Journal of Hydrosciences and Environment
ISSN: 2345-5608

The Comparison of Single-Layer and Two-Layer MLP Neural Networks with the LM Learning Method and ANFIS Network in Determining the Stability Factor of Earth Dams

H. R. Babaali[1,*], M. Heidari Chegeni[2], P. Beiranvand[3]

1. Assistant Professor, Department of Civil Engineering, Khorramabad Branch, Islamic Azad University, Khorramabad, Iran.
2. Master student of Civil Engineering, Soil and Foundation Trend, Islamic Azad University, Khomein Branch, Iran.
3. PhD Candidate, Department of Civil Engineering, Razi University, Kermanshah, Iran.

ARTICLEINFO

Article history:

Received: *4 April 2019*
Accepted: *22 November 2019*

Keywords:
Earth dam stability,
Neural network,
ANFIS network.

ABSTRACT

In this research, an MLP neural network, which has widely been used in geotechnical engineering problems, is selected and trained by defining the stability factor of an earth dam. To training the network, we first specify the effect of parameters on the earth dam stability, including dam height (H), dam width (B), dam slope (θ), internal friction angle (φ), specific gravity of soil (γ) and cohesion of soil (C) by the Plaxis finite element program. Then, a database of 240 earth dam models is created and used to train the network. Subsequently, we train a single-layer and a two-layer MLP neural network with LM method and compare them. The results show that the single-layer network exhibits better performance in processing time and training quality. Then, the results are compared with the results of the ANFIS network and it is shown that the ANFIS network has a lower capability in defining earth dam stability factor than the MLP network.

1. Introduction

Artificial neural networks, used for data processing, are inspired by biological neural systems so that they process data like a human brain. The main element in this idea is the new structure of the information processing system. The first artificial neuron cell was created in 1943 by neurophysiologist Warren McCulloch and logician Walter Pits. But, technology limitations in those days did not let them do more work.

It seems that neural network simulations are recent progress. Although it was founded before the emergence of computers and at least one historical major obstacle and several different periods have been passed. After an initial period of enthusiasm and activity on this topic, a period of reluctance and notoriety came.

During this era, investment and professional support of this topic were in the least. The important signs of progress that happened were proportional to the limited research, thereby pioneers were able to expand their convincing technology which was more prominent than limitations recognized by Minsky and Papert. Minsky and Papret published a book in 1969 which had been specified the public opinion about neural network excusion among researchers and thus this opinion was accepted without any analysis.

Neural networks with the ability to attention in inferencing complex or vague meaning of data can be used to extract patterns and recognize methods the knowledge of which is difficult and complex for humans and other computer techniques. A trained neural network can be regarded as an expert about the information that is given to it for analysis. This expert can be used to estimate new desired states and answer "what if" questions.

The slope and stability of earth dams have always been investigated by numerical and limit equilibrium methods. Limit equilibrium methods have been employed to a greater extent because of their ease and simplicity. Although their simplifying assumptions impair their accuracy considerably, this problem will be solved with numerical methods. It should be considered that to use these methods, in addition to the awareness of finite element knowledge and existing software, considerable time should be devoted to preparing input data. Neural networks can be a proper replacement for the existing methods because neural networks are highly flexible in learning problems and, at the same time, they enjoy the advantages of numerical and limit equilibrium methods, obviate their limitations and faults.

In this research, an MLP neural network, which is commonly used in geotechnical engineering problems, is selected. With LM method single and two layer network are trained and among them the network with better function is selected. Then, the results are compared with the results of the ANFIS network. It is specified that the ANFIS network proportional to the MLP network has a lower capability in defining the earth dam stability factor.

[*]*Corresponding author's email:*
h_babaali@khoiau.ac.ir

2. Methods

2.1. Artificial Neural Network Method

ANNs have increasingly been developed to find quick solutions with high results. ANNs help speed up data analysis and processing of its usefulness to identify the data given and to find a solution to unseen data and to know its behavior over future periods or classifying it according to its figures. ANNs are mathematical models that can create a linear model of data that has nonlinear characteristics. It, therefore, reduces the sources of storage needed and the easy identification of the behavior of these data. ANNs manipulate data in a way similar to a human brain. ANNs learn by example. A typical feed-forward network has neurons arranged in a distinctly layered topology. ANNs are used in different fields of research like education, engineering, medicine, social applications, etc. Learning techniques involve adjustments to the weights that exist with each neuron in the training process. After the weights are adjusted, the network can generalize the results to execute the unseen data. An epoch defines the number of times that the algorithm examines the entire data set. The most commonly used ANN architectures are feed-forward networks, recurrent networks, and stochastic networks. Typically, the network architecture determines the set of connections between particular nodes. It consists of input layers, hidden interconnection layers, and output layers. There are several topologies of neural networks and depend on the type of learning algorithm used, i.e., supervised or unsupervised. The feed-forward neural network manipulates the data in two phases. First, the training process is done to adjust the ANN weights to match the input group. The network is then generalized to match the collection of training data for the conclusion of the unseen samples. The training of data repeats the learning process and updates the weights until the desired results are achieved.

Multi-Layer Perceptron (MLP) is a type of ANN with one input and output layer and two hidden layers. The input layer distributes the input values to the hidden layer. Eq. (1) determines the output of the first layer, which is the sum of the inputs multiplied by its weights.

$$Y_i = f(\sum w_i x_i) \quad (1)$$

where x_i is an input signal, w_i is the weighting of strengths, and Y_i is the output as a function of the sum. Learning by the ANN is done by training algorithms that expand according to the learning rule, which can learn to do complex tasks, like pattern recognition, system identification, prediction, process control, and function approximation.

