

# Effect of Antecedent Conditions on Prediction of Pore-Water Pressure Using Artificial Neural Networks

Muhammad Raza Ul Mustafa

Civil Engineering Department, Universiti Teknologi Petronas, Tronoh 31750, Perak, Malaysia

Tel: 60-19-595-7132 E-mail: [raza\\_geo@hotmail.com](mailto:raza_geo@hotmail.com)

Rezaur Rahman Bhuiyan

Water Resources Engineer, Golder Associates Ltd. Calgary T2A 7W5, Alberta, Canada

Tel: 1-403-299-5600 E-mail: [cerezaur@gmail.com](mailto:cerezaur@gmail.com)

Mohamed Hasnain Isa

Civil Engineering Department, Universiti Teknologi Petronas, Tronoh 31750, Perak, Malaysia

Tel: 60-5-368-7346 E-mail: [hasnain\\_isa@petronas.com.my](mailto:hasnain_isa@petronas.com.my)

Saied Saiedi

Civil Engineering Department, Universiti Teknologi Petronas, Tronoh 31750, Perak, Malaysia

Tel: 60-5-368-7347 E-mail: [saied.saiedi@yahoo.com](mailto:saied.saiedi@yahoo.com)

Harianto Rahardjo

School of Civil & Environmental Engineering, Nanyang Technological University

Nanyang Avenue 639798, Singapore

Tel: 65-6790-5246 E-mail: [chrahardjo@ntu.edu.sg](mailto:chrahardjo@ntu.edu.sg)

Received: October 9, 2011

Accepted: December 26, 2011

Published: February 1, 2012

doi:10.5539/mas.v6n2p6

URL: <http://dx.doi.org/10.5539/mas.v6n2p6>

## Abstract

The effect of antecedent conditions on the prediction of soil pore-water pressure (PWP) using Artificial Neural Network (ANN) was evaluated using a multilayer feed forward (MLFF) type ANN model. The Scaled Conjugate Gradient (SCG) training algorithm was used for training the ANN. Time series data of rainfall and PWP was used for training and testing the ANN model. In the training stage, time series of rainfall was used as input data in one model whereas, rainfall and pore water pressure with some antecedent conditions was used in second model and corresponding time series of PWP was used as the target output. In the testing stage, data from a different time period was used as input and the corresponding time series of pore-water pressure was predicted. The performance of the model was evaluated using statistical measures of root mean square error (RMSE) and coefficient of determination ( $R^2$ ). The results of the model prediction revealed that when antecedent conditions (past rainfall and past pore-water pressures) are included in the model input data, the prediction accuracy improves significantly.

**Keywords:** Antecedent, Artificial Neural Network, Pore-water Pressure, Prediction, Rainfall

## 1. Introduction

Variations in soil pore-water pressure (PWP) due to rainfall are known to exhibit highly non-linear and complex relationship. This is due to the spatial and temporal variability of precipitation, evaporation pattern and soil properties. The knowledge of PWP is very important in the determination of strength and effective stress of a soil. Excessive PWP increase is known to cause slope failures in areas susceptible to landslide. To arrive at effective remedial and design strategies against slope failure it is necessary to know the PWP changes due to rainfall.

Tensiometers are usually used to measure PWP at different depths of soil. In general, PWP of soils depends on several soil and climate-related factors such as rainfall, soil properties (grain size, porosity, density, etc.), temperature, evaporation, solar radiation, soil depth, and antecedent conditions. Therefore, reliable prediction of pore-water pressure is significantly data-demanding.

Use of Artificial Neural Network (ANN) techniques to solve problems in civil engineering began in the late 1980's (Flood, & Kartam, 1994). Applications of ANN techniques for simulation and forecasting in water resources engineering are few and relatively recent. ANN has been applied with success in estimation and forecasting of discharge capacity of channels (Unal, Mamak, Seckin, & Cobaner, 2010), suspended sediment (Mustafa, Isa, & Rezaur 2011), discharge (Tawfik, Ibharim, & Fahmy, 1997) and hourly and daily stream flows (Kang, Kim, Park, & Ham, 1993). A comprehensive review carried out by the American Society of Civil Engineering (ASCE) Task Committee on Application of ANN in hydrology concluded that ANN can perform well as existing models (ASCE, 2000a, 2000b). The ANN approach to nonlinear behavior modeling is more effective and more efficient whenever an explicit knowledge of the hydrological process is not required (Hsu, Gupta, & Sorooshian, 1995). It appears that for predicting PWP variations application of ANN could be an ideal alternative to regression based approaches.

