Journal of Civil Engineering and Urban Planning (2024)
Clausius Scientific Press, Canada
DOI: 10.23977/jceup.2024.060203
ISSN 2616-3969 Vol. 6 Num. 2

Research on Construction Project Valuation Based on Artificial Intelligence Technology

Jue Wang

School of Construction Engineering, Chengdu Vocational & Technical College of Industry, Chengdu, Sichuan, 610000, China

Keywords: Artificial Intelligence; BP Neural Network; PSO Optimization

Abstract: With the continuous development of computer science, intelligent optimization techniques have penetrated into various research fields. It can help solve the shortcomings of large error and long preparation time in the estimation of construction project cost. This study focuses on the application of artificial intelligence methods in the field of construction cost estimation. By utilizing the data fitting ability of neural networks, an artificial intelligence estimation model is established to predict construction cost estimates. The theoretical basis and basic principles of the BP neural network are elucidated, and the MATLAB software is used to validate its excellent function approximation capability. A construction cost estimation model based on PSO-optimized BP neural network is developed, and through MATLAB programming for sample training and testing, the results demonstrate that both models have errors within acceptable ranges. The establishment of the construction cost estimation model enables the prediction of engineering costs, proving the practical value of the model.

1. Introduction

With the continuous improvement of our country's economic level and national comprehensive strength, China is also facing many new situations and challenges. In response to these new circumstances, new reform measures have been introduced, with a significant amount of funds being used for asset investment, especially witnessing a substantial increase in investment in infrastructure construction. As the investment amount in national construction projects increases, the efficient allocation and rational use of funds naturally become crucial issues. In recent years, the construction industry in China has been developing rapidly, with the scale and quantity of engineering construction continuously increasing[1]. Estimating the cost of construction projects plays an increasingly prominent role as a key step in project management. The characteristics of construction projects include large investment amounts and long durations. Cost estimation of construction projects refers to estimating the investment amount of construction projects based on existing data, investment estimation methods, indicators, and experience during the project investment decision-making stage, serving as the primary basis for project investment decisions.

An important problem in the scope of China's fixed asset investment is the lack of control over investment in construction projects. This problem stems from various reasons, with a significant factor being a heavier focus on cost control during the construction phase rather than during the investment decision-making phase. The implementation of construction projects mainly consists of investment decision-making, project design, construction, and maintenance stages. While the investment in construction projects mainly occurs during the construction phase, it is the stages before construction that are crucial for controlling and saving project investments, gradually diminishing as the project progresses. The objectives set during the investment phase represent the maximum budget for the construction project, serving as the primary basis for cost control throughout the entire project implementation process. Therefore, the urgent need to address the rapid and accurate estimation of construction project costs is crucial, as this process will determine whether project engineering costs can be effectively guided and controlled. [2]

In 1996, Shao Liangbin and Gao Shulin established an appraisal system based on artificial neural networks. After verifying the feasibility of the system model, the model was applied in the shaft and roadway engineering of mining projects. Shen Ling utilized the strong learning function of neural networks, pioneering the application of artificial neural networks in real estate appraisal. Under the guidance of the model, she successfully generalized the relationship between real estate prices and their influencing factors. Zhang Pengtao and Yang Hong established a benchmark land price prediction model using BP neural networks, determining the relationship between benchmark land prices and their influencing factors, which to a certain extent ensures the practicality of benchmark land prices. Chen Zhiqin applied neural networks to the appraisal of transportation station projects, validating the practicality of the model through engineering examples[3].

This study focuses on the application of artificial intelligence methods in the field of architectural cost estimation. It establishes an engineering cost estimation model based on the PSO-optimized BP neural network parameters and tests the network's simulation capabilities using three simple functions (polynomial function, sine function, and exponential function), demonstrating the applicability of BP neural networks in architectural cost estimation. Finally, it utilizes the newff function in MATLAB software to build an architectural cost estimation model based on PSO-optimized BP neural network parameters and conducts tests to validate the model.

2. Optimization of BP neural network evaluation model based on PSO

2.1 BP neural network

2.1.1 Basic principle of BP neural network

The BP (back propagation) neural network is a typical feed forward network (multilayer perceptron is a type of back propagation network), consisting of an input layer, one or more hidden layers, and an output layer (Figure 1). Each layer contains several nodes representing neurons, with no connection between nodes on the same layer, and full connections between adjacent layers. Information flows through the input layer, various hidden layers, and finally reaches the output layer[4-8].