The performance of ANN is computed using different methods such as MSE and R. The average of the square of the error is called the mean square error (MSE), which is the target of prediction or estimation and is referred to as the second moment of the error measured after the origin. Eq. (2) determines the value of the MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - X_i)^2 \quad (2)$$

where Y_i is the n number of predictions and X_i is the n number of true values.

The correlation coefficient gives the proportion of variance of the variable that is predictable from the other variables as computed in Eq. (3) which calculates the value of R. The smallest value of R is -1, which indicates that the two variables are negatively related; for $R = 0$, there is no relationship between them; and when R approaches 1, the relationship is strong between the variables.

$$R = \frac{\sum_{i=1}^{n}(x_i - \dot{x})(y_i - \dot{y})}{\sqrt{\sum_{j=1}^{n}(x_j - \dot{x}) \cdot \sum_{k=1}^{n}(y_k - \dot{y})}} \quad (3)$$

where R is the correlation coefficient of the independent variables x and y, $\dot{x}$ and $\dot{y}$ are the mean values. If $R = -1$, it shows the best negative correlation; if $R = 1$, it indicates the best positive correlation; and if $R = 0$, it means no correlation.

2.2. Neural networks vs. normal computers

Neural networks use different paths for problem-solving than normal computers. Normal computers use an algorithmic path which means that the computer uses a set of instructions to solve the problem without the steps it should pass being known. Computers are not capable of solving problems. This fact limits the normal computers' problem-solving capability to problems we are capable of solving that and know how they are solved. But, if computers could do tasks which we did not know how to do it exactly, they would be more profitable.

Neural networks process information in a way similar to a human brain. They consist of lots of highly interconnected process elements (neural cells) that work in parallel to solve a specific problem. Neural networks work with samples and cannot be programmed for a specific task. Samples should be selected accurately; otherwise, a useful time will be wasted or worse, the network might work wrong. The privilege of neural networks is that they themselves discover how to solve a problem and their performance is unpredictable.

On the other hand, normal computers use a specific path to solve a problem. The solution by which the problem is solved should be known before and should be described by short and unambiguous commands. These commands are then converted into high-level programming languages and after that, they are converted into codes that computers are capable of understanding. In general, these machines are predictable and if there is a problem, it will be a hardware or software problem.

Neural networks and normal computers do not compete with each other; rather, they completing one another. There are tasks that are more proper for algorithmic methods such as arithmetic operations, but there are also tasks that are more proper for neural networks. Beyond this, there are problems that need a system which is a combination of both methods. Neural networks do not do a miracle, but if they are used wisely, they can yield wonderful results.

2.3. Other advantages of neural networks

Some other advantages of neural networks can be summarized as below:

- Adaptive learning: the capability of learning how to do tasks based on given information for practice and primary experience.
- Self-organization: an ANN can create its organization or presentation for the information receiving during learning time itself.
- Proper performance: ANN computations can be done in parallel, and special hardware has been designed and created to use this capability.
- Fault tolerance without pause while coding information: networks' partial damage can cause performance reduction although some of the networks' capabilities might even stay with great damage.

2.4. Main components of neural network

2.4.1. Structure or Architecture

When dealing with neural networks, the first thing that draws attention is its structure. Structure or architecture is the physical embodiment of a neural network. The structure of a neural network defines its schema and the inside connections of this intelligent system. From a practical perspective, this aspect of neural networks is its information storing device. Information of mapping this implicit intelligent system is stored in its structure accurately. Moreover, when responding to users' questions, neural networks retrieve the required data from their structure.

2.4.2. Training mechanism or learning algorithm

Training mechanism creates a map for the neural network by checking the samples of the training phases, which is an estimation of real mapping of problem's system. In the structure of neural networks, these samples are called training samples or learning algorithms.

2.4.3. Test mechanism or Invoke

Test or invoke the mechanism of neural networks use information extracted from the training mechanism or learning algorithm and answer the users' questions. The test mechanism retrieves information from the structure of the neural network. Actually, the structure of a neural network is a type of memory in this intelligent system.

3. Results and Discussion

3.1. Effective parameters in earth dam stability

In this research, six important and effective parameters in earth dam stability are considered: dam height, dam width, slope, internal friction angle, the specific gravity of soil and cohesion of soil. Based on Table.1, different dams are modeled in the Plaxis program and safety factors are obtained in different states, which are used as the input of the neural networks.

Table 1. Specifications of dams modeled in the Plaxis

1	Dam height	20m, 30m, 40m
2	Dam width	5m, 10m
3	Dam slope (θ)	18.43, 26.6, 45
4	Cohesion of soil (C)	15, 20, 25 KPa
5	Internal friction angle (φ)	20, 27, 35
6	Specific gravity of soil (γ)	17,19, 21 KN/m^3

3.2. Training the network with Levenberg-Marquardt method

The structure of the network in question is shown in Fig.1.

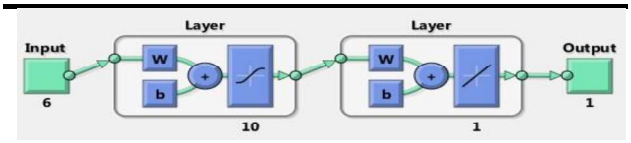


Fig. 1. A single-layer network with 10 neurons in each layer

In order to gain a better result from the network, data is converted to their dimensionless counterparts in [-1 1] domain, the network is trained, and then the data are converted back to their dimensionful state. So, 20% of data are considered for testing which means they are not used in the training of the network. In total, there is 240 data 48 of which are kept for testing and 192 remainings are used in training.

