0.05357
Turan	1	0.00333	5.140	0.00910	0.06102
Turan	1	0.00333	5.700	0.00910	0.07441
Turan	1	0.00167	1.730	0.00910	0.01346
Turan	1	0.00167	2.840	0.00910	0.02718
Turan	1	0.00167	3.700	0.00910	0.03935
Turan	1	0.00167	4.670	0.00910	0.05605
Turan	1	0.00167	5.130	0.00910	0.06162
Turan	1	0.00167	5.810	0.00910	0.07797
Rajaratnam	0.46	0.00500	3.6905	0.01992	0.04800
Rajaratnam	0.46	0.00500	8.7535	0.01992	0.14382
Rajaratnam	0.46	0.01000	8.5705	0.01992	0.14363
Rajaratnam	0.46	0.00999	7.381	0.01992	0.11907

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Rajaratnam	0.46	0.00997	5.49	0.01992	0.08112
Rajaratnam	0.46	0.00994	6.7405	0.01992	0.10661
Rajaratnam	0.46	0.00000	9.7295	0.02044	0.16465
Rajaratnam	0.46	0.00000	6.466	0.02044	0.09154
Rajaratnam	0.46	0.00000	11.224	0.02044	0.19814
Rajaratnam	0.46	0.00000	4.2395	0.02044	0.05582
Rajaratnam	0.46	0.00000	2.4095	0.02044	0.02791
Rajaratnam	0.46	0.00000	12.5355	0.02044	0.23498
Rajaratnam	0.46	0.00000	7.3505	0.02044	0.11442
Rajaratnam	0.46	0.00000	5.3985	0.02044	0.06977
Rajaratnam	0.46	0.00000	3.477	0.02044	0.04047
Rajaratnam	0.46	0.00727	4.7885	0.02044	0.06419
Rajaratnam	0.46	0.00727	8.357	0.02044	0.14233
Rajaratnam	0.46	0.00724	17.507	0.02044	0.22884
Rajaratnam	0.46	0.01209	10.675	0.02044	0.22326
Rajaratnam	0.46	0.01209	5.3985	0.02044	0.08372
Rajaratnam	0.46	0.01211	7.93	0.02044	0.14791
Rajaratnam	0.46	0.01193	1.83	0.02044	0.02791
Rajaratnam	0.46	0.01358	3.233	0.02044	0.03963
Rajaratnam	0.46	0.01354	2.6535	0.02044	0.03349
Rajaratnam	0.46	0.01357	1.9825	0.02044	0.02791
Rajaratnam	0.46	0.01347	3.8125	0.02044	0.05386
Rajaratnam	0.46	0.00000	7.4725	0.02134	0.11610
Rajaratnam	0.46	0.00000	4.9105	0.02134	0.06977
Rajaratnam	0.46	0.00000	3.3245	0.02134	0.04186
Rajaratnam	0.46	0.00000	4.453	0.02134	0.06140
Rajaratnam	0.46	0.00000	2.7755	0.02134	0.03070
Rajaratnam	0.46	0.00000	3.8125	0.02134	0.05023
Rajaratnam	0.46	0.00000	5.6425	0.02134	0.08233
Rajaratnam	0.46	0.00975	8.54	0.02134	0.16047
Rajaratnam	0.46	0.00971	5.7645	0.02134	0.09377
Rajaratnam	0.46	0.00970	4.1175	0.02134	0.06140
Rajaratnam	0.46	0.00966	2.8975	0.02134	0.04186
Davis et al.	0.305	0.03300	0.530233	0.00990	0.00396
Davis et al.	0.305	0.03300	1.823256	0.00990	0.02476
Davis et al.	0.305	0.03300	0.813953	0.01470	0.00702
Davis et al.	0.305	0.03300	1.627907	0.01470	0.01699

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Davis et al.	0.305	0.03300	2.35814	0.01470	0.03062
Davis et al.	0.305	0.03300	3.055814	0.01470	0.04593
Davis et al.	0.305	0.02000	0.497674	0.00990	0.00288
Davis et al.	0.305	0.02000	0.948837	0.00990	0.00756
Davis et al.	0.305	0.02000	1.64186	0.00990	0.01767
Davis et al.	0.305	0.02000	0.813953	0.01470	0.00487
Davis et al.	0.305	0.02000	1.706977	0.01470	0.01402
Davis et al.	0.305	0.02000	2.376744	0.01470	0.02456
Davis et al.	0.305	0.02000	3.2	0.01470	0.03819
Davis et al.	0.305	0.01000	0.604651	0.00990	0.00256
Davis et al.	0.305	0.01000	0.813953	0.00990	0.00448
Davis et al.	0.305	0.01000	1.683721	0.00990	0.01353
Davis et al.	0.305	0.01000	2.325581	0.00990	0.02197
Davis et al.	0.305	0.01000	0.697674	0.01470	0.00426
Davis et al.	0.305	0.01000	1.432558	0.01470	0.01092
Davis et al.	0.305	0.01000	2.237209	0.01470	0.01903
Davis et al.	0.305	0.01000	3.032558	0.01470	0.02919
Davis et al.	0.305	0.01000	3.651163	0.01470	0.03925
Davis et al.	0.305	0.00330	0.56092	0.00990	0.00212
Davis et al.	0.305	0.00330	1.02069	0.00990	0.00144
Davis et al.	0.305	0.00330	1.981609	0.00990	0.01639
Davis et al.	0.305	0.00330	1.117241	0.01470	0.00521
Davis et al.	0.305	0.00330	1.43908	0.01470	0.00955
Davis et al.	0.305	0.00330	2.312644	0.01470	0.01631
Davis et al.	0.305	0.00330	2.873563	0.01470	0.02331
Ferro	0.299	0.00000	1.67	0.01470	0.01070
Ferro	0.299	0.00000	2.34	0.01470	0.01739
Ferro	0.299	0.00000	2.4	0.01470	0.01806
Ferro	0.299	0.00000	2.51	0.01470	0.02007
Ferro	0.299	0.00000	2.77	0.01470	0.02308
Ferro	0.299	0.00000	2.91	0.01470	0.02475
Ferro	0.299	0.00000	3.17	0.01470	0.02776
Ferro	0.299	0.00000	3.59	0.01470	0.03411
Ferro	0.299	0.00000	4.05	0.01470	0.04013
Ferro	0.299	0.00000	4.38	0.01470	0.04448
Ferro	0.299	0.00000	4.71	0.01470	0.05017
Ferro	0.299	0.00000	5.09	0.01470	0.05753

