Indian Journal of Science and Technology, *Vol 7(6), 838–843, June 2014*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

A New Approach to Software Project Cost Estimation using a Hybrid Model of Radial Basis Function Neural Network and Genetic Algorithm

Maryam Molani[1*], Ali Ghaffari[2] and Ahmad Jafarian[3]

[1]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, West Azerbaijan, Iran; molanymaryam@gmail.com
[2]Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran; a.ghaffari@iaut.ac.ir
[3]Department of Mathematics, Urmia Branch, Islamic Azad University, Urmia, Iran; Jafarian5594@yahoo.com

Abstract

The use of Artificial Neural Networks (ANNs) is growing day by day in various areas of software engineering and the relevant studies are significantly focused on the software project cost estimation. The problem of high accuracy cost estimation is one of the most important issues, which is still a challenge for researchers in the area of software. In this paper, a new method is presented based on Radial Basis Function (RBF) neural network and Genetic Algorithm (GA) to estimate software project cost. So, to provide optimal models, we tried to achieve good results by selecting proper parameter and network architecture. To prove the optimality of our model, the output results of the proposed model were compared with those of the estimation via the COCOMO81 algorithmic model. Test results show that the proposed adaptive models provide greater accuracy than algorithmic models, without the need to involve estimator in the details and complex relationships of the attributes and drivers of software project cost.

Keywords: Artificial Neural Network Model, Genetic Algorithm, Radial Basis Function Neural Network, Selecting the Appropriate Attributes, Software Project Cost Estimation

1. Introduction

Today, with the advancement of technology, the complexity of software has been growing day by day so that algorithmic models no longer have effectiveness and stability in their results, and also need the inputs that are very difficult to obtain during the early stages of software development projects[1]. In addition, they face difficulties in modeling complex relationships between the effective factors and reasoning abilities[2]. The increased accuracy of cost estimation, budget management and scheduling can facilitate software projects during the development process[3]. Effort estimation models for software help project managers to estimate project delivery time, cost and manpower required to develop software[4]. In recent years, to achieve high accuracy estimation, learning-based non-algorithmic methods were created as a supplement and alternative to algorithmic techniques[5] and are based on techniques such as fuzzy logic, ANNs, decision trees, evolutionary computation[6] so that, in recent decades, they are proven to have better performance than algorithmic models[7]. Due to the ability to learn and model the

*Author for correspondence

complex nonlinear relationships, ANNs have been able to achieve great success in the cost estimation[3,8].

In this paper, we intend to develop software project cost estimation models consistent with the COCOMO81 model using the technique of RBF neural network. In addition, to present a reduced model, GAs will help us to select a set of the attributes related to the total attributes of software projects. The test results show that if some parameters are ignored in the software projects, estimation based on a number of them not only do not harm the accuracy of the estimation, but also increase its accuracy. As an optimal algorithm, GA ensures the selection of the most relevant attributes so that, on the whole, the network training will be improved. Finally, to prove the optimality of the proposed models, we compare them with the output of the COCOMO81 algorithmic model.

Furthermore, this paper is organized as follows: Section 2 examines studies related to the software project cost estimation problem. In Section 3, we will compare the estimation based on the COCOMO81 algorithmic model and two selected models RBF neural network model and the reduced model of RBF network, using GAs, and then will evaluate the tests and compare the simulation results of the selected models in Section 4. Finally, Section 5 will be done conclusions.

2. Related Works

RBF neural network is an alternative to multilayer perceptron neural network, which has been able to perform better in some cases, so that it was used to estimate software projects[9]. Here, two tests were used to compare the results of the performance of the ANNs with those of regression analysis model: the first involves the third generation languages, and the second is a combination of the third and fourth generations. Output results based on the MAPE evaluation criterion for the model of this network are equal to 47.6% for the function point index and 31.96% for the index of lines of code.

Another study based on RBF model was performed[10], which compares the two models of RBF network and support vector regression simultaneously. This simulation is based on dataset that is taken from NASA software projects. The sequential optimization algorithm is also used to optimize the parameters of support vector regression model. The best results of the MMRE and PRED criteria (25%) for RBF network are equal to 0.1870 and 72.22%, respectively. The value of the second criterion for support vector regression indicates that 15 of 18 projects are within 25% of the properly estimated effort, while it includes only 13 projects for radial basis function network.

Much research has considered the use of GAs as a powerful tool to optimize model parameters based on machine learning, among which[11] combines the GA with the neural network model of Support Vector Machine (SVM), to predict software reliability. In this paper, which examines the major impact of key parameters of the SVM on the performance, GA is used to optimize the parameters of the SVM model simultaneously. So, if a certain range of parameters is considered as the search space, coding to convert it to decimal form of chromosomes was conducted on GA, and the fitness evaluation is made according to the MSE criterion. According to the results obtained in this study, the proposed model has been proven to perform better than other models that use only SVM without optimizing the parameters, and MSE value of the proposed method reached 3.55.