The objectives of this study are (i) to develop an ANN model using Scaled Conjugate Gradient (SCG) learning algorithm for predicting time series of pore-water pressure responses to variations in rainfall pattern, and (ii) to evaluate the influence of antecedent conditions (past rainfall and past pore-water pressures) on the prediction accuracy of the ANN model.

## 2. ANN Model Theory

The architecture of ANN is a data processing system that consists of a large number of simple units called neurons or nodes having a local memory and interconnected with weights and biases. ANN differ from the traditional physically based models in a way that, ANN has the ability of self adaptation, can capture functional relationships and extract patterns between input and output variables by learning from examples even if the primary relationship is complex and difficult to describe by a physically based relationship. Thus application of ANN seems appropriate for problems whose solutions necessitate information that are hard to describe and lack of sufficient data or observations. Due to its self-learning, self-adaptable processing, non-linear pattern classification, and identification capabilities, application of ANN to different aspects of hydrologic modeling has attracted interest in recent years (Cigizoglu, 2004; Cigizoglu, & Alp, 2004; Hu, Lam, & Ng, 2001).

Application of an ANN consists of three steps; (i) training, (ii) validation and (iii) testing. Available data are divided for all these steps. Usually, as a rule of thumb, 60% of the available data are used for training, 20% for validation and the remaining 20% for testing. However, this rule is not fixed and could be changed depending on the availability of data. Training and validation could be viewed as the same process because in both steps input and target data are introduced to the network and the network is trained. Therefore, in some instances when a large dataset is not available, training and validation could be combined together in a single training session (i.e. 80% data could be used for training instead of 60% for training and 20% for validation). During testing, a new set of data which has not been used in training is provided to the network to produce the output (prediction). To assess the performance of the ANN model the outputs generated during testing are compared with the corresponding observations.

An ANN is designed by weights between the neurons, transfer function and learning laws (Caudill, 1987). Multilayer Feed Forward (MLFF) networks consists of more than one layer (i.e. input, hidden and output layers) and in Feed Forward (FF) networks all the information are transferred in the forward direction only, i.e. from input neurons to output neurons through the hidden layer. There is no cycle or loop in the feed forward network. The architecture of an MLFF network is shown in Figure 1. Subscripts  $i, j$ , and  $k$  denote the  $i^{\text{th}}$  ( $1 \leq i \leq L$ ),  $j^{\text{th}}$  ( $1 \leq j \leq M$ ), and  $k^{\text{th}}$  ( $1 \leq k \leq N$ ), neuron in the input, hidden and output layers, respectively. The letters  $L, M, N$  denote the number of neurons in the input, hidden and output layers respectively. The symbols  $y_k$  and  $z_k$  denote the output and target values respectively of the  $k^{\text{th}}$  neuron in the output layer.

The objective of training is to minimize the error between target and output values by adjusting the weights and biases through an algorithm called the learning law. During training a neuron obtains inputs from the previous layer which is multiplied by its weight and the bias value is added. Thus, the combination of the net weighted inputs and bias  $net_j$  to the  $j^{\text{th}}$  neuron in the hidden layer is represented as

$$net_j = \sum_{i=1}^L x_i w_{ji} + b_j \quad (1)$$

where,  $x_i$  is the input to the  $i^{\text{th}}$  neuron in the input layer;  $w_{ji}$  is the weight of  $j^{\text{th}}$  neuron of the hidden layer connected to the  $i^{\text{th}}$  neuron of the input layer and  $b_j$  is the bias of the  $j^{\text{th}}$  neuron in the hidden layer. The net input to the  $j^{\text{th}}$  neuron of the hidden layer is then passed through a transfer function  $f$  to produce the output of the  $j^{\text{th}}$  neuron in the hidden layer. If the activation level of neurons is strong enough then it produces an output. The output of the  $j^{\text{th}}$  neuron in the hidden layer can be expressed as

$$y_j = f(\text{net}_j) = f\left(\sum_{i=1}^L (x_i w_{ji} + b_j)\right) \quad (2)$$

The output of the hidden layer is an input to the neuron in the output layer and the same operation as in the hidden layer is repeated in the output layer to obtain the output of neurons in the output layer. The net weighted input and bias to the  $k^{\text{th}}$  neuron in the output layer is represented as

$$\text{net}_k = \sum_{j=1}^M f(y_j) w_{kj} + b_k = \sum_{j=1}^M w_{kj} f\left(\sum_{i=1}^L (x_i w_{ji} + b_j)\right) + b_k \quad (3)$$

where,  $w_{kj}$  is the weight of  $k^{\text{th}}$  neuron of the output layer connected to the  $j^{\text{th}}$  neuron of the hidden layer and  $b_k$  is the bias of the  $k^{\text{th}}$  neuron in the output layer. The output from the  $k^{\text{th}}$  neuron in the output layer is given by

$$y_k = f(\text{net}_k) = f\left(\sum_{j=1}^M w_{kj} f\left(\sum_{i=1}^L (x_i w_{ji} + b_j)\right) + b_k\right) \quad (4)$$

