2nd International Conference on Science, Technology and Management
University of Delhi (DU), Conference Centre, New Delhi (India)
27 September 2015, www.conferenceworld.in, www.icstmdu.com
978-81-931039-6-8

ASSESSMENT OF APARTMENT PRICES CALCULATED BY DIFFERENT NEURAL NETWORK ALGORITHMS

Kosa Golic

Associate Professor, Faculty of Construction Management, University Union Nikola Tesla, (R of Serbia)

ABSTRACT

This paper examines applications of different types of Artificial Neural Network (ANN) training algorithms for the assessment of the apartments prices in the city Budva, R of Montenegro. For each algorithm an optimized ANN model is developed comprising the combination of various model parameters: the value of momentum coefficient and learning rate, different transfer functions, the number of hidden layers and the number of neurons in hidden layers as well. The results are compared and discussed. The benefits of the ANN algorithms are pointed out and the guidelines for the further research are given.

Keywords: Apartment Price Assessment, Neural Network Model

I. INTRODUCTION

Evaluation of the apartment prices is a complex problem because it depends on a large number of mutually dependent, subjective and dynamic factors as well. Numerous methods can be found in the literature for solving this type of problem, but it has been proved that currently used statistical or traditional methods have definite drawbacks, because they can not appropriately deal with highly correlated and/or incomplete data. Fuzzy methods, owing to the fuzzy set theory, have made possible to overcome many drawbacks of traditional methods but still the problems with a high level complexity and uncertainty neither can it be appropriately solved by them. On the other hand, it has been proved that any arbitrarily irregular pattern can be mapped by ANNs provided that the set of input data is large enough [1]. Also, it has been proved that problems with multi-attributes can be solved better by ANNs then by traditional methods [2]. Furthermore, ANNs can learn by themselves and generalize solutions as well. There are many publications describing the theory of ANNs from the introductory to more advanced levels. A detailed description of various types of ANNs is presented in [3], [4], [5], [6], [7], [8], [9], [10]. An exhaustive explanation of the feedforward networks and the back-propagation method for network training is presented in [11], [12], [13]. A statistical interpretation of the methods used to train ANN is given in [14].

Evaluation of the apartment prices on the basis of their characteristics is a typical problem of finding nonlinear multivariate function between two sets. For the problem in question, the following characteristics of the apartments are considered: area of the apartment, city zone, floor on which apartment is placed, number of rooms and apartment quality. A set of data, in total 237, for the network training was collected from the Real

978-81-931039-6-8

State Agencies in the city Budva, R Monte Negro. A part of them is shown in Table 1. For the assessment of the apartment quality, the qualitative values assigned by the agencies: “excellent”, ”very good”, “good” and “poor” were weighted as follows: 6 for “excellent”, 4 for ”very good”, 3 for “good” and 1 for “poor”. Quantification of the city zones was given in analogous way: 1 for downtown, 2 and 3 for the zones less attractive, while for “building age” as follows: 1 for the buildings less than 10 years old, 2 for less than 20 years old, and so on up to 7 meaning more than 60 years old building. The values of the apartment areas and the number of bedrooms were taken in their original state, since they were already quantified. All these values were reduced to the interval [-1,1] in order to make training process more efficient.

For the problem in question, first of all a network architecture was determined and then for each applied training algorithm the optimization of the model parameters was executed. The accuracy of the proposed prediction models was tested trough the comparison of the calculated and gathered data by the application of linear regression function.

