European Journal of Business and Management
ISSN 2222-1905 (Paper) ISSN 2222-2839 (Online)
Vol.5, No.31, 2013
www.iiste.org

# Engineering Project Management Modeling Using Artificial Neural Networks

Jamal M. Assbeihat, PhD
Faculty of Engineering Technology - Albalqa Applied University - Jordan
Jordan- Amman- P.O. Box (962800) Email: assbeihat@hotmail.com

**Abstract**
Performance evaluation of the comprehensive management level of engineering projects is advantageous case of study. Benefited from constructive and fluctuant of artificial neural networks (ANN) and based on their self-study, self-adjustment and nonlinear mapping (activation) function of the ANN inputs to outputs the performance evaluation model of engineering project management was established. Compared with conventional method, the influence of human factor is eliminated, thus the correctness of the measured results is increased. Different model structures were discussed with different ANN parameters and satisfactory results were concluded giving a new approach to evaluate the engineering project management.
**Keywords:** ANN structure, training rate, training time, activation function, performance evaluation**.**

## 1. Introduction and Background Theory

Project management is the focal task of objective evaluation for the management of completed projects and it needs systematic analysis and study. It helps for some suggestions for the future management and improving decision-making levels. Scientific and rational project management performance evaluation is conducive to improve the level of integrated management. At present the fields of academia and engineering had been achieved some results on this issue. On the basis of fuzzy theory, a fuzzy integrative evaluation model of engineering management performance evaluation is developed by Cai, Zhou and Ye (2002) and Zhang (2007). Besides, main object method is used for project management performance evaluation (Irland ,1985) . However, the relationship between index systems of project management performance evaluation are non-linear, it is difficult to determine the model to express. And the subjectivity of the evaluation process is increased when specialists are required to determine the index weight. So there are some drawbacks in the traditional evaluation model. Artificial neural network (ANN) is constrictive and fluctuant of transformation and has self-study, self-adjustment and nonlinear activation functions of neural network which has made certain research achievements in the field of pattern recognition ( Brian and Szu, 1994). Project management performance evaluation also belongs to pattern recognition, thus this paper tried to set a model using wavelet neural network model, with a view to produce good results.

The initial step of building a model for engineering project management is to set the architecture of ANN and we mean by the architecture is to set the input layer neurons, the hidden layer neurons**,** and the output layer neuron**.**

For project management modeling developed by Cai, Zhou and Ye (2002), and Zhang (2007), we can use the following inputs which are used widely by Brian and Szu 1994; Zhang and Benvenise 1992, and, Zhao; Fu and Xing ( 2007) for engineering project management evaluation: Constructive period advance rate, cost saving rate, quality control score, security control. These parameters (inputs) can be used to obtain the value of performance appraisal as shown in figure 1.

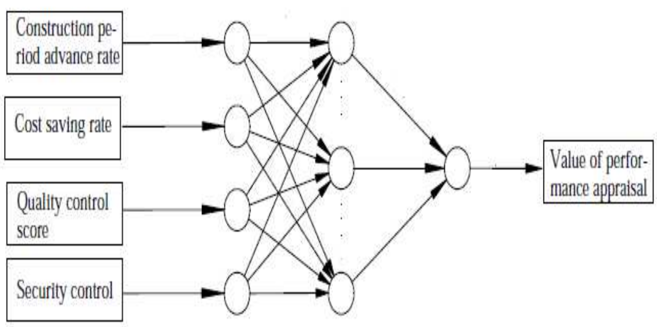


Figure 1: Engineering project management model

## 2. Project management model description

The proposed model is based on ANN and it is consisted of several layers, mentioned by Matee Serearuno and Tony Holden, (2001) and Liangtsan and Huizhu, (2000): input layer, one or more hidden layers, and one output layer. Different works (Liangtsan and Huizhu, 2000; Hsein-Ching et.al.,2002, and Farooq and Datta, 2003) show that the number of neurons in the input layer depends on the number of parameters used in project management evaluation, where the number of hidden layers and the number of neurons in each hidden layer depends on the desired optimization results.

Each proposed model needs selected values of input parameters which can be selected randomly and can be used to train the model using target calculated outputs, table 1 shows the selected input data.