The error  $e_k$  of the  $k^{\text{th}}$  neuron in the output layer which is the squared difference between the target and output values can be written as

$$e_k = (z_k - y_k)^2 \quad (5)$$

If there are  $P$  numbers of input data pairs such that ( $1 \leq p \leq P$ ), the global error of the network in terms of mean squared error is written as

$$E = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^N (z_{pk} - y_{pk})^2 \quad (6)$$

Equation (6) is called the error function of the network and is a function of network connection weights. During training the global error function is minimized by the learning rule. The learning rule enables updating (adjusting) the connection weights through successive iterations such that the difference between target and output values for all dataset is within a predefined tolerance limit.

There are several learning algorithms available, the commonly used ones are the (i) Back propagation (BP), (ii) Back propagation with momentum, (iii) Conjugate Gradient (CG), (iv) Scaled Conjugate Gradient (SCG), (v) Quasi Newton's and (vi) Levenberg-Marquardt (LM) algorithm. Each of these algorithms has its own distinct advantages and weaknesses. However, the SCG algorithm is based on the second order gradient supervised learning rule and is claimed to have learning speed (convergence rate) about 2 times faster than the CG algorithm and about 20 times faster than the regular BP algorithm (Moller, 1993; Schraudolph, & Graepel, 2002). Other major advantages of the SCG algorithm are that it does not depend on any critical user selected parameter (e.g. learning rate, momentum) as in BP algorithm and it does not involve computationally expensive line-search to scale the step size as in CG algorithm, instead it uses a trust-region method to scale the step size (Cestisli & Barkana, 2010; Falas, & Stafylopatis, 2005).

### 3. Methodology

#### 3.1 Data Source and Study Area

The data used in this study are the time series of pore-water pressure and rainfall records from 3 m soil depth, for a period of three and a half months and at 4 hour interval during dry periods (no rainfall) and 10 min interval during wet periods. The data was collected through a field instrumentation program of a residual soil slope in Yishun, Singapore (Rahardjo, Leong, & Rezaur, 2008; Rezaur, Rahardjo, Leong, & Lee, 2003). The three and a half month time series data consists of 1450 pore-water pressure measurements varying between both positive and negative values with a maximum negative PWP of magnitude  $-15$  kPa. The entire monitoring period spanned over three years and included time series of pore-water pressure and rainfall measurements at 4 different

slopes in 2 major geological formations (Bukit Timah Granitic Formation and Sedimentary Jurong Formation) and at soil depths 0.5, 1, 2, and 3 m. The data was primarily collected with a view to understand rainfall-induced slope failure mechanism (Rahardjo, et al., 2008) and hydrological responses of slopes under tropical climate (Rezaur, et al., 2003). Climate at the study area is hot and humid (equatorial) with no marked dry season. The temperature varies little throughout the year with an annual average of 26.6 °C and a relative humidity of 84% (Meteorological-Service-Singapore, 1997). The soil at the Yishun site is residual soil from the Bukit Timah granite which varies from silty or clayey sand to silty or sandy clay, depending on the degree of weathering, but is commonly sandy clayey silt (Rahardjo, et al., 2008).

### 3.2 ANN Architecture

Two different ANN model architectures were used in this study; hereafter called the ANN1 and ANN2. ANN1 consisted of one neuron in the input layer for rainfall, 4 neurons in the hidden layer and one neuron in the output layer, denoted by ANN architecture 1–4–1. ANN2 denoted by 8–4–1 architecture, consisted of 8 neurons in the input layer out of which 3 were used for rainfall values (1 current and 2 antecedent) and the remaining 5 were used for PWP (antecedent values). ANN1 architecture was adopted to examine the suitability of modeling the variation of pore-water pressure at the current time  $u_t$  as a direct function of variation of rainfall at current time  $r_t$ , (i.e.  $u_t = f(r_t)$ ), i.e. without accounting for any effect from antecedent conditions. Whereas ANN2 architecture was adopted to examine the suitability of modeling the variation of pore-water pressure as a function of variation of rainfall and pore-water pressure at the present time  $t$  and past (antecedent) times ( $t-1, t-2, \dots, t-5$ ) such that;

$$u_t = f(r_t, r_{t-1}, r_{t-2}, u_{t-1}, u_{t-2}, u_{t-3}, u_{t-4}, u_{t-5}) \quad (7)$$

There is no hard and fast rule for selecting the number of neurons in the hidden layer. In this study, the number of neurons in the hidden layer was arrived at by trial and error method (Maier, & Dandy, 2000). A program code using ANN toolbox in MATLAB was written for the application of the ANN algorithm described in the ANN model and theory section.