In this method, the iteration number is considered to be 200. To train the network, the neural network toolbox of the MATLAB software is used although in training with the genetic algorithm, the programming computer is written independently and the toolbox is not used. Fig. 2 is a graphic schema of this toolbox while running the program.

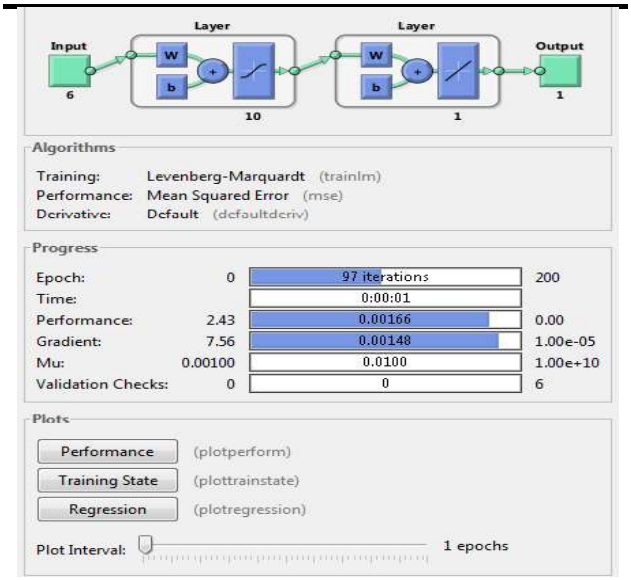


Fig. 2. Neural Network Toolbox

After 200 network training iterations, the network output for the training data and testing with real data is shown in Fig. 3.

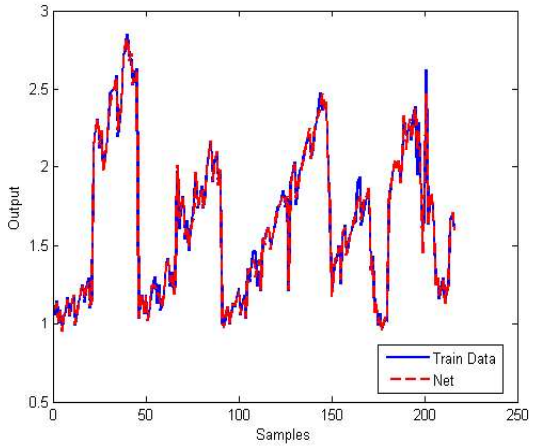


Fig. 3. The output of the single-layer network with tangent function for training data

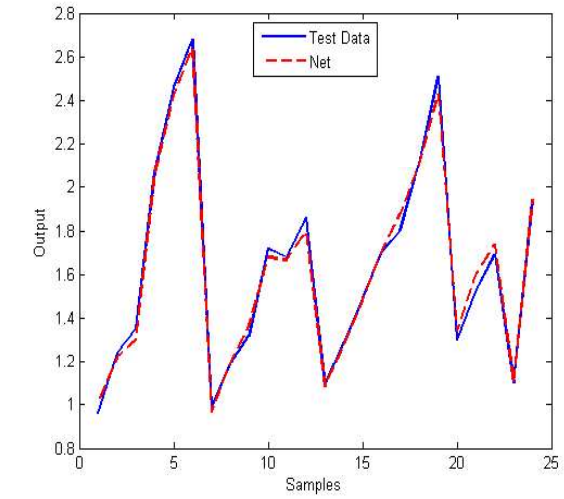


Fig. 4. The output of the single-layer network with tangent function for test data

As can be seen in Figs. 3 and 4, the network has been trained properly for training and testing data and the network output is accurately on real data. Error values for the training and testing data are shown in Fig. 5.

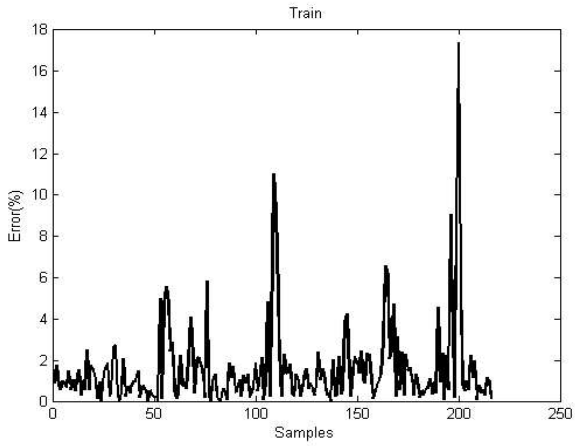


Fig. 5. The error of the single-layer network with tangent function for training samples with LM learning method

Based on Fig. 5, it is obvious that the network error for training data is less than 10% and only in one case, the error is about 20%. Fig. 6 depicts the diagram for the testing data.

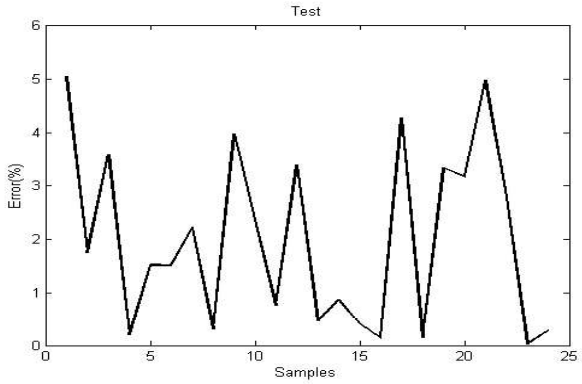


Fig. 6. The error of the single-layer network with tangent function for the testing data

Based on Fig. 6, the error is generally less than 5% which is very appropriate. The other criterion use to investigate the quality of the network is the correlation coefficient between real data and the network output. The closer the correlation is to 1, the better the quality of the network is. The correlation coefficient between real data and network data for training data (left side) and testing data (right side) is shown in Fig. 7.