Furthermore, model[11] which was inspired by Wang and Hung model, another GA-based method is used to select the optimal parameters and attributes for machine learning regression, in order to estimate software effort. This paper evaluates and compares three techniques of Support Vector Regression (SVR), multilayer perceptron neural network (MLP) and GA-based tree model, which its simulation results show an improvement of this method, compared to the recently studied methods for estimating software effort.

In addition, Zeng[12] improved the software effort estimation using the neural function approximation. It only used the appropriate attributes from those considered in software projects, in order to improve the estimation performance. To build the model, 16 independent features of the original dataset (which include COCOMO 81, NASA 60, NASA 93, Albrecht CF and Desharnais) were used to adjust the COCOMO81 equation; and the feed forward architecture was selected for the ANN. The simulation results show that the proposed model performs better than the single ANN model so that the MMRE and PRED evaluation criteria (25%) are equal to 1.0 and 0.56 for the COCOMO81 dataset and 0.18 and 0.81 for the NASA 60 dataset, respectively.

3. The Proposed Model

The main problem for creating an optimal model is to make a more accurate estimation of the amount of software

project cost, as compared with the proposed algorithmic models. ANNs, which are a subset of the machine learning-based models, are a powerful tool for the problem of high precision estimation. In this paper, the RBF model is trained on the basis of the COCOMO81 dataset, and a reduced model is proposed based on GAs. First, we do estimation based on the intermediate COCOMO model, in order to compare the estimation results of the three models and to show the improved accuracy of the ANN models as compared to algorithmic models.

For an overview, the work process in the estimation using the models based on ANN techniques is given as a flowchart in Figure 1.

3.1 RBF Networks

RBF network is a popular alternative to the multilayer perceptron network, and has comparatively better performance when there is a large number of training samples[5]. Training RBF network is easier than the perceptron network training[13].Although it is similar to the multilayer perceptron network in terms of the number of layers and the architecture[8]. Its hidden units have a structure different from the units of the input/output layer[14]. The transfer function in the neurons of the hidden layer is based on a Gaussian distribution[9], which is calculated according to Equation (1)[10].

$$\varphi_j\left(\left\|x-c_j\right\|\right)=e^{\left(1/2\sigma^2\left\|x-c_j\right\|^2\right)} \quad (1)$$

where, c_j is the center of RBF for the j^{th} neuron, σ is the Gaussian width, and x is the input vector[13]. The output of RBF network is calculated according to Equation 2[13].

$$y_k(x)=\sum_{j=1}^{n} w_{kj}\varphi_1(x) \quad (2)$$

3.1.1 RBF Network Architecture

The model proposed in this paper on the RBF network architecture consists of two layers[15]. The neurons in the first layer calculate the distance between the centers and the input[16]. That we were considered equal to 16 neurons (number of attributes of the intermediate COCOMO dataset) and the second layer calculates the weighted sum of the outputs of the first layer and passes it through the linear activation function[17]. Figure 2 shows the RBF network architecture.

$$y=\text{pureline}\left(\sum_{j=1}^{n}\left(w_j\,\text{radbas}\left(\left\|w-p\right\|\right)*b_1\right)+b_2\right) \quad (3)$$

where, i and j are the number of neurons in the first layer and the hidden layer, respectively; and R represents the total number of elements in the input vector[14].