Table 1: Learning samples of project management performance evaluation model

| Construction period advance rate | Cost saving rate | Quality control score | Security control | Target |
|---|---|---|---|---|
| 0.6555 | 0.7828 | 99.8274 | 0.0003 | 1 |
| 0.8906 | 0.7017 | 86.1415 | 0.0010 | 1 |
| 0.8207 | 0.5451 | 91.5090 | 0.0008 | 1 |
| 0.4573 | 0.3263 | 87.3103 | 0.0009 | 1 |
| 0.3033 | 0.9687 | 89.8753 | 0.0005 | 1 |
| 0.1064 | 0.0539 | 78.6508 | 0.0019 | 2 |
| 0.0799 | 0.0347 | 78.4761 | 0.0019 | 2 |
| 0.0836 | 0.0338 | 78.2145 | 0.0019 | 2 |
| 0.0848 | 0.0538 | 80.7094 | 0.0020 | 2 |
| 0.0763 | 0.0528 | 75.7117 | 0.0012 | 2 |
| 0.0335 | 0.0194 | 72.6387 | 0.0025 | 3 |
| 0.0012 | 0.0191 | 74.4602 | 0.0022 | 3 |
| 0.0456 | 0.0238 | 66.7535 | 0.0021 | 3 |
| 0.0350 | 0.0056 | 71.0947 | 0.0028 | 3 |
| 0.0195 | 0.0065 | 67.3856 | 0.0023 | 3 |
| -0.1334 | -0.0119 | 24.2475 | 0.1252 | 4 |
| -0.9901 | -0.1962 | 33.3788 | 0.4619 | 4 |
| -0.2843 | -0.8583 | 47.2422 | 0.6146 | 4 |
| -0.9360 | -0.2155 | 41.1326 | 0.8282 | 4 |
| -0.7105 | -0.9060 | 64.6641 | 0.4198 | 4 |

After selecting the model and choosing ANN structure we have to define the behavior of the model implementation, thus we have to select the activation function and the training method.

Activation function is used to conduct the neuron output and it can be calculated as follows(Diaa Eldin, et.al. 2003; Dionisis et. al. 2004; Stefan and Sagar 1999) :

$$\beta_i = \frac{1-e^{-ky}}{1+e^{-ky}} \qquad (1)$$

Where: y is the summation of products; k is relative to impact laws of the evaluation.

## 3. Steps of Model Implementation

Through the study of network optimization indexes (Zhao, Fu and Xing 2007; Chang- Tien and Lily 2004 ), set amendment of the network and wavelet function parameters by error back propagation algorithm, then reach the most optimal learning effects gradually. Learning algorithm steps as followed with application of Matlab7.0 programming:

Step 1: Set the input samples and define the target outputs.
Step 2: Define the model structure.
Step 3: Initialize the ANN weights, training rate, end expecting error (goal), and the number of training iterations ( Zhao, Fu and Xing 2007; Dionisis et. al. 2004).
Step 4: Define the activation function for each layer.
Step 5: Model training which can be applied in several iterations each of them contains 2 steps: the feed forward step which calculates the output of each neuron according to the inputs, the input weights and activation function, and back word steps which modifies compares the target with the calculated output and regarding the result of comparing (error) adjust the weights.
Step 6: When a network error is less than a pre-determined value or learning steps of maximum training value is reached, ANN learning is stopped, otherwise return to the third step to repeat training until the expected output of the network is generated.

**Model implementation**

The following mat lab code was used to implement the various model structures with various parameters:

```
clc
cpar=[0.6555;0.8906;0.8207;0.4573;0.3033;0.1064;0.0799;0.0836;0.0848;0.0763;
   0.0335;0.0012;0.0456;0.0350;0.0195;-0.1334;-0.9901;-0.2843;-0.9360;-0.7105];
csr=[0.7828;0.7017;0.5451;0.3263;0.9687;0.0539;0.0347;0.0338;0.0538;0.0528;
   0.0194;0.0191;0.0238;0.0056;0.0065;-0.0119;-0.1962;-0.8583;-0.2155;-0.9060];
qcs=[99.8274;86.1415;91.5090;87.3103;89.8753;78.6508;78.4761;78.2145;80.7094;75.7117;
   72.6387;74.4602;66.7535;71.0947;67.3856;24.2475;33.3788;47.2422;41.1326;64.6641];
sc=[0.0003;0.0010;0.0008;0.0009;0.0005;0.0019;0.0019;0.0019;0.0020;0.0012;0.0025;
   0.0022;0.0021;0.0028;0.0023;0.1252;0.4619;0.6146;0.8282;0.4198];
PP=[cpar csr qcs sc];
PP=PP';
target=[1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4];

%backpropagation
net2 = newff(minmax(PP),[4 1],{ 'logsig' 'purelin'}, 'trainlm');
net2.trainParam.epochs = 10000; % training stops if epochs reached
net2.trainParam.goal = 10e-9;
net2.trainParam.show = 10; % plot the performance function at every epoch
net2.trainParam.lr=0.0001;
tic
net2 = train(net2, PP, target);
toc
c = sim(net2, PP);
```