Clearly, the correlation coefficient for the training and test data is close to 1. The correlation diagram for whole the data is shown in Fig. 8. It is 0.996, implying a good value.

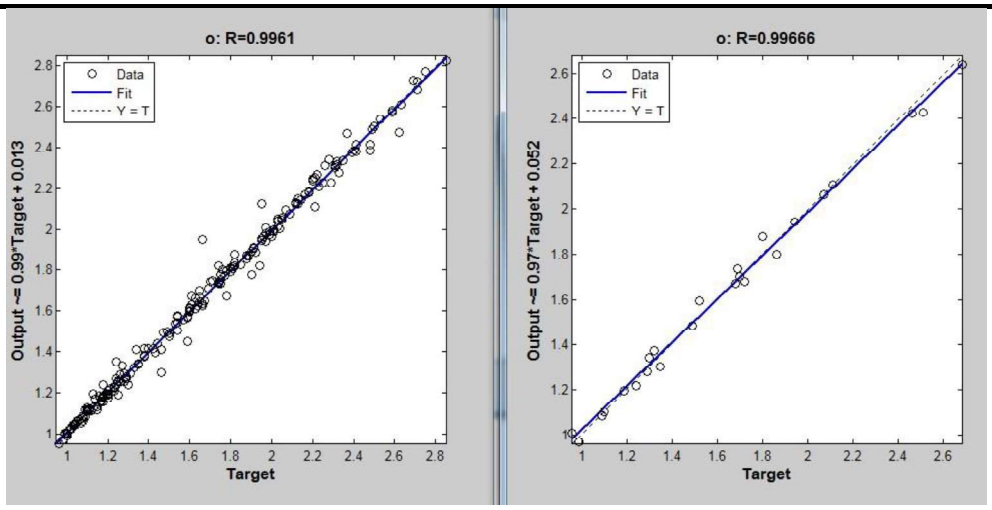


Fig. 7. The correlation coefficient of the single-layer network with tangent function for training and testing data

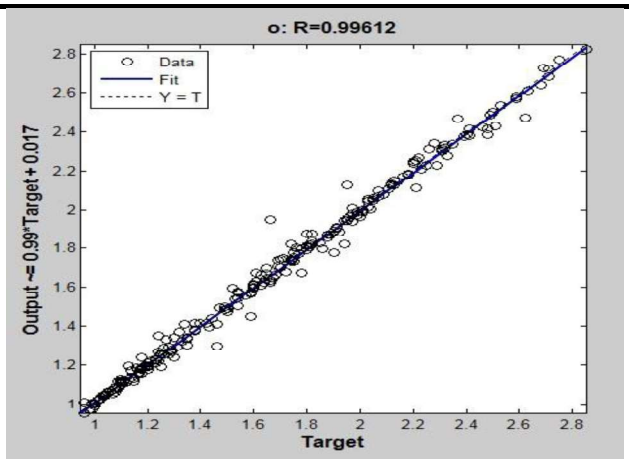


Fig. 8. The correlation coefficient of the single-layer network with tangent function for whole the data

3.2.1. Comparison of the single-layer and two-layer networks

By increasing the layers of a network, the network complexity and the number of the problem parameters are increased. So, it is predictable that by increasing the number of layers, proper training happens because of the stronger structure of the network and the larger number of variables data over fitting might happen and the result of network might be inappropriate for test data. This means that the network will try hard to fit the training data and define parameters in such a way that it has the minimum error in training data, while the required model was not so complex in the first place. So, in complex and non-linear models, increasing the number of layers can reduce error for training and test data but if the system's nature is not such complex, the two-layer network will cause an error.

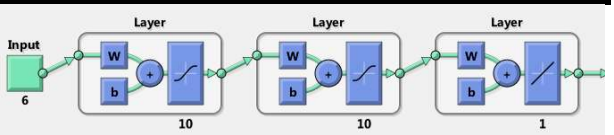


Fig. 9. A two-layer network with 10 neurons in each layer

Fig. 10 compares the output of training data with real data.

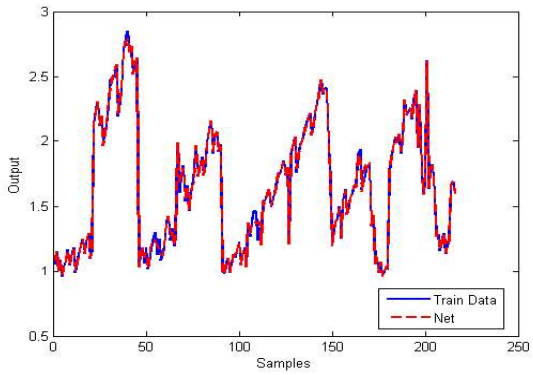


Fig. 10. The output of a two-layer network with tangent function for training data

It is clear that the results are in perfect agreement with the real results. For more accuracy, the error diagram is illustrated in Fig. 11.

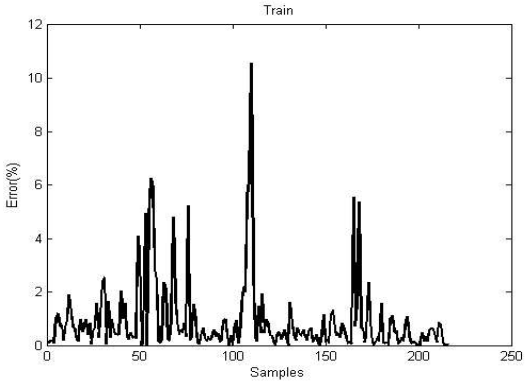


Fig. 11. The error of the two-layer network with tangent function for training samples with LM learning method

It is obvious that the error of the training data was reduced by almost half versus the single-layer network whose error was 15%. The output of the network in front of real data for test data is shown in Fig.12.