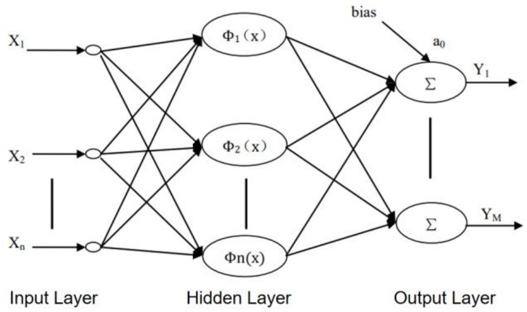


Figure 1: Structure of BP neural network

Given n input signals, where the input layer has n inputs represented by the vector $\mathrm{X} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$; there are h hidden layer nodes, with the hidden layer output vector $\mathrm{Y} = (y_1, y_2, \ldots, y_n)^{\mathrm{T}}$; the output layer consists of m nodes, with the network output vector $\mathrm{O} = (o_1, o_2, \ldots, o_n)^{\mathrm{T}}$, and the expected output vector is $\mathrm{D} = (d_1, d_2, \ldots, d_n)^{\mathrm{T}}$. The connection weight matrix from the input layer to the hidden layer is represented by $\mathrm{V} = (v_1, v_2, \ldots, v_n)^{\mathrm{T}}$, and the connection weight matrix from the hidden layer to the output layer is represented by $\mathrm{W} = (w_1, w_2, \ldots, w_n)^{\mathrm{T}}$.

The input and output of the j-th neuron node in the hidden layer can be represented as:

$$y_i = f(net_k)k = 1,2,\ldots,h \tag{1}$$

$$net_j = \sum_{i=1}^{h} v_{ij}\, x_i j = 1,2,\ldots,h \tag{2}$$

The input and output of the k-th neuron node in the output layer can be represented as:

$$o_k = f(net_k)k = 1,2,\ldots,m \tag{3}$$

$$net_k = \sum_{i=1}^{h} w_{jk}\, y_i k = 1,2,\ldots,m \tag{4}$$

In the BP neural network, $f\,(x)$ is the activation function, which mainly includes linear and nonlinear activation functions. The commonly used nonlinear activation function, the Sigmoid function, has two main types: the S-shaped Log-Sigmoid function transforms any input value into a number between [0,1] according to formula (5). The other type is the hyperbolic tangent function, Tan-Sigmoid function, which transforms any input value into a number between [-1,1] according to formula (6). The activation function of the output layer generally uses a linear function, allowing the output value to take any value.

$$f\,(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

$$f\,(x) = \frac{2}{1 + e^{-x}} - 1 = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{6}$$

2.1.2 Function fitting of BP neural network

The latest MATLAB Neural Network Toolbox contains almost all the new content related to neural network research, offering a variety of algorithms for various neural network models. When designing a BP neural network using MATLAB, key considerations include the network's topology (number of hidden layers and neurons in each layer), selection of activation functions for neurons, network initialization (connection weights and initial values), training parameter settings, normalization of training samples, and methods for importing sample data. In summary, the network implementation involves four main steps. [9]

(1) Establishing the Network: The function newff is used to create the network, where the number of input and output layer neurons is determined by the sample data. Users then decide on the activation functions for the hidden and output layers, training algorithms, the number of hidden layers, and the number of neurons in each hidden layer based on experience and practical needs.

(2) Initialization: The function init is used for initialization. When creating a network object with

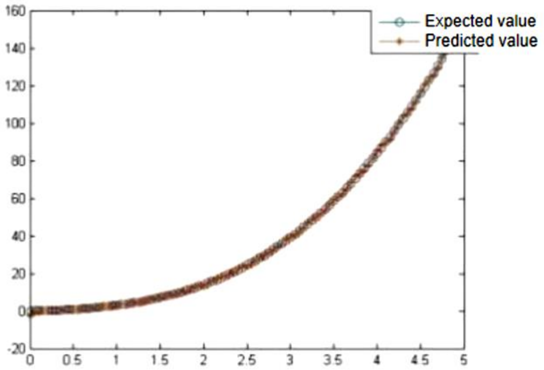
newff, the initialization function init is automatically called to initialize the connection weights and thresholds of the network based on default parameters.

(3) Network Training: The function train is used for training. The software trains the network based on the input vector P, target vector T, and predefined training function parameters.