**Case 1:**

The model structure is consisted of one input layer, one hidden layer with 4 neurons and one output layer with one neuron, training rate=0.0001, the activation function of the hidden layer is logsig..
Table 2 shows the result of model implementation in this case.

 

Table 2: case 1 results.

| Construction period advance rate | Cost saving rate | Quality control score | Security control | Target | Modeling output | Error |
|---|---|---|---|---|---|---|
| 0.6555 | 0.7828 | 99.8274 | 0.0003 | 1 | 1.0005 | 0.049975 |
| 0.8906 | 0.7017 | 86.1415 | 0.0010 | 1 | 1.0266 | 2.591077 |
| 0.8207 | 0.5451 | 91.5090 | 0.0008 | 1 | 0.9746 | 2.606197 |
| 0.4573 | 0.3263 | 87.3103 | 0.0009 | 1 | 0.9999 | 0.010001 |
| 0.3033 | 0.9687 | 89.8753 | 0.0005 | 1 | 1 | 0 |
| 0.1064 | 0.0539 | 78.6508 | 0.0019 | 2 | 1.9868 | 0.664385 |
| 0.0799 | 0.0347 | 78.4761 | 0.0019 | 2 | 2.0237 | 1.171122 |
| 0.0836 | 0.0338 | 78.2145 | 0.0019 | 2 | 2.0127 | 0.630993 |
| 0.0848 | 0.0538 | 80.7094 | 0.0020 | 2 | 1.9886 | 0.573268 |
| 0.0763 | 0.0528 | 75.7117 | 0.0012 | 2 | 1.9946 | 0.270731 |
| 0.0335 | 0.0194 | 72.6387 | 0.0025 | 3 | 2.9778 | 0.745517 |
| 0.0012 | 0.0191 | 74.4602 | 0.0022 | 3 | 2.9996 | 0.013335 |
| 0.0456 | 0.0238 | 66.7535 | 0.0021 | 3 | 2.9415 | 1.988781 |
| 0.0350 | 0.0056 | 71.0947 | 0.0028 | 3 | 2.9912 | 0.294196 |
| 0.0195 | 0.0065 | 67.3856 | 0.0023 | 3 | 3.0861 | 2.789929 |
| -0.1334 | -0.0119 | 24.2475 | 0.1252 | 4 | 3.9992 | 0.020004 |
| -0.9901 | -0.1962 | 33.3788 | 0.4619 | 4 | 3.9992 | 0.020004 |
| -0.2843 | -0.8583 | 47.2422 | 0.6146 | 4 | 3.9992 | 0.020004 |
| -0.9360 | -0.2155 | 41.1326 | 0.8282 | 4 | 3.9992 | 0.020004 |
| -0.7105 | -0.9060 | 64.6641 | 0.4198 | 4 | 3.9992 | 0.020004 |

Average error=0.724976

**Case 2:**

The model structure is consisted of one input layer, one hidden layer with 4 neurons and one output layer with one neuron, training rate=0.0001, the activation function of the hidden layer is purelin..
Table 3 shows the result of model implementation in this case.