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Ferro	0.299	0.00000	5.91	0.01470	0.06722
Ferro	0.299	0.00000	6.14	0.01470	0.07057
Ferro	0.299	0.00000	6.55	0.01470	0.07993
Ferro	0.299	0.00000	7.37	0.01470	0.09365
Ferro	0.299	0.00000	7.61	0.01470	0.09900
Ferro	0.299	0.00000	7.84	0.01470	0.10201
Ferro	0.251	0.00000	2.44	0.01470	0.01753
Ferro	0.251	0.00000	3.04	0.01470	0.02510
Ferro	0.251	0.00000	3.41	0.01470	0.02988
Ferro	0.251	0.00000	3.73	0.01470	0.03426
Ferro	0.251	0.00000	4.1	0.01470	0.03984
Ferro	0.251	0.00000	4.4	0.01470	0.04422
Ferro	0.251	0.00000	4.95	0.01470	0.05378
Ferro	0.251	0.00000	5.71	0.01470	0.06335
Ferro	0.251	0.00000	5.84	0.01470	0.06813
Ferro	0.251	0.00000	6.39	0.01470	0.07649
Ferro	0.251	0.00000	6.79	0.01470	0.08446
Ferro	0.251	0.00000	7.26	0.01470	0.09562
Ferro	0.2	0.00000	2.63	0.01470	0.01950
Ferro	0.2	0.00000	2.86	0.01470	0.02250
Ferro	0.2	0.00000	3.13	0.01470	0.02600
Ferro	0.2	0.00000	3.3	0.01470	0.02850
Ferro	0.2	0.00000	3.78	0.01470	0.03450
Ferro	0.2	0.00000	4.22	0.01470	0.04050
Ferro	0.2	0.00000	4.49	0.01470	0.04500
Ferro	0.2	0.00000	4.9	0.01470	0.05050
Ferro	0.2	0.00000	5.45	0.01470	0.06000
Ferro	0.2	0.00000	5.89	0.01470	0.06800
Ferro	0.2	0.00000	6.27	0.01470	0.07500
Ferro	0.2	0.00000	6.77	0.01470	0.08250
Ferro	0.2	0.00000	7.08	0.01470	0.08950
Ferro	0.2	0.00000	7.29	0.01470	0.09500
Ferro	0.151	0.00000	2.86	0.01470	0.02185
Ferro	0.151	0.00000	3.1	0.01470	0.02517
Ferro	0.151	0.00000	3.42	0.01470	0.02848
Ferro	0.151	0.00000	3.59	0.01470	0.03179
Ferro	0.151	0.00000	3.71	0.01470	0.03642

Table A.1 (cont'd)

Researcher	b(m)	S_o(m/m)	y_e(cm)	n	q(m^3/s/m)
Ferro	0.151	0.00000	4.07	0.01470	0.04172
Ferro	0.151	0.00000	4.37	0.01470	0.04570
Ferro	0.151	0.00000	4.95	0.01470	0.05166
Ferro	0.151	0.00000	5.39	0.01470	0.05894
Ferro	0.151	0.00000	6.01	0.01470	0.06887
Ferro	0.151	0.00000	6.59	0.01470	0.07947
Ferro	0.151	0.00000	7.14	0.01470	0.08940
Ferro	0.151	0.00000	7.39	0.01470	0.09470
Ferro	0.1	0.00000	2.9	0.01470	0.02200
Ferro	0.1	0.00000	3.19	0.01470	0.02700
Ferro	0.1	0.00000	3.57	0.01470	0.03200
Ferro	0.1	0.00000	3.74	0.01470	0.03400
Ferro	0.1	0.00000	3.92	0.01470	0.03700
Ferro	0.1	0.00000	4.18	0.01470	0.04000
Ferro	0.1	0.00000	4.57	0.01470	0.04600
Ferro	0.1	0.00000	4.63	0.01470	0.04900
Ferro	0.1	0.00000	5.53	0.01470	0.06300
Ferro	0.1	0.00000	5.77	0.01470	0.07000
Ferro	0.1	0.00000	5.96	0.01470	0.07300
Ferro	0.1	0.00000	6.63	0.01470	0.08500
Ferro	0.05	0.00000	4.25	0.01470	0.04200
Ferro	0.05	0.00000	4.57	0.01470	0.04600
Ferro	0.05	0.00000	4.72	0.01470	0.05200
Ferro	0.05	0.00000	5.05	0.01470	0.06200
Ferro	0.05	0.00000	5.62	0.01470	0.07200
Ferro	0.05	0.00000	6.39	0.01470	0.07800