### 3.3 Input Data Selection

Success in the identification of a non-linear system by ANN training depends on the selection of appropriate training data which should be representative of the non-linear system to be mapped during the ANN training (Rojas, 1996). In this study the available three and half month synchronized time series data of pore-water pressure and rainfall was divided into two sets. Data from Oct 12–1998 to Dec 17–1998 (about 70% of the available data) was used for training-validation while data for the period Dec 18–1998 to Jan 24–1999 (remaining 30% of the data) was used for testing (prediction).

### 3.4 Activation Function Selection

In this study, the hyperbolic tangent sigmoid transfer function, also known as hyperbolic tangent or tansig (Eq. 8) was used for neurons in the hidden layer. Linear transfer function, also known as purelin (Eq. 9) was used for neurons in the output layer.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (8)$$

$$f(x) = x \quad (9)$$

### 3.5 Data Normalization

The input and target data (raw data) need to be normalized before use in the ANN training and testing to commensurate with the upper and lower bound limits of the activation functions used in the hidden neurons. This ensures fast processing and convergence during training and minimizes prediction error (Rojas, 1996). Since hyperbolic tangent sigmoid activation function whose upper and lower bounds are in the range of  $-1$  to  $1$  was used in the hidden layer neurons, the pore-water pressure and rainfall data was normalized by transforming the data to the range of  $-1$  to  $1$  using the equation;

$$z_p = 2 \times \frac{(x_p - x_{\min})}{(x_{\max} - x_{\min})} - 1 \quad (10)$$

where,  $z_p$  is the normalized or transformed data series,  $x_p$  is the original data series,  $x_{\min}$ ,  $x_{\max}$  are the minimum and the maximum value of the original data series respectively,  $1 \leq p \leq P$  and  $P$  is the number of data.

### 3.6 Performance Evaluation Criteria

In hydrological studies root mean square error (RMSE) is commonly used to evaluate model performance. The ideal value of RMSE for best performance is zero. In this study, the performance of the model was evaluated using RMSE between the observed and predicted values, as

$$\text{RMSE} = \left[ \frac{1}{P} \sum_{p=1}^P (z_p - y_p)^2 \right]^{1/2} \quad (11)$$

where,  $z_p$  and  $y_p$  are the observed and predicted values of pore-water pressure, respectively,  $P$  is the number of observations for which the error has been computed.

### 3.7 Stopping Criterion

The rule for stopping the training was based on either the relative error of the sum of square error  $E \leq \text{goal}$  or the maximum number of given epochs, whichever is satisfied first. In the program code, values of 0.001 and 500 were used for goal and number of epochs, respectively.

## 4. Results and Discussion

The summary of ANN model performance statistics are shown in Table 1. The performance of both the ANN models in terms of minimizing the mean square error (MSE) to a desired goal and the time and number of epochs to reach the desired goal during training of the networks are shown in Figure 2. ANN1 model started training with an initial global MSE of 5.29 and minimized the MSE to 0.3 within only 3 epochs. Thereafter, the network appears to be trapped with local minima and the MSE remained constant at 0.28 until the maximum number of epochs (500) was reached, the goal however was not achieved (Figure 2a). For ANN2 model the MSE decreased very fast (0.8 to 0.003) for the first 7 epochs and then continued to decrease at a slower rate until the goal (0.001) was reached at 82 epochs (Figure 2a). Similarly, Figure 2b shows that for ANN1 the MSE decreased from 5.29 to 0.3 in 0.02s but remained constant at 0.28 till the training finished at 4s when the maximum number of epochs (500) was reached. The ANN2 model minimized the error very fast and achieved the goal with only 0.83s (Figure 2b).

A comparison between observed and trained time series of pore-water pressures are shown in Figure 3. Figure 3a shows that ANN1 model failed to learn the non-linear behavior of pore-water pressure throughout the training duration and the trained values did not follow the target values. While the ANN2 model learned the complex behavior of the data well, adopted the non-linearity and then followed the same trend as the target values (Figure 3b).