Table 3: case 2 results.

| Construction period advance rate | Cost saving rate | Quality control score | Security control | Target | Modeling output | Error |
|---|---|---|---|---|---|---|
| 0.6555 | 0.7828 | 99.8274 | 0.0003 | 1 | 0.6467 | 54.6312 |
| 0.8906 | 0.7017 | 86.1415 | 0.0010 | 1 | 0.9897 | 1.040719 |
| 0.8207 | 0.5451 | 91.5090 | 0.0008 | 1 | 1.0174 | 1.710242 |
| 0.4573 | 0.3263 | 87.3103 | 0.0009 | 1 | 1.5652 | 36.1104 |
| 0.3033 | 0.9687 | 89.8753 | 0.0005 | 1 | 0.9822 | 1.812258 |
| 0.1064 | 0.0539 | 78.6508 | 0.0019 | 2 | 2.2883 | 12.59887 |
| 0.0799 | 0.0347 | 78.4761 | 0.0019 | 2 | 2.3338 | 14.30285 |
| 0.0836 | 0.0338 | 78.2145 | 0.0019 | 2 | 2.2398 | 10.70631 |
| 0.0848 | 0.0538 | 80.7094 | 0.0020 | 2 | 2.3956 | 16.51361 |
| 0.0763 | 0.0528 | 75.7117 | 0.0012 | 2 | 2.5436 | 21.37128 |
| 0.0335 | 0.0194 | 72.6387 | 0.0025 | 3 | 2.509 | 19.56955 |
| 0.0012 | 0.0191 | 74.4602 | 0.0022 | 3 | 2.7083 | 10.77059 |
| 0.0456 | 0.0238 | 66.7535 | 0.0021 | 3 | 2.6016 | 15.31365 |
| 0.0350 | 0.0056 | 71.0947 | 0.0028 | 3 | 2.9912 | 0.294196 |
| 0.0195 | 0.0065 | 67.3856 | 0.0023 | 3 | 2.7211 | 10.24953 |
| -0.1334 | -0.0119 | 24.2475 | 0.1252 | 4 | 4.058 | 1.429276 |
| -0.9901 | -0.1962 | 33.3788 | 0.4619 | 4 | 4.3044 | 7.071833 |
| -0.2843 | -0.8583 | 47.2422 | 0.6146 | 4 | 4.0076 | 0.18964 |
| -0.9360 | -0.2155 | 41.1326 | 0.8282 | 4 | 3.8761 | 3.196512 |
| -0.7105 | -0.9060 | 64.6641 | 0.4198 | 4 | 3.8844 | 2.976007 |

Average error= 12.09293

**Case 3:**

The model structure is consisted of one input layer, one hidden layer with 4 neurons and one output layer with one neuron, training rate=0.001, the activation function of the hidden layer is logsig..

Table 4 shows the result of model implementation in this case.

 

Table 4: case 3 results.

| Construction period advance rate | Cost saving rate | Quality control score | Security control | Target | Modeling output | Error |
|---|---|---|---|---|---|---|
| 0.6555 | 0.7828 | 99.8274 | 0.0003 | 1 | 0.6467 | 54.6312 |
| 0.8906 | 0.7017 | 86.1415 | 0.0010 | 1 | 0.9897 | 1.040719 |
| 0.8207 | 0.5451 | 91.5090 | 0.0008 | 1 | 1.0174 | 1.710242 |
| 0.4573 | 0.3263 | 87.3103 | 0.0009 | 1 | 1.5652 | 36.1104 |
| 0.3033 | 0.9687 | 89.8753 | 0.0005 | 1 | 0.9822 | 1.812258 |
| 0.1064 | 0.0539 | 78.6508 | 0.0019 | 2 | 2.2883 | 12.59887 |
| 0.0799 | 0.0347 | 78.4761 | 0.0019 | 2 | 2.3338 | 14.30285 |
| 0.0836 | 0.0338 | 78.2145 | 0.0019 | 2 | 2.2398 | 10.70631 |
| 0.0848 | 0.0538 | 80.7094 | 0.0020 | 2 | 2.3956 | 16.51361 |
| 0.0763 | 0.0528 | 75.7117 | 0.0012 | 2 | 2.5436 | 21.37128 |
| 0.0335 | 0.0194 | 72.6387 | 0.0025 | 3 | 2.509 | 19.56955 |
| 0.0012 | 0.0191 | 74.4602 | 0.0022 | 3 | 2.7083 | 10.77059 |
| 0.0456 | 0.0238 | 66.7535 | 0.0021 | 3 | 2.6016 | 15.31365 |
| 0.0350 | 0.0056 | 71.0947 | 0.0028 | 3 | 2.9912 | 0.294196 |
| 0.0195 | 0.0065 | 67.3856 | 0.0023 | 3 | 2.7211 | 10.24953 |
| -0.1334 | -0.0119 | 24.2475 | 0.1252 | 4 | 4.058 | 1.429276 |
| -0.9901 | -0.1962 | 33.3788 | 0.4619 | 4 | 4.3044 | 7.071833 |
| -0.2843 | -0.8583 | 47.2422 | 0.6146 | 4 | 4.0076 | 0.18964 |
| -0.9360 | -0.2155 | 41.1326 | 0.8282 | 4 | 3.8761 | 3.196512 |
| -0.7105 | -0.9060 | 64.6641 | 0.4198 | 4 | 3.8844 | 2.976007 |