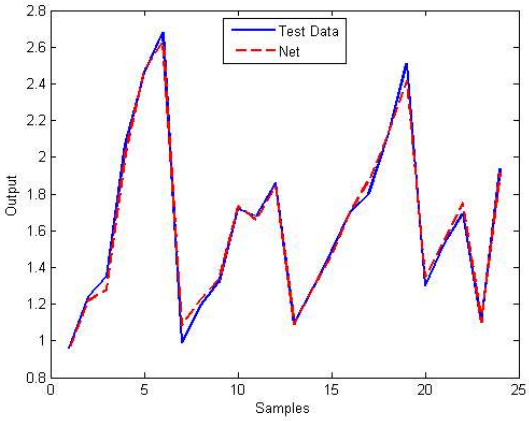


Fig. 12. The output of the two-layer network with tangent function for test data

As we can see in Fig. 12, the results are in a perfect match with each other. For more investigation, the error diagram between these two layers is shown in Fig.13.

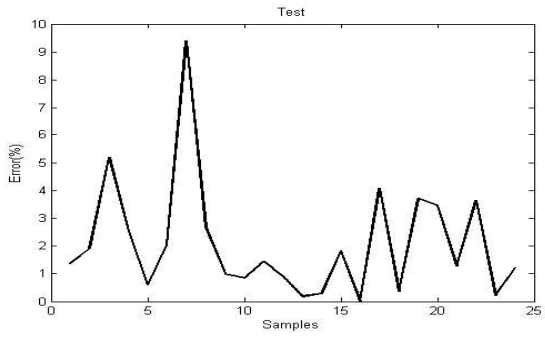


Fig. 13. The error of the two-layer network with tangent function for test data

It is obvious that the maximum error in this method has reached 10% which is very accurate and proper.

The diagram of the correlation coefficient of testing and training data with real data is shown in Fig. 14.

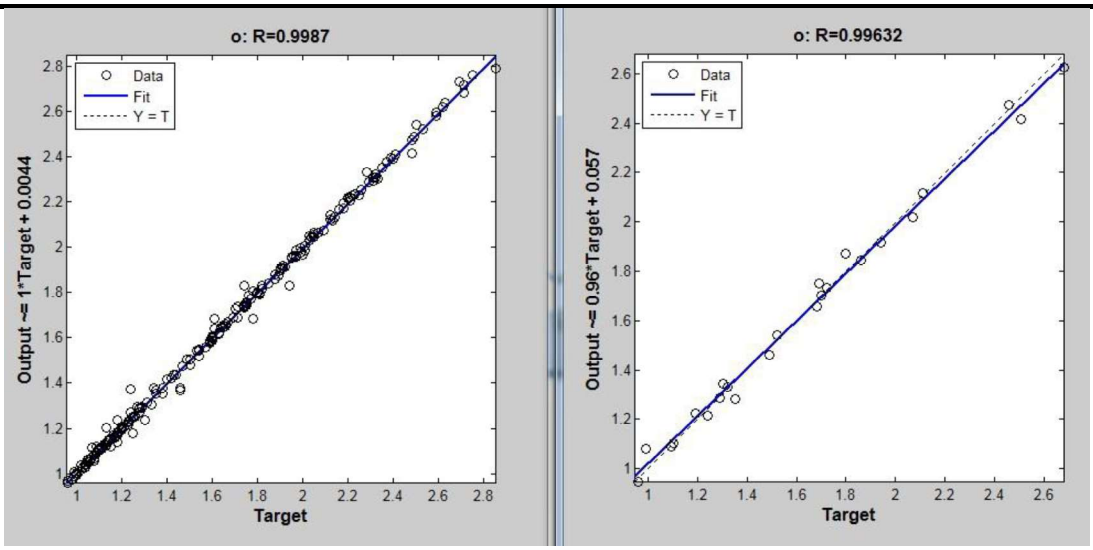


Fig. 14. The correlation coefficient of the two-layer network with tangent function for training and test data

It can be seen that the correlation coefficient for training and test data is very close to 1. The error percent for the training and testing data and also the processing time of the two-layer MLP network are shown below.

It is clear that the total error percentage for the training and testing data has decreased versus the one-layer network. Also, the processing time of the two-layer network is three times longer than the single-layer network. It is important to note that the accuracy of the single-layer and two-layer networks are so close, but the structure of the two-layer network is complex and the processing time is three times longer than that of the single-layer network. So, if we consider both processing time and training quality, the single-layer network is acceptable. Thus, in the rest of our research, the single-layer network is used and we try to optimize the network structure from the perspectives of neuron function and neuron number.

3.2.2. Neuron function effect

This section investigates the effect of neuron numbers. To this end, in a single layer with 10 neuron networks, we will use a logarithmic function instead of the tangent function. The structure of this network is shown in Fig. 15.

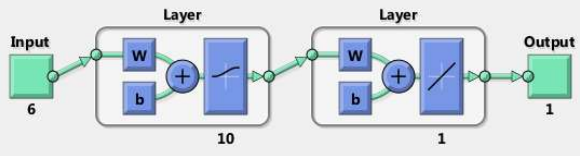


Fig. 15. A single layer network with 10 neurons in each layer and logarithmic function

In this case, output diagram for training data is also compared with real data.