Prediction results of time series of pore-water pressures during testing stage of the ANN models are shown in Figure 4. It is clear that ANN1 (with architecture 1-4-1) which gave poor results during training (Figure 3a) also showed bad performance during testing (Figure 4a). ANN2 (with architecture 8-4-1) performed well during both training (Figure 3b) and prediction (Figure 4b). Figure 3b and Figure 4b show that in both cases i.e. during training and prediction, the trends in the trained and predicted time series followed closely the trends of the observed pore-water pressures.

Comparison between observed pore-water pressures and pore-water pressures predicted by the ANN models are shown in Figure 5. The coefficient of correlation ( $R^2$ ) and the line of perfect agreement between observed and predicted values are also shown in Figure 5. The correlation between the observed and predicted pore-water pressures for ANN1 is very poor ( $R^2=0.065$ , Figure 5a) whereas the correlation between the observed and predicted pore-water pressures for ANN2 is good ( $R^2=0.973$ ); confirming the superior performance of ANN2 model (Figure 5b).

A comparison of performance statistics in terms of RMSE, epochs, and time to reach goal in Table 1 show the high superiority of ANN2 over ANN1, both in training and testing stages. Both networks ANN1 and ANN2 were trained with the same learning law (SCG), had the same number of neurons in the hidden (4 neurons) and output (1 neuron) layers; while the only difference between ANN1 and ANN2 was the number of neurons in the input layer (ANN1: 1 input neuron, ANN2: 8 input neurons) and the choice of input parameters. The superior performance of ANN2 over ANN1 both in training and prediction could be attributed to the additional number of neurons in the input layer and the inclusion of antecedent conditions. The 8 input neurons in ANN2 were used to account for antecedent rainfall and pore-water pressure conditions (see Eq.12) and this resulted in superior performance in training and prediction by ANN2. This shows that during pore-water pressure prediction it is necessary to account for antecedent pore-water pressure and rainfall conditions.

The prediction results (Figure 4b, Figure 5b and Table1) also suggest that SCG training algorithm is suitable for pore-water pressure prediction. ANN model with appropriate network architecture can be used for predicting time series of pore-water pressure responses to variations in rainfall pattern. It also appears that time series of pore-water pressure responses to rainfall which is known to be a function of many factors such as rainfall, evaporation, soil properties, antecedent conditions (Rahardjo, et al., 2008; Rezaur, et al., 2003) could be predicted using ANN from the knowledge of a relatively few factors only.

## 5. Conclusions

Multilayer Feed Forward neural network model with Scaled Conjugate Gradient (SCG) learning algorithm was developed to predict time series of pore-water pressure responses to rainfall. An appropriate network configuration that could map the nonlinear behavior of pore-water pressure responses (at 3 m soil depth) to climatic condition was identified to be 8–4–1. The study indicated that the SCG learning algorithm is suitable for application to problems associated with predictions of non-linear and complex behavior such as pore-water pressure variation during rainfall.

Predictions with a network architecture of 1-4-1 configuration led to unacceptable errors. The superiority of prediction accuracy with 8-4-1 configuration indicates the necessity of incorporating the antecedent values for the parameters. In other words, it is necessary to account for antecedent pore-water pressure and rainfall for prediction of pore-water pressure with reasonable accuracy.

Highly dynamic, non-linear and complex behavior of pore-water pressure, which is a function of many independent variables (i.e. rainfall, evaporation, soil properties, soil depth and antecedent conditions), could be predicted with ANN models with a modest number of input variables.

## Acknowledgment

The first author gratefully acknowledges the financial support provided by the Universiti Teknologi PETRONAS as part of a PhD scholarship.