Table 1 One Partition of the Collected Data Set for Network Training

No	Flore	Area	No of Rooms	Zone	Quality	Price
1	0	31	1	1	1	25000
2	0	39	1	1	3	31000
3	3	36	1	2	3	28000
4	1	49	1	2	4	42000
5	3	43	1	2	6	43000
6	1	41	1	2	3	31000
7	5	39	1	2	6	39000
8	1	35	1	2	4	31000
9	3	48	1	2	4	40000
10	1	31	1	2	3	25000
11	2	47	1	2	6	52000
12	0	29	1	2	1	23000
13	5	35	1	2	4	33000
14	0	38	1	2	4	35000
15	1	38	1	2	3	27000
16	0	37	1	2	3	28000
17	5	42.06	1	2	1	27000
18	3	37	1	2	3	29000
19	2	52	1	2	4	44200
20	4	52	1	2	3	37000
21	1	42.46	1	2	4	36000
22	4	40	1	2	6	40000
23	0	59.5	1	3	3	46000
24	2	44	1	3	6	39600
25	2	40	1	3	1	27000

26	2	50	1	3	3	38000
27	4	38	1	3	6	35000
28	1	55	1	3	3	43000
29	4	40	1	3	3	29000
30	0	40	1	3	1	25300
31	4	80.5	2	1	3	75000
32	2	86	2	1	3	92000

II. DESCRIPTION OF THE ANN MODEL

ANN offers a paradigm to computation that is different from conventional analytical methods or statistical methods. There is diverse types of ANNs in the terms of topology and mode of operation [6]. Fig. 1 illustrates a sample three layer neural network used in this model. The problem of learning in neural network is simply the problem of finding a set of connection strengths w_{ij}, v_{jk}, Fig. 1, which allows the network to perform requested computation [5], [6]. The learning method used in the model presented in this paper is Back propagation (BP) method [11]. It is the most widely used algorithm in neural networks. The main idea applied in the networks with BP algorithm is to propagate the input forward through the layers up to the output layer and then propagate the errors back through the network from the output layer to input layer. Input data are presented to the network for each sample with their forecasting targets. The model is required to learn the underlying patterns between these sets of input and output. Each input node of input layer receives input signal *Xi* and transmits this signal to all nodes in the hidden layer. Then, each hidden unit Y_j sums its weighted input signal (1) and applies assigned activation function to compute output signal. Each output unit O_k sums its weighted input signal and applies specified activation function to compute its output signal (2).

$$Y_j = f_a\left(\sum_{i=1}^{m} w_{ij}\, x_i + b_j\right) \quad (1)$$

$$O_k = f_o\left(\sum_{i=1}^{n} v_{jk}\, y_j + b_k\right) \quad (2)$$

Where v_{jk} is the weight of the connection from the *j*th hidden unit to the *k*th output unit. The parameter of bias *b* in (1) and (2), also called the threshold value is permanently set to 1 in hidden layer as well as output layer so that corresponding weight shifts the activation function along the *x* axis. The activation functions used in this study are sigmoid and tangent sigmoid in the hidden layer and the linear activation function is used in output layer.

The BP training algorithm is an iterative gradient descent algorithm, designed to minimize the mean of square error (MSE) and mean absolute error which are averaged over all patterns [11], [12], [13]. Back/propagation algorithm represents one of the most important discoveries in the neural network development It was independently developed developed by two tims: one from France (Fogelman-Soulie, Gallinari and Le Cun) and the other from USD (Rumelhart, Hinton and Williams). Training is the process of continuously adjusting the connection weights until they reach unique values that allow the network to produce outputs that are close enough to actual desired outputs. The accuracy of the developed model, therefore, depends on these weights. Once optimum weights are attained, the weights and biased values decipher the network's state of knowledge. In order to determine them, it is recommended to divide collected data set into three partitions: training, validation and testing set. For the problem presented, the training set contained 70% of the collected data, while

978-81-931039-6-8

the validation and testing set contained 20% and 10%, respectively [15], [16]. The optimal model is considered the one which gives the best results both for validation and testing set.

III. PROBLEM MODELLING

One of the most important tasks in ANN model development is to find the optimal network architecture. The network architecture is to be selected out of several network configurations comprising the combination of various model parameters: the number of hidden layers, the number of neurons in hidden layers, different activation functions, the value of momentum coefficient and learning rate as well, [9], [10].