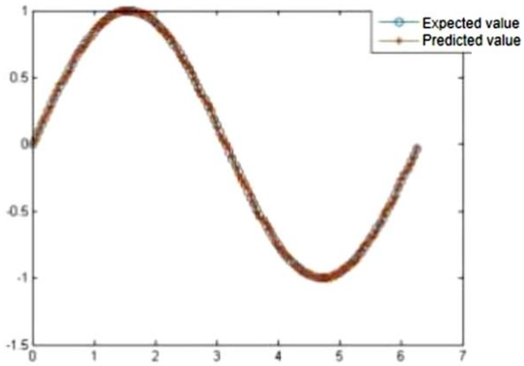
(4) Network Simulation: The function sim is used for simulation. The software performs simulation calculations on test data using the trained network.

Due to the ability of BP neural networks to approximate any function with arbitrary precision and simulate any nonlinear mapping in the real world, this section first uses the MATLAB Neural Network Toolbox to fit three simple functions (polynomial functions, sine functions, exponential functions) with BP neural networks to test the function approximation capability of BP neural networks.

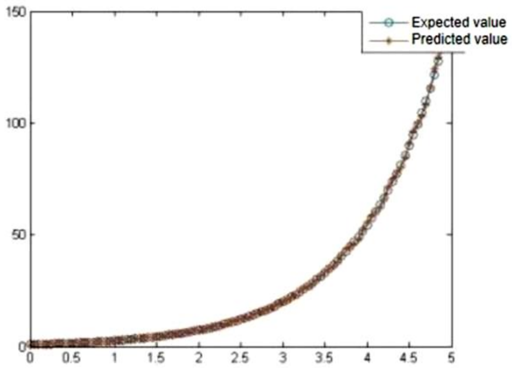
As show in Figure 2, it can be seen that the BP neural network is capable of rapidly and accurately fitting continuous functions. Due to this property of neural networks, this chapter will utilize the BP neural network to approximate the non-linear mapping of construction project costs in reality.



(a) Polynomial fitting



(b) Sine function fitting



(c) Exponential function fitting

Figure 2: BP neural network fitting function

3. Design of engineering evaluation model based on PSO optimization BP

3.1 Establishment of BP neural network prediction model

To establish an engineering cost estimation model using a BP neural network, the main tasks include:

(1) Data preprocessing

The input of neural network data is crucial as it directly impacts the accuracy of the model. After determining the input variables, data normalization is required using the standard deviation normalization method.

(2) Determining the network layers

The selection of hidden layers in the BP neural network directly affects the accuracy of the engineering cost estimation model. Based on previous experiences, a BP neural network with a single hidden layer can be used to fit most practical problems. Increasing the number of hidden layers may drastically increase the training time without benefiting the model training. Therefore, this article adopts a three-layer BP neural network with a single hidden layer.

(3) Selection of incentive function

When using a three-layer BP neural network, the choice of activation functions mainly focuses on the activation functions from the input layer to the hidden layer and from the hidden layer to the output layer. In MATLAB software, the default activation function for BP neural networks is the Log-Sigmoid function, which converts input values to numbers between [0,1], meeting the requirements of engineering valuation mapping. The output function uses a linear function, allowing output values to take on any value. Therefore, the activation functions set for the neural network model in this chapter are: Log-Sigmoid activation function from the input layer to the hidden layer; linear activation function from the hidden layer to the output layer.

(4) Selection of the number of nodes in the hidden layer

Determining the number of nodes in the hidden layer is a crucial step in defining the topology of a BP neural network model. It impacts the performance of the network model being established. Optimal selection of hidden layer nodes can effectively prevent the occurrence of "overfitting" during the training process, thereby enhancing the network's generalization ability. However, there is currently no scientific or universally accepted method for determining the number of hidden layer nodes. Most formulas proposed in existing literature for determining the number of hidden layer nodes are based on arbitrary numbers of training samples, often geared towards worst-case scenarios, making them challenging to apply in practical engineering. Given that Particle Swarm Optimization (PSO) possesses strong global search capabilities, this study utilizes PSO to optimize the number of hidden layer nodes in BP neural networks, identifying the appropriate number for estimating construction project samples.

(5) Choosing the error function

Once the network's topology and training data are determined, the total error function is determined by the activation function.

(6) Neural network weight initialization

A set of small numbers was randomly selected to initialize the network to ensure that the network will not enter the saturation state prematurely due to too much weight, resulting in training failure.