## References

- ASCE, T. C. (2000a). Artificial Neural Networks in Hydrology. I: Preliminary Concepts pages. *Journal of Hydrologic Engineering*, 5(2), 115-124.
- ASCE, T. C. (2000b). Artificial Neural Networks In Hydrology. II: Hydrologic Applications. *Journal of Hydrologic Engineering*, 124-137.
- Caudill, M. (1987). Neural Networks Primer, Part I. *AI Expert*, 2(12), 46-52.
- Cestisli, B., & Barkana, A. (2010). Speeding up the scaled conjugate gradient algorithm and its application in neuro-fuzzy classifier training. *Soft Computing*, 14, 365-378.
- Cigizoglu, H. K. (2004). Estimation and forecasting of daily suspended sediment data by multi-layer perceptrons. *Advances in Water Resources*, 27(2), 185-195. <http://dx.doi.org/10.1016/j.advwatres.2003.10.003>
- Cigizoglu, H. K., & Alp, M. (2004). Rainfall-Runoff Modelling Using Three Neural Network Methods. In L. Falas, T., & Stafylopatis, A. (2005). Symbolic rule extraction with scaled conjugate gradient version of CLARION. *Proceedings of the International Joint Conference on Neural Networks, Montreal, Canada. July 31-August, 4*, 845–848.
- Flood, I., & Kartam, N. (1994). Neural network in civil engineering-I: Principles and understanding. *Journal of Computing in Civil Engineering, ASCE*, 8(2), 131-148.
- Hsu, K. L., Gupta, H. V., & Sorooshian, S. (1995). Artificial Neural Network Modeling of the Rainfall-Runoff Process. *Water Resour. Res.*, 31(10), 2517-2530. <http://dx.doi.org/10.1029/95wr01955>
- Hu, T. S., Lam, K. C., & Ng, S. T. (2001). River flow time series prediction with a range-dependent neural network. *Hydrological Sciences Journal*, 46(5), 729-745. <http://dx.doi.org/10.1080/02626660109492867>
- Kang, K. W., Kim, J. H., Park, J. C., & Ham, K. J. (1993). Evaluation of hydrological forecasting system based on neural network model. *Proceedings of the 25th Congress of IAHR IAHR Delft, The Netherlands*, 257–264.
- Maier, H. R., & Dandy, G. C. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental Modelling and Software*, 15(1), 101-124. [http://dx.doi.org/10.1016/S1364-8152\(99\)00007-9](http://dx.doi.org/10.1016/S1364-8152(99)00007-9)
- Meteorological-Service-Singapore. (1997). Summary of Observations (Annual Publication), Singapore.
- Moller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4),

525–533.

Mustafa, M. R., Isa, M. H., & Rezaur, R. B. (2011). A Comparison of Artificial Neural Networks for Prediction of Suspended Sediment Discharge in River- A Case Study in Malaysia. *International Conference on Environmental Sciences and Engineering ICESE-211*, River View Hotel, Singapore, 372-376.

Rahardjo, H., Leong, E. C., & Rezaur, R. B. (2008). Effect of antecedent rainfall on pore-water pressure distribution characteristics in residual soil slopes under tropical rainfall. *Hydrological Processes*, 22(4), 506-523. <http://dx.doi.org/10.1002/hyp.6880>

Rezaur, R. B., Rahardjo, H., Leong, E. C., & Lee. (2003). Hydrological behavior of residual soil slopes in Singapore. *Journal of Hydrologic Engineering, ASCE*, 8(3), 133-144.

Rojas, R. (1996). *Neural Networks: A Systematic Introduction*. Springer Verlag, Berlin, 151–184.

Rutkowski, J., Siekmann, R., Tadeusiewicz, & Zadeh, L. A. (2004). *Artificial Intelligence and Soft Computing – ICAISC* (Eds.), 3070, 166-171, Springer Berlin / Heidelberg.

Schraudolph, N. N., & Graepel, T. (2002). Towards stochastic conjugate gradient methods. *Proceedings of the 9th International Conference on Neural Information Processing, ICONIP-2002*. Piscataway, NJ, USA, 853–856.

Tawfik, M., Ibharam, A., & Fahmy, H. (1997). Hysteresis sensitive Neural Network for modeling rating curves. *Journal of Computing in Civil Engineering, ASCE*, 11(3), 206-211.

Unal, B., Mamak, M., Seckin, G., & Cobaner, M. (2010). Comparison of an ANN approach with 1-D and 2-D methods for estimating discharge capacity of straight compound channels. *Adv. Eng. Softw.*, 41(2), 120-129. <http://dx.doi.org/10.1016/j.advengsoft.2009.10.002>

Table 1. Performance statistics of ANN models during training and testing

Model	Architecture**	Goal	Time (sec)	Epochs	RMSE	
					Training	Testing
ANN1	1-4-1	0.001	3.28	500*	7.87	6.61
ANN2	8-4-1	0.001	0.83	82	0.48	0.98