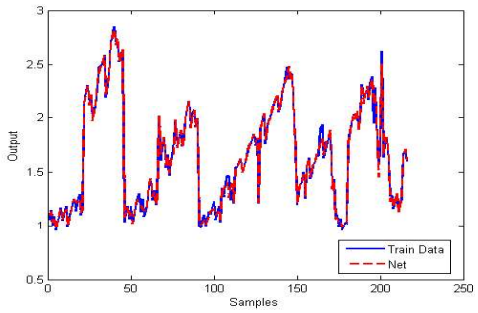


Fig. 16. The output of the single-layer network with logarithmic function for the training data

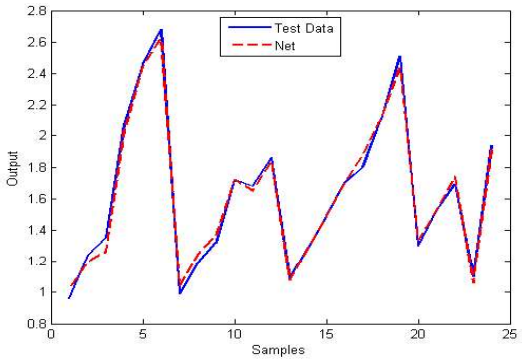


Fig. 17. The output of the single-layer network with logarithmic function for the testing data

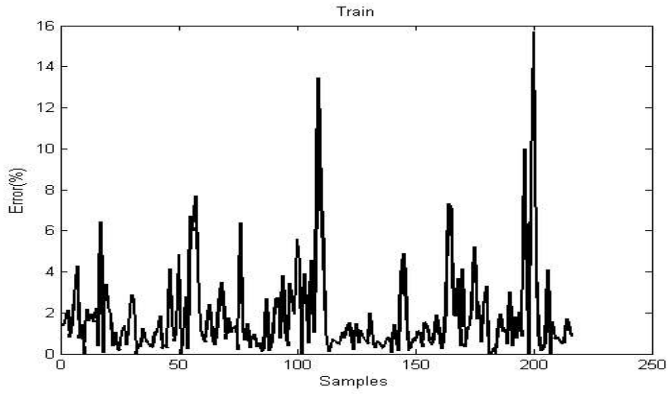


Fig. 18. The error of the single-layer network with logarithmic function for the training samples with LM learning method

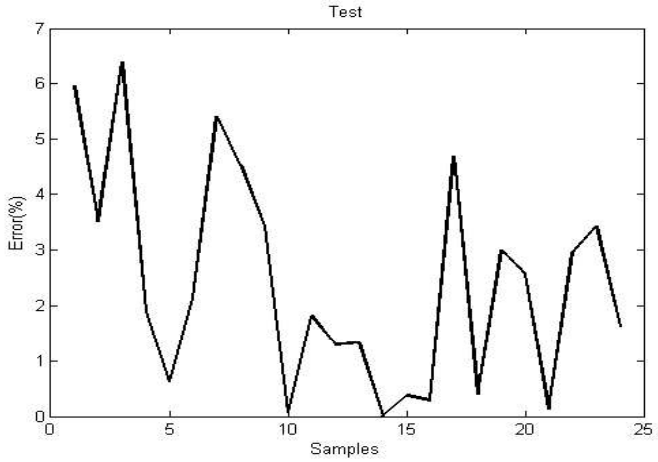


Fig. 19. The error of the single-layer network with logarithmic function for the testing data

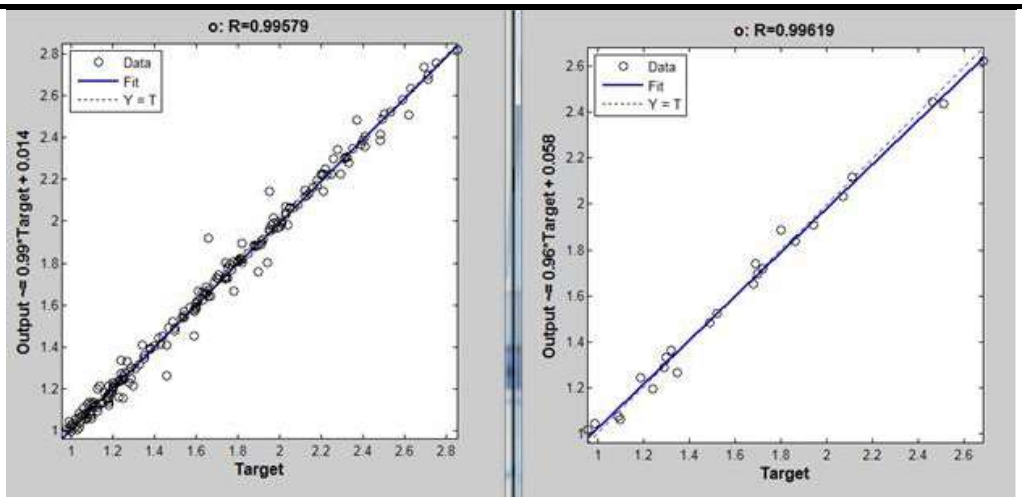


Fig. 20. The correlation coefficient of the single-layer network with logarithmic function for the training and testing data

3.2.3. Neurons number effect

This section explores the effect of neuron numbers on the efficiency of the MLP network. Here, a single-layer network with a tangent function is considered. The number of neurons varies from 3 to 15 and the value of sum square error for the training and testing data is shown in Fig. 21.

Figure 21 shows that the number of neurons has a strong effect on training quality. For example, if the number of neurons is considered to b 7 or 15, the error of the network

for testing data will be too much. Figure 22 displays the correlation coefficient changes between the real data and the training and testing data according to neuron numbers. As can be seen for the neuron number of 7 and 15 which has a higher error, the amount of correlation coefficient is lower.