3.2 PSO optimizes BP neural network parameters

Although the back propagation (BP) neural network can approximate any function with arbitrary precision and achieve any complex nonlinear mapping from the input layer to the output layer, its algorithm is based on the gradient descent of the error function, which is essentially a single-point search method and lacks global search capability. Therefore, during the learning process, the BP neural network inevitably suffers from drawbacks such as slow convergence speed, susceptibility to local minima, poor robustness, and suboptimal network performance. On the other hand, the Particle Swarm Optimization (PSO) [10] algorithm has excellent global search capabilities by enabling collaboration and competition among individuals, reducing the risk of falling into local optimal solutions and exhibiting strong robustness. Hence, this chapter attempts to combine the strengths of both approaches by establishing a BP neural network estimation model based on particle swarm optimization.

The learning rate η also known as the step size in training a standard BP neural network, is typically kept constant. However, in practice, it is challenging to determine a fixed optimal learning rate. The choice of learning rate directly impacts the stability of the network. A high learning rate can reduce training time but may lead to oscillations, increasing the number of iterations. On the other hand, a low learning rate can prolong training time by increasing the number of iterations. Currently, the selection of the learning rate is mostly based on empirical methods, lacking a rational approach for selection and derivation. Additionally, the number of nodes in the hidden layer influences the performance of the network, as mentioned earlier in the text.

Particle swarm optimization algorithm can iteratively find the optimal solution. The basic idea of this section is to adjust the parameters of the particle swarm: N (number of particles), ω(inertia weight), M (evolutionary generation) to optimize the hid number (number of hidden layer nodes) and lr (learning rate) of the BP neural network. Through the PSO optimization algorithm, a parameter range with a higher probability of obtaining smaller errors is identified.

MATLAB software was used to import the engineering review case library, and 80% of the data in the case library is randomly selected as training samples, and the remaining 20% is used as test samples for network testing. The calling format is:

$$[xm, \breve{f}v] = PSO(\breve{@}fitness, N, c1, c2, \omega, M, D) \tag{7}$$

After running the file, the data in Table 1 shows that the "hidnumber" is between 21 and 38. The learning rate (lr) is generally below 0.08. Therefore, this article uses 35 as the number of hidden layer nodes and a learning rate of 0.05 as the optimized parameters for the engineering estimation model. With these parameters, there is a high probability that the training objective error will achieve a small value.

Table 1: BP neural network parameter optimization results

Optimization order	N	ω	M	hidnumber	lr
1	10	0.5	10	38	0.026
2	10	0.5	30	26	0.029
3	20	0.5	30	38	0.046
4	25	0.6	30	26	0.086
5	30	0.7	30	21	0.025

3.3 Training and simulation of evaluation model based on PSO-optimized BP

The number of training samples determines the input and output layer nodes of the BP neural network. The engineering valuation model constructed in this chapter uses the existing engineering case database for statistical analysis as the sample set. Therefore, the input nodes of the BP neural network engineering estimation model established in this chapter are 10, representing 9 features of the input samples: engineering purpose, engineering category, structure type, underground floors, aboveground floors, pile foundation, masonry, waterproofing and insulation, decoration. The number of hidden layer nodes is determined as 35 by optimizing the parameters of the PSO for the BP neural network. The output layer consists of 1 node, representing the construction cost per square meter of the project.

In summary, the engineering cost estimation model designed in this chapter utilizes a 3-layer BP neural network structure with 9-35-1 nodes in each layer. The learning rate is set at 0.05, determined by optimizing the BP neural network parameters using PSO. The activation function for the hidden layer is set to the default Log-Sigmoid function in MATLAB, while the output layer's activation function is a linear function. The MATLAB programming for the engineering cost estimation model based on PSO optimized BP neural network is shown in Figure 3.

```
clear,clc
data=xlsread(' data ');
Inputdata=data(:,1:end-1)';
Outputdata=data(:,end)'; long1=length(Outputdata); index1=randperm(long1);
numberTest=fix(0.2*long1);
                                   ⋮
%   Output plot
figure;
plot(T2,'-+');
holdon;
plot(testvalue,'r:*');
  legend('expected value ','predicted value');
  title('BP neural network model outputs prediction curves');
  xlabel(' input sample points');
  ylabel('cost per square meter');
```

Figure 3: Engineering valuation model MATLAB program

20% of 32 cases are randomly selected from the database as prediction samples, and the remaining samples are used as training samples to train and simulate the evaluation model of PSO-optimized BP neural network parameters. The comparison between the training simulation results and the actual values can be seen in Table 2 and Figure 4.