\* Maximum numbers of epochs was reached but did not reach goal

\*\* Number of input neurons-Hidden neurons-Output neurons

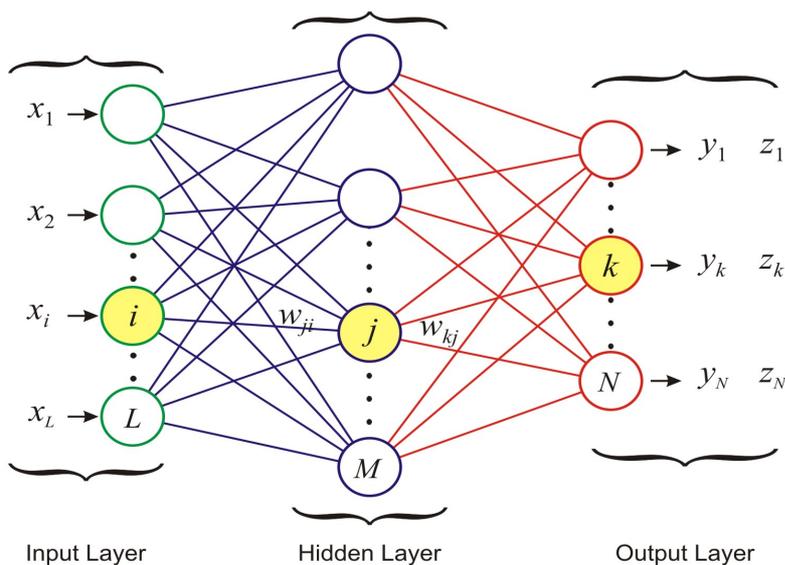


Figure 1. Schematic representation of an MLFF network

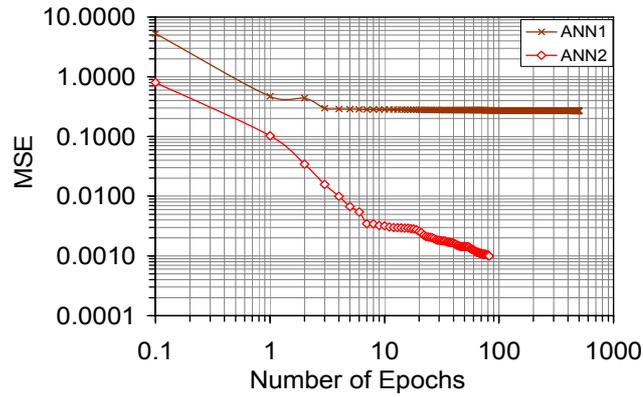


Figure 2a. Comparison of performance in terms of number of epochs to reach goal

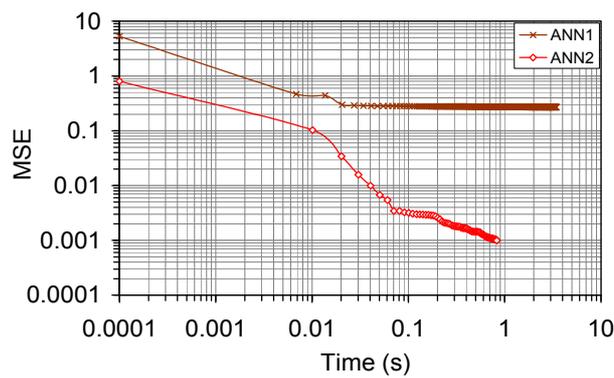


Figure 2b. Comparison of performance in terms of time to reach goal

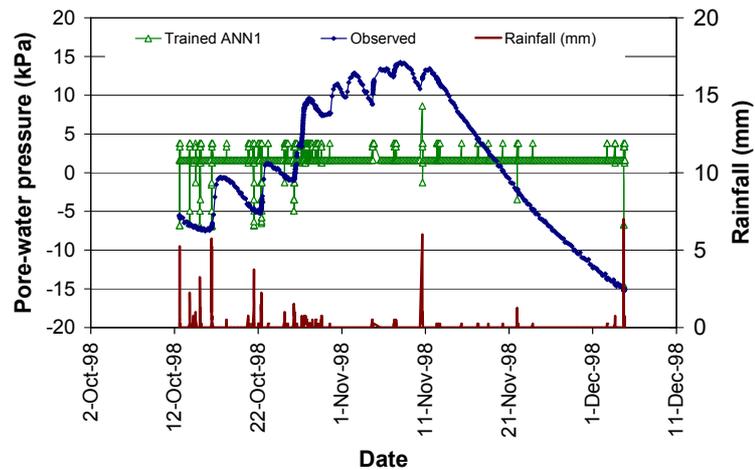


Figure 3a. Comparison between observed and trained time series of pore-water pressures after training using ANN1 model

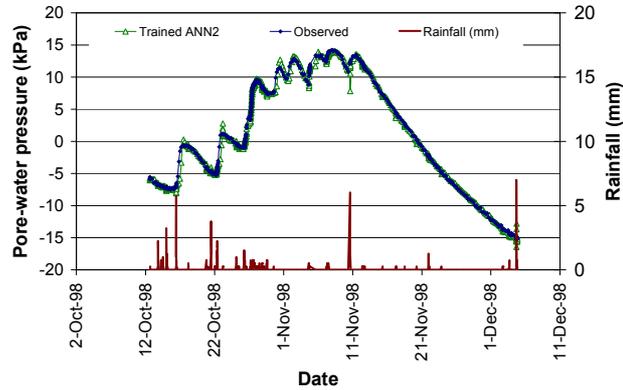


Figure 3b. Comparison between observed and trained time series of pore-water pressures after training using ANN2 model