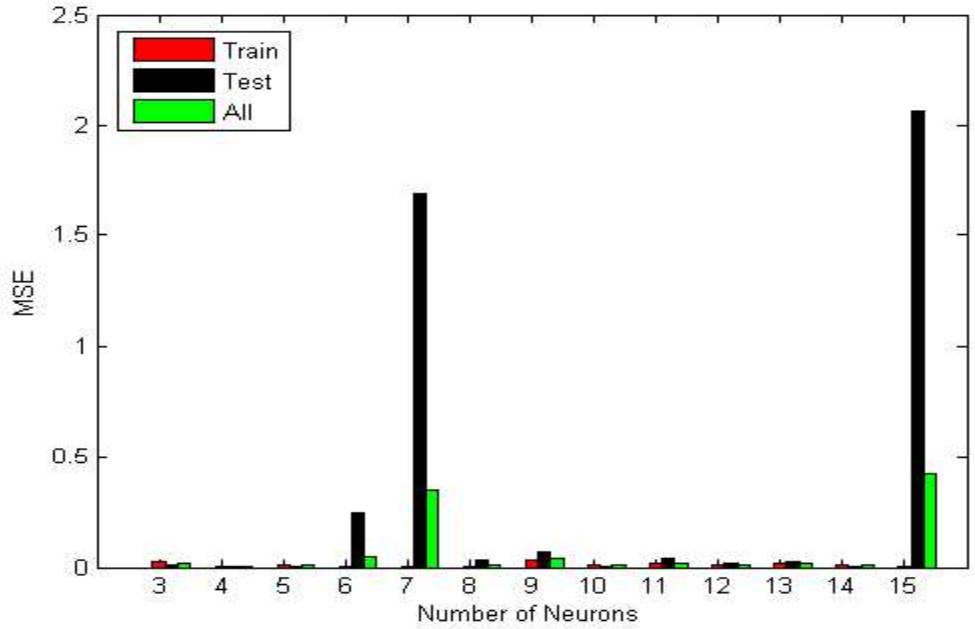


Fig. 21. The effect of neuron number on the network error for the training and testing data and the whole dataset

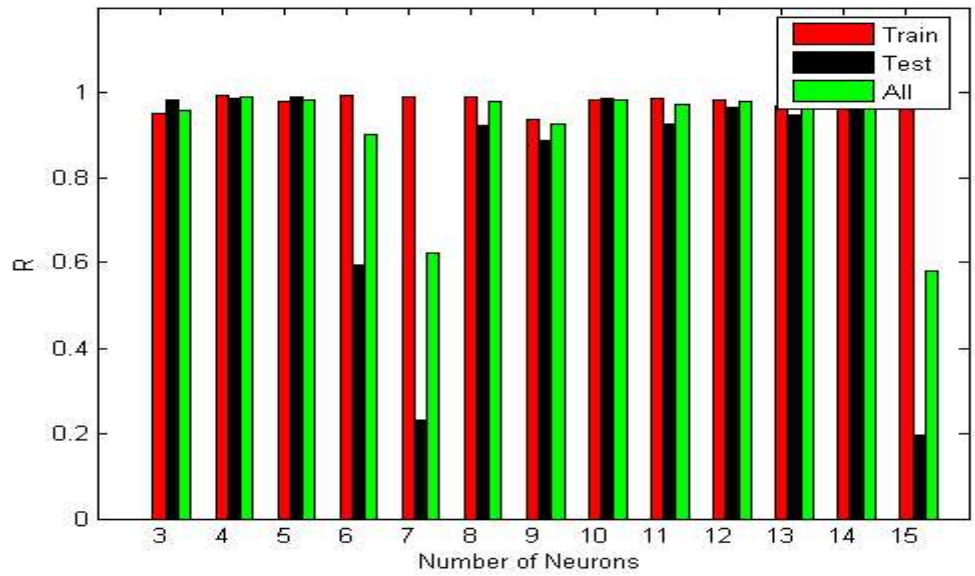


Fig. 22. The effect of neuron numbers on the network correlation coefficient for the training and testing data and the whole dataset

3.2.4. ANFIS network

In this section, we predict the data by Adaptive Neuro Fuzzy Inference System (ANFIS). ANFIS is a neural network that is based on Takagi-Sugeno fuzzy logic. This method was proposed in 1990 and is a combination of fuzzy logic and neural network basics. One of the main drawbacks of ANFIS is that as the number of inputs increases, it becomes exponentially more time-consuming to solve it because of the increase in fuzzy membership

functions and rules. In this research, we have six inputs with 240 samples as inputs. A system with these inputs is too complicated to use the ANFIS method because of defining membership function and fuzzy rules for each input by using fuzzy logic. So, first of all, we look for a way to reduce the number of data with a minimum loss of information. The best method is to analyze the main members. This is a statistical method that converts some observations into linearly independent data by vertical conversion.

The output of the ANFIS model for the training and testing data and also their errors and their correlation coefficient diagrams are shown in Figs. 23-28.