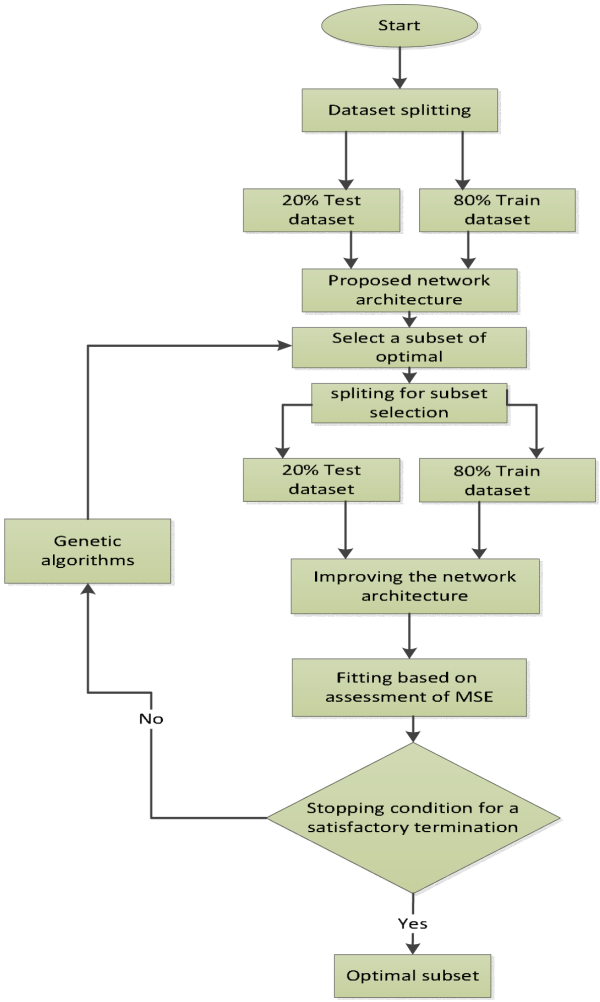

Figure 1. A schematic of the proposed model.

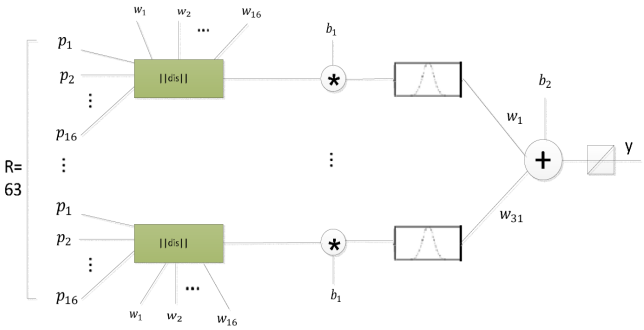

Figure 2. RBF network architecture.

3.1.2 *Training RBF Networks*

Training of a RBF network involves a two-step learning process[17]. As a first step, the input dataset is used to determine the parameters of basis function φ_j (for example, σ and c_j), and the weights w_{kj} are obtained in the second step[18].

Here, the newrb function is used to generate RBF network and requires two parameters of objective (based on MSE) and spread[19]. To achieve the optimal value of spread, different values of the parameter are given to the function, and then, the network performance is compared[13]. According to Table 1 and Figure 3, the RBF network model with a spread of 0.05, 31 hidden layer neurons, is best performance.

Figure 3 shows the learning curve for the best RBF network model, wherein the trained network has reached its best performance in the step 5.

Table 1. Results of training RBF networks

MSE(test)	Epochs	Neurons	Spread	Goal
0.03482	25	25	2	0.00001
0.00767	18	18	1	0.00001
0.00179	16	16	0.5	0.00001
0.00112	11	11	0.1	0.00001
0.00033	31	31	0.05	0.00001
0.00063	34	34	0.01	0.00001
0.00090	31	31	0.005	0.00001
0.00093	22	22	0.001	0.00001

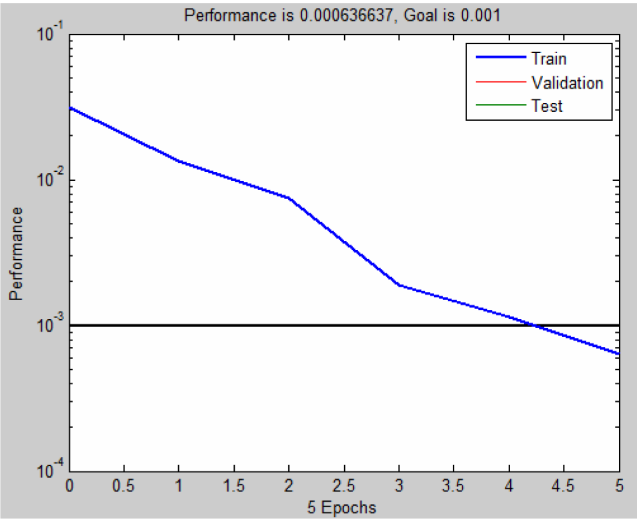

Figure 3. Performance of the improved RBF network during training.

3.2 Creating a Reduced Model using GA

Selecting a subset of relevant attributes from a dataset plays an important role in increasing the performance of estimation problem, or the very machine learning[11]. Among the 16 attributes that have been intended for the intermediate COCOMO model to estimate the amount of software project effort, we select 10 of the most relevant and the most appropriate attributes by using GAs, so that it not only reduces the complexity of the model but also raises the network estimation performance[4].