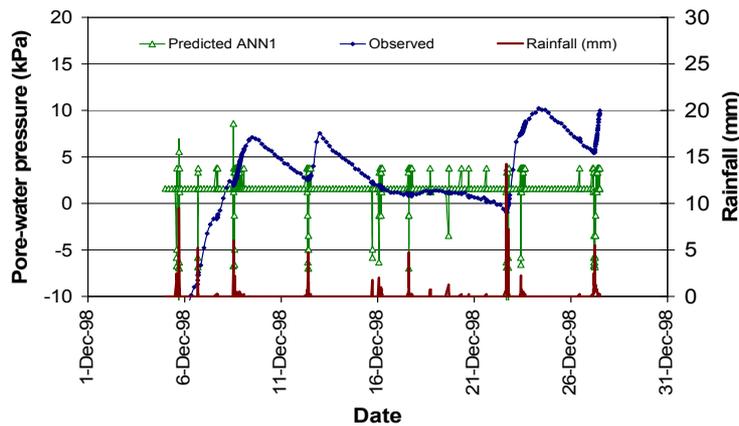


Figure 4a. Comparison between observed and predicted time series of pore-water pressures obtained during testing of ANN1 model

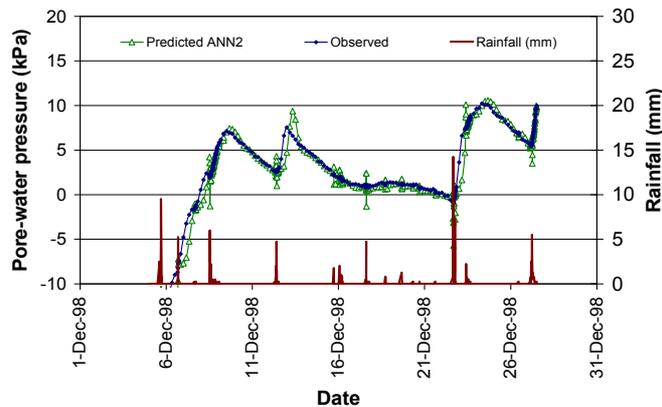


Figure 4b. Comparison between observed and predicted time series of pore-water pressures obtained during testing of ANN2 model

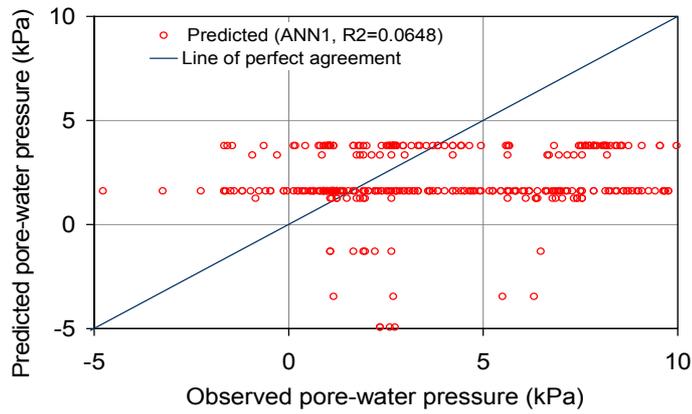


Figure 5a. Comparison between observed and predicted pore-water pressures of ANN1 model

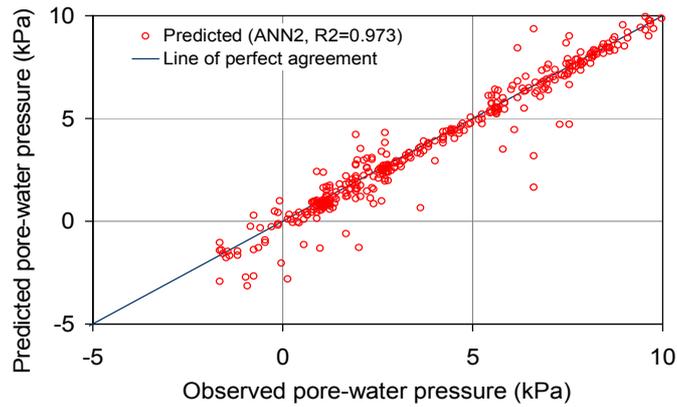


Figure 5b. Comparison between observed and predicted pore-water pressures of ANN2 model