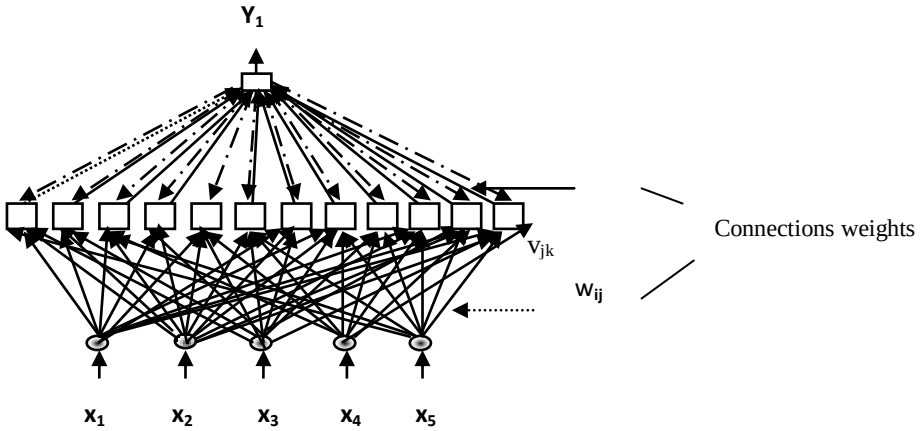

Figure 1. Graphical Representation of ANN Model

For the problem in question, a three-layer feedforward network with five nodes in the input layer and one in the output layer was designed as the initial model, Fig. 1. Each node in the input layer corresponds to one of the specified apartment characteristics X: area of the apartment, city zone, floor on which apartment is placed, number of rooms and apartment quality, while the output node corresponds to the estimated prices Y. Twelve nodes in the hidden layer were adopted for the initial model and during the training process this number was adjusted for each applied training algorithm.

The sigmoid and tansig transfer function were used in the hidden nodes and the linear for the output. The list of back-propagation algorithms applied for network training is given in Table 2.

Table 2. List of Back Propagation Training Algorithms

Acronim	Algorithm
GD	Gradient Descent
GDM	Gradient Discent Momentum
GDX	Variable Learning Rate Backpropagation with momentum
GDA	Variable Learning Rate Backpropagation
RP	Resilient Backpropagation.
LM	Levenberg- Marquardt
LMREG	Levenberg-Marquardt with Bayesian regularization

Training algorithm has enormous influences on the model accuracy because it calculates the weights W and

978-81-931039-6-8

biases b of the network. In order to obtain optimal model, each of the model parameters was varied while keeping the others constant to examine its influence on the accuracy of the model. This is consistently done for each training algorithm. All calculations are done by Matlab R2009a software.

The learning rate and momentum coefficient values affect the ANN performance in a great deal as well [17]. The greater the learning rate, the greater the modification of the weight values, but too low learning rate leads the network to learn slowly. Contrary, too high learning rate makes the weights and objective function diverge. Therefore, a compromise solution should be found. In general, a small learning rate and large momentum are desirable. Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Commonly, the optimal values of these parameters are reached by a trial and error.

Optimal number of neurons in the hidden layers of ANN model was obtained by trial and error as well. Generally, using too few neurons in the hidden layers will result to under-fitting. Under-fitting happens when there are too few neurons in the hidden layers to adequately reveal the signals in a complex data set. Contrary, too many neurons in the hidden layers may result in over-fitting. It appears when the neural network has so much information processing capacity that the amount of information contained in the training set is not big enough to train all of the neurons in the hidden layers. When the network begins to over-fit the data, then the error in validation set will usually start to increase [10]. At first, the ANN model with one hidden layer made of 12 neurons with sigmoid activation function was used. In the present case, it resulted that for GD and GDM algorithm the optimum number of hidden nodes was 12, for GDA and GDX was 10 while for LM, LMREG was 7 and 9 for RP. Application of two hidden layers have not caused more accurate approximation of the problem as well the implementation of tansig transfer function instead of sigmoid function did in the hidden nodes.