Genetic algorithm is an efficient, parallel and global optimization solution based on the principle of survival of the fittest[20]. Genetic algorithm is based on the chromosome that offer a candidate solution for a given problem, and the evaluation of each candidate solution or chromosome is made on the basis of their fitness value, which is determined by the objective function, and has the ability to survive chromosomes in the current population[3]. After the chromosomes with higher fitness values were selected for the new generation, a series of operations on the chromosomes lead to promote optimal solutions and overcome the disadvantage of local minimization in the presented problem, and a new population is generated. The most popular of these operations include selection, crossover and mutation[21].

The results of implementing the reduced model RBF network on the COCOMO81 dataset based on the MSE are shown in Figure 4 and Table 2, respectively.

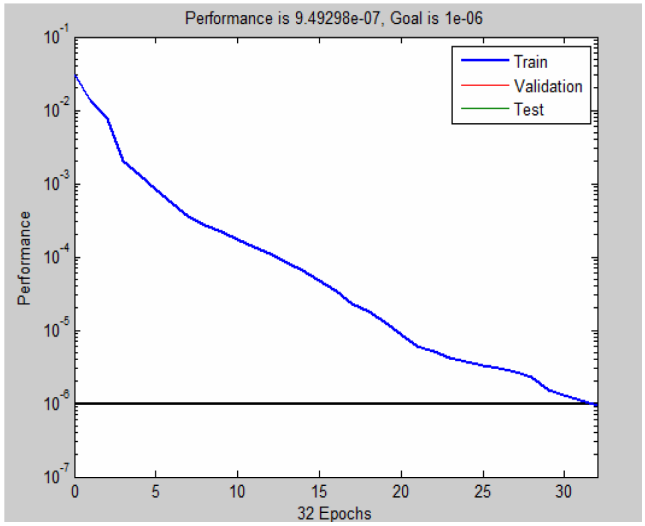

Figure 4. Performance of the improved RBF network using GA during the training.

4. Evaluation of Results

This research was carried out using the MATLAB programming language. At first, the value of effort estimation error for a hybrid model of RBF neural network using a GA with the amount of effort of COCOMO algorithmic model on 63 COCOMO81 dataset projects is depicted in Figure 5, to demonstrate optimal performance of ANN models. It can be seen that in most cases, the error of the hybrid model of ANN with GA has a value much less than an algorithmic model.

The results of the intermediate COCOMO algorithmic model are based on three evaluation criteria of the Mean Square Error (MSE), the Mean Magnitude of Relative Error (MMRE), and the Median Magnitude of Relative Error (hereinafter briefly called MdMRE), which are listed in Table 3.

The results of training the best RBF network architecture and the reduced model of RBF network, along with some of the most relevant attributes, are presented in Tables 4 and 5, respectively. The mean square error of the test is equal to 0.0033 for the reduced model of RBF network and 6.7922e-06 for the reduced model of RBF network. A comparison of the results of the two network models shows that RBF models have much higher accuracy than algorithmic models, and that the reduced model of RBF neural network has the error value less than a simple model of RBF network.

Table 2. Results of the Combined model of a GA with RBF network

MSE(train)	MSE(test)	Neurons	Spread
0.0014	0.0269	25	2
2.9054e-04	0.0121	25	1
1.0006	0.0222	25	0.5
1.2440e-05	0.0192	25	0.1
9.0092e-07	0.0024	34	0.05
9.9614e-07	6.7922e-04	33	0.01
8.7999e-06	5.58764e-02	36	0.005
2.8941e-06	0.0013	32	0.001
7.7561e-06	0.0284	37	0.0001

5. Conclusions

The software project cost estimation using ANNs is one of the most powerful techniques to estimate software projects, so that high accuracy estimation eliminates the need of software companies for experts in this field and does not involve them in details and knowledge of the attributes of software projects, to estimate the amount of effort. Since we are increasingly faced with the advances in software technology, more complex algorithmic equations will be naturally required, which may not be amenable to do high accuracy estimation and may need more efficient professionals to offer. Due to the strong capability of learning

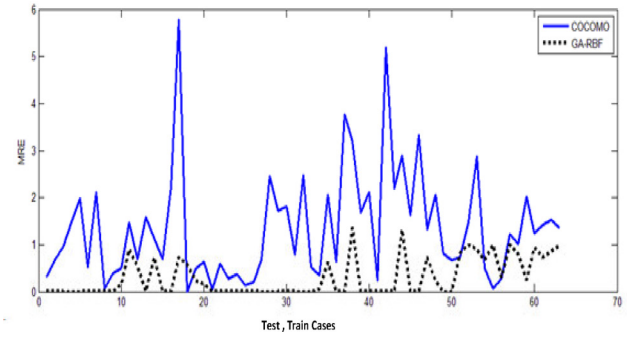

Figure 5. Comparison of the value of effort estimation error of COCOMO algorithm model and reduced rbf-GA model.