Table 2: Comparison between test sample predictive value and actual value

sample number	1	2	3	4	5	6
predictive value/yuan/m2	1588	1345	1837	1084	1426	1155
actual value/yuan/m2	1709	1253	2003	1096	1502	1224
relative error/%	-7.1	7.3	-8.2	-1.1	-5.1	-5.6

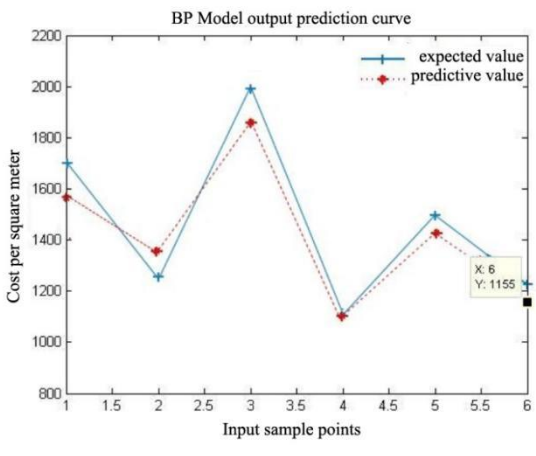


Figure 4: Comparison between predicted value and actual value

From Table 2 and Figure 4, the results of the engineering cost estimation model based on PSO-optimized BP neural network data show a maximum error of -8.2% and a minimum error of -1.1%. This meets the requirement of investment estimation accuracy not exceeding 20% during the feasibility study phase of engineering construction. Therefore, this network estimation model can assist in estimating project costs. Some individual cases have relatively large errors in estimation prediction, indicating that the generalization ability of this network model is still imperfect. This issue can be addressed by selecting more representative engineering features and increasing training samples.

4. Conclusion

This article provides a brief introduction to artificial neural networks, focusing on the basic principles of BP neural networks, verifying their data fitting capabilities. The chapter tests the network's fitting ability using polynomial functions, exponential functions, and sine functions. The results show that BP neural networks can fit various functions well, with fast and effective fitting, making them suitable for establishing neural network engineering valuation models. To address the shortcomings of BP neural networks such as slow convergence speed, susceptibility to local minima, poor robustness, and network performance, the article optimizes the network's parameters using a particle swarm optimization algorithm known for its excellent global search capability. The main parameters optimized are the number of nodes in the hidden layer and the learning rate of the network. By establishing an engineering valuation model based on PSO-optimized neural network parameters and training and testing the data using the MATLAB Neural Network Toolbox, the model demonstrates low error rates and the ability to quickly predict construction costs per square meter, providing significant reference value for practical engineering applications.

References

[1] Peng Guangjin. Research on intelligent Learning Method of Small sample engineering cost Data and its Application in power transmission and transformation engineering [D]. Chongqing: Department of Electrical Engineering, Chongqing University, 2010:89-93.

[2] Chen Junjia. Discussion on the control of engineering cost in the process of high stage design [J]. Metal Mine, 2002, 34(3): 50-51.

[3] Ma Guofeng, Chen Zhiqin. Research on project Evaluation of rail transit station based on BP Neural Network [J]. Project Management Technology, 2011, 9 (7): 55-60.

[4] Linli Jiang, Jiansheng Wu. Hybird Evolutionary Algorithms for Artificial Neural Network Training in Rainfall Forecasting [M]. New York: Lecture Notes in Computer Science, 2013:78-82.

[5] Calbort Master. Praetieal Neural Network [J]. IEEE Transaetions on Neural Network, 1995, 27(5): 67-73.

[6] J.Park, W.Sandberg. Universal Approximation Using Radial Basis Funetion Networks [J]. Neural, 1991, 35(12): 83-95.

[7] Speeht D F. General Regression Neural Networks [J]. IEEE Trans on Neural Networks, 1991, 23(12): 35-42.

[8] F C Chen.Baek-propagation Networks for Nonlinear Self-tuning AdaPtive Control [J]. IEEE IEEE Control Systems Magazine, 1990, 10(3): 44-48.

[9] Angeline P J. Using selection to improve particle swarm optimization system [J]. Intelligent Engineering Systems through Artificial Neural Networks, 1998, 18(4): 253-258.

[10] Kannan S, Mary Raja Slochanal S. Application of particle swarm optimization technique and its variants to generation expansion planning problem [J]. Electric Power Systems Research, 2004, 12(70): 203-210.