IV. DATA ANALYSIS

The training process revealed that the best results are obtained by the LMREG algorithm, Table 3. This is followed by Rp-algorithm, then by GDX and GDA. The standard GD and GDM could not give satisfactory results even if the number of layers and number of nodes were increased. It follows that a better approximation could be reached by increasing the training data set if it is possible or by improving the network generalizations [18]. This was just done by using Bayesian regularization method in combination with Levenberg-Marquardt algorithm-LMREG [19]. In this case, the learning rate of 0.4 led to the best convergence and the lowest model error.

Table 3 Deviation of the Results Calculated by Different Training Algoritms from the Target Values in %

Area (m^2)	TRAINING ALGORITHMS						
	LM (%)	LMREG (%)	RP (%)	GDX (%)	GDA (%)	GDM (%)	GD (%)
31	-0.01	4.80	11.53	6.67	10.28	-0.68	-3.02
39	0.00	-10.51	-1.00	-0.75	2.34	-12.30	28.35
36	-1.89	0.83	-3.48	-2.55	-7.99	-78.86	39.77
49	-0.04	-2.52	2.40	1.61	5.34	16.84	1.85
43	0.14	-2.10	-6.33	-2.78	-2.41	16.04	-25.10
41	0.32	-2.88	-4.65	-2.80	-5.37	-6.44	-11.96
39	0.06	-1.98	-2.32	0.57	-3.76	10.07	-7.89
35	-0.10	-2.72	-9.06	1.35	-0.72	10.86	-17.21
48	0.01	-3.59	12.11	0.17	4.14	-6.10	12.64
31	0.34	0.21	-19.36	-5.33	-12.92	-1.19	-26.17
47	-0.03	7.89	1.40	4.23	8.31	34.63	-21.07
29	-0.05	3.60	-13.47	0.44	-2.71	54.79	-54.96
35	0.01	5.68	-2.62	0.09	8.87	-31.89	8.69
38	-0.09	1.95	-1.42	11.70	8.07	61.81	4.85
38	-0.79	-10.09	-16.75	-11.06	-16.16	-14.11	-25.63
37	0.10	-4.87	-12.91	1.98	-6.56	54.88	-16.24
42.06	0.01	8.81	-25.10	3.50	-7.32	-140.30	9.34
37	1.76	1.84	-0.26	-0.48	-5.87	-72.58	39.09
52	-0.04	-2.60	9.11	-1.42	3.67	6.95	2.44
52	-0.01	-7.59	11.54	-0.04	-0.02	-32.76	16.72
42.46	0.24	-4.32	-2.57	0.18	0.92	11.92	-8.38
40	-0.06	-2.61	-6.77	-0.02	-5.90	9.46	-17.95
59.5	-0.01	6.00	4.43	-1.64	12.87	8.11	12.77
44	0.00	-2.16	-8.66	-0.83	-5.73	32.64	-8.87
40	0.02	4.08	-6.98	10.04	-17.88	-11.86	-15.03
50	-0.07	3.80	11.24	-1.35	3.57	36.08	18.73
38	0.00	-1.63	0.22	0.75	-8.39	1.17	-1.30
55	0.04	6.79	11.87	5.19	5.19	21.46	20.49
40	-0.07	-2.43	3.29	1.11	4.39	28.14	-6.14

40	0.06	-8.09	-7.43	-1.45	-4.94	27.76	-47.11
80.5	0.00	-3.28	-1.27	0.20	-0.83	21.72	-2.19
86	0.00	9.82	10.08	0.78	11.31	30.41	4.64

The main reason for which the standard methods GD and GDX were not able to give satisfactory results was because the field of input values was not entirely covered by the training data set. It happened due to the complexity of the input variable ”apartment quality” incorporating many apartment attributes such as: interior quality (quality of the floors, carpentry, interior design of kitchen and bathrooms, etc.), furthermore quality of the building (construction year of the building, general maintenance of the building, roof and façade maintenance, type of construction system, number of flats, etc.) as well as apartment view and orientation, local landscape, street noise, etc.