Table 3. Results of the COCOMO dataset using an intermediate COCOMO algorithmic model

	MdMRE	MMRE	MSE
COCOMO81	113.2	2.92	0.0287

Table 4. Results of the COCOMO dataset using the proposed rbf model

	MdMRE(train)	MdMRE(test)	MMRE(train)	MMRE(test)	MSE(train)	MSE(test)
RBF	0.0077	0.0137	0.4220	0.9665	3.5476e-06	0.0033

Table 5. Results of the evaluation criteria for the best architecture of reduced rbf-GA model

	MdMRE(train)	MdMRE(test)	MMRE(train)	MMRE(test)	MSE(train)	MSE(test)
RBF-GA	0.0125	0.0044	0.1636	0.5485	9.9614e-07	6.7922e-06

and modeling complex non-linear relationships, the neural network eliminates the need of specialists to use complicated algorithmic models.

6. References

1. Kaushik A, Soni AK, Soni R. An adaptive learning approach to software cost estimation. National Conference on Computing and Communication Systems (NCCCS); 2012. p. 1–6.
2. Park H, Baek S. An empirical validation of a neural network model for software effort estimation. Expert Systems with Applications. 2008; 35(3):929–37.
3. Li Z. Intelligently predict project effort by reduced models. 2010 International Conference on E-Business and E-Government (ICEE); 2010; Guangzhou. p. 1536–42.
4. Oliveira ALI, Braga PL, Lima RMF, Cornélio ML. GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. Inform Software Tech. 2010; 52(11):1155–66.
5. Dejaeger K, Verbeke W, Baesens B. Data mining techniques for software effort estimation: a comparative study. IEEE Trans Software Eng. 2012; 38(2):375–97.
6. Chavoya A, Lopez-Martin C. Meda-Campana ME: applying genetic programming for estimating software development effort of short-scale projects. 2011 Eighth International Conference on Information Technology New Generations (ITNG2011). 2011; Las Vegas: IEEE. p. 174–79.
7. Gharehchopogh FS. Neural networks application in software cost estimation: a case study. 2011 International Symposium on Innovations in Intelligent Systems and Applications (INISTA 2011); 2011; Istanbul, Turkey: IEEE. p. 69–73.
8. Idri A, Khoshgoftaar TM, Abran A. Can neural networks be easily interpreted in software cost estimation?. Proceedings of the 2002 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2002); 2002; Honolulu, HI, USA. p. 1162–67.
9. Heiat A. Comparison of artificial neural network and regression models for estimating software development effort. Inform Software Tech. 2002; 44(15): 911–22.
10. Oliveira ALI. Estimation of software project effort with support vector regression. Neurocomputing. 2006; 69(13–15):1749–53.
11. Lo J-H. Early software reliability prediction based on support vector machines with genetic algorithms. 2010 the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA). 2010; Taichung. p. 2221–26.
12. Zeng X, Wu S, Han L. sensitivity-based adaptive learning rules for binary feedforward neural networks. IEEE Transactions on Neural Networks and Learning Systems. 2012; 23(3): 480–91.
13. Orr MJL. Introduction to Radial Basis Function Networks. Technical Report. Center for Cognitive Science, University of Edinburgh; 1996.
14. Moody J, Darken C. Fast learning in networks of locally-tuned processing units. Neural Computing. 1989; 1:281–94.
15. Spech D. A general regression neural network. IEEE Trans Neural Network. 1991; 2(6):568–79.
16. Idri A, Zahi A, Mendes E, Zakrani A. Software cost estimation models using rradial basis function neural networks. Software Process and Product Measurement. 2008; 21–31.
17. Shin M, Goel AL. Empirical data modeling in software engineering using radial basis functions. IEEE Trans Software Eng. 2000; 26(6):567–76.
18. Idri V, Mbarki S, Abran A. Validating and understanding software cost estimation models based on neural networks. 2004 International Conference on Information and Communication Technologies: From Theory to Applications; 2004. p. 433–34.
19. Yu H, Xie TT, Paszczynski S, Wilamowski BM. Advantages of radial basis function networks for dynamic system design. IEEE Trans Ind Electron. 2011; 58(12):5438–50.
20. Saberi M, Safaai D. Feature selection method using genetic algorithm for the classification of small and high dimension data. IEEE Trans Pattern Anal Mach Intell. 2005; 32(11):91–107.
21. Oh IS, Lee JS, Moon BR. Hybrid genetic algorithms for feature selection. IEEE Trans Pattern Anal Mach Intell. 2004; 26(11):1424–37.