Average error= 0.7364

**Case 4:**

The model structure is consisted of one input layer, two hidden layer with 4 neurons each and one output layer with one neuron, training rate=0.0001, the activation function of the hidden layer is losig.. Table 5 shows the result of model implementation in this case.

 

Table 5: Ccase 4 results.

| Construction period advance rate | Cost saving rate | Quality control score | Security control | Target | Modeling output | Error |
|---|---|---|---|---|---|---|
| 0.6555 | 0.7828 | 99.8274 | 0.0003 | 1 | 1.0001 | 0.009999 |
| 0.8906 | 0.7017 | 86.1415 | 0.0010 | 1 | 1.0001 | 0.009999 |
| 0.8207 | 0.5451 | 91.5090 | 0.0008 | 1 | 1.0001 | 0.009999 |
| 0.4573 | 0.3263 | 87.3103 | 0.0009 | 1 | 1 | 0 |
| 0.3033 | 0.9687 | 89.8753 | 0.0005 | 1 | 0.9997 | 0.030009 |
| 0.1064 | 0.0539 | 78.6508 | 0.0019 | 2 | 2 | 0 |
| 0.0799 | 0.0347 | 78.4761 | 0.0019 | 2 | 1.9998 | 0.010001 |
| 0.0836 | 0.0338 | 78.2145 | 0.0019 | 2 | 2.0002 | 0.009999 |
| 0.0848 | 0.0538 | 80.7094 | 0.0020 | 2 | 2 | 0 |
| 0.0763 | 0.0528 | 75.7117 | 0.0012 | 2 | 2 | 0 |
| 0.0335 | 0.0194 | 72.6387 | 0.0025 | 3 | 3 | 0 |
| 0.0012 | 0.0191 | 74.4602 | 0.0022 | 3 | 3 | 0 |
| 0.0456 | 0.0238 | 66.7535 | 0.0021 | 3 | 3 | 0 |
| 0.0350 | 0.0056 | 71.0947 | 0.0028 | 3 | 3 | 0 |
| 0.0195 | 0.0065 | 67.3856 | 0.0023 | 3 | 3 | 0 |
| -0.1334 | -0.0119 | 24.2475 | 0.1252 | 4 | 4 | 0 |
| -0.9901 | -0.1962 | 33.3788 | 0.4619 | 4 | 4 | 0 |
| -0.2843 | -0.8583 | 47.2422 | 0.6146 | 4 | 3.9999 | 0.0025 |
| -0.9360 | -0.2155 | 41.1326 | 0.8282 | 4 | 4 | 0 |
| -0.7105 | -0.9060 | 64.6641 | 0.4198 | 4 | 4.0001 | 0.0025 |

Average error= 0.00425

**Case 5:**

The model structure is consisted of one input layer, one hidden layer with 8 neurons each and one output layer with one neuron, training rate=0.0001, the activation function of the hidden layer is losig.. Table 6 shows the result of model implementation in this case.

 

Table 6: Ccase 5 results.