Therefore, compressing all these specified characteristics into one obfuscates insight into the relationship with the apartment prices and it usually happens that the field of input data is not entirely covered. Generally, the more complex is the input variable the greater is the training data set required. The situation becomes even more complicated when the variables are exposed to subjective evaluations which was also the case in the present problem.

V. CONCLUSION

Application of the ANN model is successful in calculating apartment prices. Also, by virtue of Levenberg-Marquardt method with Bayesian regularization, this problem can be solved even if the input variables are highly complex and mutually dependent as well as in case of incomplete set of training data. But, in general, the more complex the input variables the greater the training data set is required. The LMREG algorithm, apart from being the most successful in approximation of apartment prices, it was the fastest algorithm as well. It needed a small fraction of time for network training taken by the other algorithms.

Concerning this problem, further research should be concentrated on the apartment price changes over time as a consequence of economical and political changes as well as a result of globalization processes, technological innovations and climate changes also..

REFERENCES

[1] P.R. Hush, B.G. Horne, Progress in supervised neural networks, IEEE Signal Processing Journal, 1, 1993, 8-39.

[2] T. Masters, Practical neural networks recipes in C^{++} (London, Academic Press London, 1993).

[3] T. Kohonen, An introduction to neural computing, Neural Networks, Vol. 1(1), 1998, 3-16.

[4] Haykin, S., Neural Networks A Comprehensive Foundation, (New York, MacMillan, 1994).

[5] I. Aleksander, H. Morton, An Introduction to Neural Computing (London, International Thomson Computer Press, 1995).

[6] Fauset, L., Fundamentals of Neural Networks: Architectures, Algorithms and Applications (Englewood Cliffs, NJ: Prentice Hall, 1994).

[7] A. K. Palit and D. Popovic, Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications (London, Springer, 2005).

[8] S. Karnarthi, V. Sanvido, and R. Kumara, Neuroform – neural network system for vertical formwork selection, ASCE Journal of Computing in Civil Engineering, 6(2), 1992, 178-99.

[9] P. S. Hush, and B.G. Horne, Progress in supervised neural networks, IEEE Signal Processing Journal, 1, 1993, 8-39.

[10] J.A. Herz, R.G. Krogh, Introduction to the Theory of Neural Computation (New York: Addison-Wesley, 1991).

[11] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Learning internal representations by error propagation, Parallel Distributed Processing, 1(8) , 1986, Cambridge, MA: MIT Press.

[12] P.J. Werbos, Building and understanding adaptive systems: A statistical/numerical approach to factory automatization and brain research, IEEE Trans. System, Man, Cybern. 23, 1974, 131-147.

[13] C. Peterson, J.R., Anderson, A mean field theory learning algorithm for neural networks, Complex Systems 1, 1987, 995-1019.

[14] E. Levin, N. Tishby, and S.A. Solla, A statistical approach to learning and generalization in layered neural networks, IEEE Proceedings, 78(10), 1990, 1568-74.

[15] B. Widrow, D. Rumelhart, and A. Lehr, Neural networks: application in industry, business and science, Communications of the ACM, 37(3), 1994, 93-105.

[16] A. H. Boussabaine, The use of artificial neural networks in construction management: a review, Construction Management and Economics, 14(5), 1996, 427-36.

[17] R. A. Jacobs, Increased rates of convergence through learning rate adaption, Neural Networks 1, 1988, 295-307.

[18] H. Drucker, Y. LeCun, Improving generalization performance using double backpropagation, IEEE Trans. Neural Networks 3(6), 1992, 991-997.

[19] M. T. Hagan, M. Menhaj, Training feed forward networks with Marquardt algorithm, IEEE Transaction on Neural Networks, 5(6), 1994, 989-993.