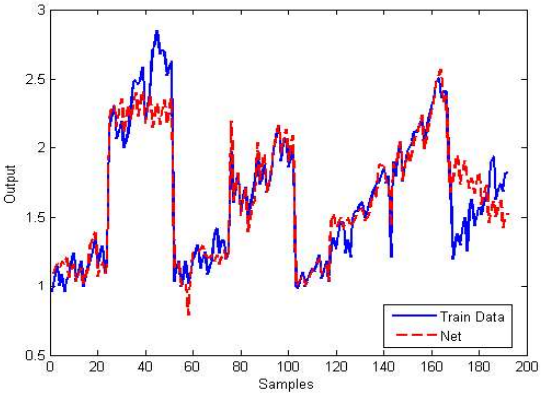


Fig. 23. The output of the ANFIS model for the training data

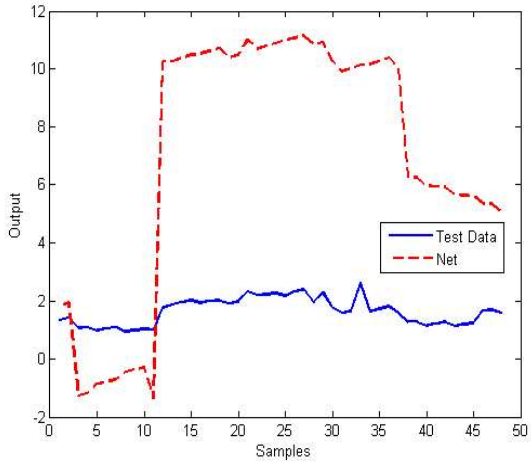


Fig. 24. The output of the ANFIS model for the testing data

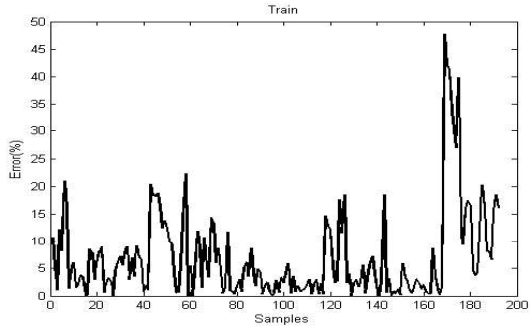


Fig. 25. The error of the ANFIS model for the training data

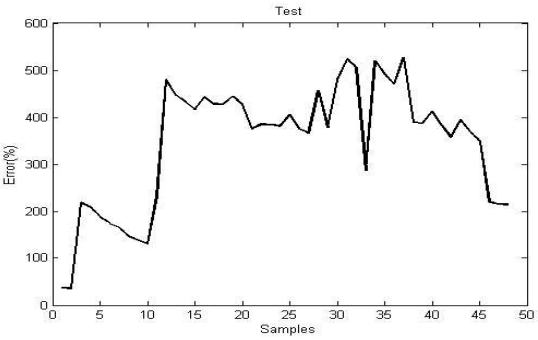


Fig. 26. The error of the ANFIS model for the testing data

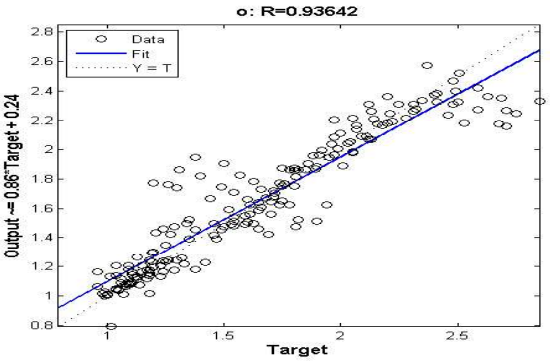


Fig. 27. The correlation coefficient of the ANFIS model for the training data

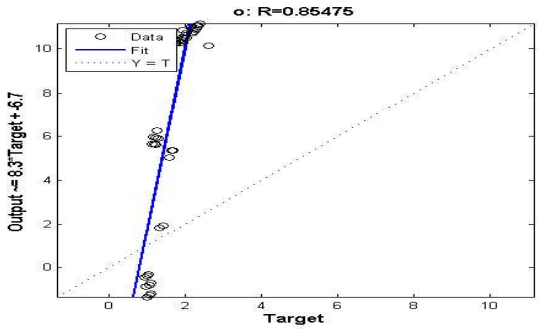


Fig. 28. The correlation coefficient of the ANFIS model for the testing data

It can be seen in Figs. 23-28 that the model is not as proper as the neural network for the training and testing data.

4. Conclusion

In this study, it is obvious that the accuracy of the two-layer and single-layer networks is so close but the structure of the two-layer network is very complex and its processing time is three times longer than the single-layer network. So, if we consider both processing time and learning quality parameters, the single-layer network is acceptable. So, a proper network in this research is a single-layer network and after investigating the network structure based on neuron number and optimizing neuron function, 10 neurons are appropriate from the quality point of view. So, the proper network is a single-layer network with 10 neurons and a tangent or logarithmic function in neurons. From the comparison of the MLP network and ANFIS network, it is revealed that the ANFIS network is not as good as the MLP network from the perspectives of training quality and processing time. So, the proper and acceptable network is a single-layer network with 10 neurons and a tangent or logarithmic algorithm in neurons. We suggest other researchers investigate other network training methods and compare the obtained results with the results of this research.

5. References

1. Metaxiotis , Kostas & John Psarras (2004) "The Contribution of Neural networks and genetic algoritms to business decision support "Management decision , vol 42 ,no .2, Emerald group publishing limited , pp. 229.242
2. Curry , B & L. Moutinho (1993) "Neural Network in marketing : Modelling consumer Responses to Advertising Stimuli "European Journal of marketing , vol 27 , no . 7 , MCB university press , pp 5. 20
3. Wray , B , A. palmer & D. Bejou (1994) " Using Neural Network Analysis to evaluate Buyer Seller Relationships " European Journal of Marketing , vol 28 , no. 10 , MCB university press , pp 32.48
4. Venugopal .V & W. Beats (1994) " Neural networks and Statistical Techniques in marketing research " Marketing intelligence & planning , vol 12 , no. 7 , MCB university press , pp 30 . 38
5. Davies , F , L . Moutinho & B . Curry (1996) " ATM user attitudes : a neural network analysis " , marketing intelligence & planning , vol 14 , no . 2 , MCB university press , pp 26