| Construction period advance rate | Cost saving rate | Quality control score | Security control | Target | Modeling output | Error |
|---|---|---|---|---|---|---|
| 0.6555 | 0.7828 | 99.8274 | 0.0003 | 1 | 1 | 0 |
| 0.8906 | 0.7017 | 86.1415 | 0.0010 | 1 | 1.0001 | 0.009999 |
| 0.8207 | 0.5451 | 91.5090 | 0.0008 | 1 | 0.9999 | 0.010001 |
| 0.4573 | 0.3263 | 87.3103 | 0.0009 | 1 | 1 | 0 |
| 0.3033 | 0.9687 | 89.8753 | 0.0005 | 1 | 1 | 0 |
| 0.1064 | 0.0539 | 78.6508 | 0.0019 | 2 | 2 | 0 |
| 0.0799 | 0.0347 | 78.4761 | 0.0019 | 2 | 2 | 0 |
| 0.0836 | 0.0338 | 78.2145 | 0.0019 | 2 | 2.0001 | 0.005 |
| 0.0848 | 0.0538 | 80.7094 | 0.0020 | 2 | 2 | 0 |
| 0.0763 | 0.0528 | 75.7117 | 0.0012 | 2 | 2 | 0 |
| 0.0335 | 0.0194 | 72.6387 | 0.0025 | 3 | 3 | 0 |
| 0.0012 | 0.0191 | 74.4602 | 0.0022 | 3 | 3 | 0 |
| 0.0456 | 0.0238 | 66.7535 | 0.0021 | 3 | 3 | 0 |
| 0.0350 | 0.0056 | 71.0947 | 0.0028 | 3 | 3 | 0 |
| 0.0195 | 0.0065 | 67.3856 | 0.0023 | 3 | 3 | 0 |
| -0.1334 | -0.0119 | 24.2475 | 0.1252 | 4 | 4 | 0 |
| -0.9901 | -0.1962 | 33.3788 | 0.4619 | 4 | 4 | 0 |
| -0.2843 | -0.8583 | 47.2422 | 0.6146 | 4 | 4.0003 | 0.007499 |
| -0.9360 | -0.2155 | 41.1326 | 0.8282 | 4 | 4 | 0 |
| -0.7105 | -0.9060 | 64.6641 | 0.4198 | 4 | 3.9997 | 0.007501 |

Average error=0.002

**Case 6:**

The model structure is consisted of one input layer, one hidden layer with 16 neurons each and one output layer with one neuron, training rate=0.0001, the activation function of the hidden layer is losig.. Table 7 shows the result of model implementation in this case.

 

Table 7: Case 6 results.

| Construction period advance rate | Cost saving rate | Quality control score | Security control | Target | Modeling output | Error |
|---|---|---|---|---|---|---|
| 0.6555 | 0.7828 | 99.8274 | 0.0003 | 1 | 1 | 0 |
| 0.8906 | 0.7017 | 86.1415 | 0.0010 | 1 | 1.0002 | 0.019996 |
| 0.8207 | 0.5451 | 91.5090 | 0.0008 | 1 | 0.9996 | 0.040016 |
| 0.4573 | 0.3263 | 87.3103 | 0.0009 | 1 | 1 | 0 |
| 0.3033 | 0.9687 | 89.8753 | 0.0005 | 1 | 1.0001 | 0.009999 |
| 0.1064 | 0.0539 | 78.6508 | 0.0019 | 2 | 2 | 0 |
| 0.0799 | 0.0347 | 78.4761 | 0.0019 | 2 | 2 | 0 |
| 0.0836 | 0.0338 | 78.2145 | 0.0019 | 2 | 2 | 0 |
| 0.0848 | 0.0538 | 80.7094 | 0.0020 | 2 | 2 | 0 |
| 0.0763 | 0.0528 | 75.7117 | 0.0012 | 2 | 2 | 0 |
| 0.0335 | 0.0194 | 72.6387 | 0.0025 | 3 | 3 | 0 |
| 0.0012 | 0.0191 | 74.4602 | 0.0022 | 3 | 3 | 0 |
| 0.0456 | 0.0238 | 66.7535 | 0.0021 | 3 | 3 | 0 |
| 0.0350 | 0.0056 | 71.0947 | 0.0028 | 3 | 3 | 0 |
| 0.0195 | 0.0065 | 67.3856 | 0.0023 | 3 | 3 | 0 |
| -0.1334 | -0.0119 | 24.2475 | 0.1252 | 4 | 4 | 0 |
| -0.9901 | -0.1962 | 33.3788 | 0.4619 | 4 | 3.9999 | 0.0025 |
| -0.2843 | -0.8583 | 47.2422 | 0.6146 | 4 | 4 | 0 |
| -0.9360 | -0.2155 | 41.1326 | 0.8282 | 4 | 4 | 0 |
| -0.7105 | -0.9060 | 64.6641 | 0.4198 | 4 | 4.0001 | 0.0025 |

Average error=0.003751

Table 6 summarizes the performance parameters for different model structures.

Table 6: the performance parameters for different model structures

| Model structure | Training iterations | Training time(seconds) | Average error |
|---|---|---|---|
| 4 | 827 | 6.260000 | 0.724976 |
| ***8*** | ***795*** | ***6.085000*** | 0.002 |
| 16 | 1723 | 17.144000 | 0.003751 |
| 4X4 | 502 | 4.713000 | 0.00425 |
| 8x8 | 842 | 9.927000 | 0.004 |
| 16x16 | 152 | 6.754000 | 0.0039 |

## 4. Conclusions

Started from the purpose and requirements of project management performance evaluation, using artificial neural network models which can describe the complex and nonlinear relationship, the following conclusions are obtained:

1) There is no need to determine the model weights by people using ANN models. They are set randomly and updated automatically.

2) We can achieve less modulation error by decreasing the training ratio.

3) It is better to use one hidden layer with number of neurons equal the double of inputs, here we can achieve the smallest error and minimum training time.

**References**

1) Brian T. and Szu H. (1994), “Adaptive wavelet classification of acoustic backscatter and imagery,” Optical Engineering, 7, pp. 2192-2202.

2) Cai S. H., Zhou M. Y. and Ye Z. C. ( 2002), “Fuzzy integrative evaluation method of management performance of engineering project,” Journal of YangZhou university (natural science), 5, pp. 57-60.

3) Chang-Tien Lu , Lily R. Liang, (2004),Wavelet fuzzy classification for detecting and tracking region outliers in meteorological data, Proceedings of the 12th annual ACM international workshop on Geographic information systems, Washington DC, USA, 12-13, (November, 2004) .

4) Diaa Eldin M. Nassar , Hany H. Ammar, (2003), A prototype automated dental identification system (ADIS), Proceedings of the 2003 annual national conference on Digital government research, , Boston, MA p.1-4, 18-21, (May, 2003)

5) Dionisis Kehagias , Andreas L. Symeonidis , Kyriakos C. (2004), Chatzidimitriou , Pericles A. Mitkas, Information agents cooperating with heterogenous data sources for customer-order management, Proceedings of the 2004 ACM symposium on Applied computing, Nicosia, Cyprus, 14-17. (March, 2004).

6) Farooq O., Datta S. (2003), Phoneme recognition using wavelet based features, Information Sciences—Informatics and Computer Science: An International Journal, v.150 n.1-2, p.5-15. , (March, 2003).

7) Hsien-Cheng Chang , David C. Kopaska-Merkel , Hui-Chuan Chen, (2002). Identification of lithofacies using Kohonen self-organizing maps, Computers & Geosciences, v.28 n.2, p.223-229. (March, 2002).

8) Ireland V. (1985). “The role of management actions in the cost, time and quality performance of high-rise commercial building projects,” Construction Management and Economics, 3, pp. 59-87.

9) Liangtsan G. Wu , Huizhu Lu, (2000). F3MCNN: a fuzzy minimum mean maximum clustering neural network, Proceedings of the 2000 ACM symposium on Applied computing, Como, Italy, p.10-14. (March, 2000)

10) Matee Serearuno , Tony Holden, (2001). A comparison in training time of the single and multiple-output MLP neural networks, Proceedings of the 2001 ACM symposium on Applied computing, Las Vegas, Nevada, United States, p.32-35, (March,2001).

11) Stefan Pittner , Sagar V. Kamarthi, (1999). Feature Extraction From Wavelet Coefficients for Pattern Recognition Tasks, IEEE Transactions on Pattern Analysis and Machine Intelligence, v.21 n.1, p.83-88, (January 1999)

12) Xin Li, (2002). Interpolation by ridge polynomials and its application in neural networks, Journal of Computational and Applied Mathematics, v.144 n.1-2, p.197-209,( July 2002).

13) Zhang L. X. “, (2007). Fuzzy comprehensive evaluation of the expressway project management performance evaluation,”Road traffic technology (applications) 5, pp. 169-171.

14) Zhang Q. H. and Benveniste A. (1992). “Wavelet networks,” IEEE Transactions on Neural Networds, 6, 889-898.

15) Zhao X. Y., Fu Q., and Xing Z. X., (2007) “Application of projection pursuit grade evaluation model in comprehensive evaluation of changes in soil quality,” Acta Pedologica Sinica. 1, pp. 164-168.