

Software Cost Estimation Using Artificial Neural Network

Shaina Arora and Nidhi Mishra

Abstract Cost estimation is something we can call the most challenging task of software project management. Cost estimation is to precisely assess required assets and schedules for software improvement ventures, and it includes a number of things under its wide umbrella, for example, estimation of the effort required, estimation of the size of the software product to be produced, and last but not the least estimating the cost of the project. The overall project life cycle is impacted by the accurate prediction of the software development cost. Lots of models for software cost estimations are proposed by researchers. The COCOMO model makes employments of multilayer feedforward neural system while being actualized and prepared to utilize the perceptron learning algorithm. To test and prepare the system, the COCOMO dataset is actualized. Whatever result is generated from multilayer neural system is then compared with Kaushik [12]. This paper has the goal of creating the quantitative measure not only in the current model but also in our proposed model.

Keywords Software cost estimation · Artificial neural network · COCOMO Feedforward neural network · Magnitude relative error

1 Introduction

The estimation of required resources and schedule can be done through accurate cost estimation. Moreover, cost estimation can also relate as well as organize advancement of tasks apportion what an assets to perform the undertaking and by what technique these assets will be utilized. Talking about the parameters, the exactness of the product advancement and the precision of the administration

S. Arora (✉) · N. Mishra

Department of Computer Science & Engineering, Poornima University, Jaipur, India
e-mail: shaina.arora22@gmail.com

N. Mishra

e-mail: nidhi.mishra@poorinma.edu.in

© Springer Nature Singapore Pte Ltd. 2018

M. Pant et al. (eds.), *Soft Computing: Theories and Applications*,

Advances in Intelligent Systems and Computing 584,

https://doi.org/10.1007/978-981-10-5699-4_6

choices both are interrelated. The exactness of the previous is dependent on the precision of the last mentioned in terms of relying; that is, the former will rely on the latter. There are number of parameters, for example, improvement time, group and effort estimation, and for the calculation of each one of the models is required. COCOMO model is said to be the most popular models for the estimation of software cost effort estimation technique. What is unique about COCOMO model is that it makes use of mathematical formula to analyze project cost effort estimation. The paper based on COCOMO model is making use of multilayer neural network technique using perception algorithm.

2 COCOMO

The Constructive Cost Model or better known as the COCOMO model, first presented by Dr. Barry Boehm in 1981, surpassed all the software development practices that took place in those days. Software development techniques have been undergoing many changes and evolving since those days. The COCOMO model can be subdivided in the following models based on the type of application [1].

2.1 *Basic COCOMO*

Project quality details are not needed to implement parameterized equation of basic COCOMO model.

$$\text{Person Month} = a(\text{KLOC})^b \quad (1)$$

$$\text{Development Time} = 2.5 * \text{PM}^c \quad (2)$$

Three modes of progress of projects are there, on which all the three parameters depend, namely a, b, and c.

2.2 *Intermediate COCOMO*

According to basic COCOMO model, there is no provision to do software development. To add the accuracy in basic COCOMO model, there are 15 cost drivers provided by Boehm. Cost driver can be classified into

1. Product attributes

- Required software reliability or better known as RELY
- Database size or better known as DATA
- Product complexity or better known as CPLX

2 Computer attributes

- Execution time constraint or better known as TIME
- Main storage constraint or better known as STOR
- Virtual machine volatility or better known as VIRT
- Computer turnaround time or better known as TURN

3 Personnel attributes

- Analyst capability or better known as ACAP
- Application experience or better known as AEXP
- Programmer capability or better known as PCAP
- Virtual machine experience or better known as VEXP
- Programming language experience or better known as LEXP

4. Project attributes

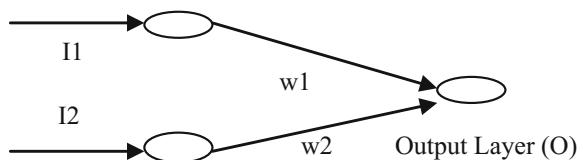
- Modern programming practices or better known as MODP
- Use of software tools or better known as TOOLS
- Required development schedule or better known as SCED [1].

3 Artificial Neural Network

Neurons or the interconnected basic processing devices are used to build artificial neural network (ANN). Much like the human brain, an ANN is stimulated by the way biological nervous systems are. What plays a key role in this paradigm is the novel structure of the input data processing system. It is constructed from number of highly interconnected processing elements (neurons) working in unification to resolve specific tribulations [2].

A simple artificial neural network is shown in Fig. 1 with two inputs neurons (I_1 , I_2) and one output neuron (O). The inter connected weights are given by w_1 and w_2 . Represented by the mathematical symbol $I(n)$, various inputs to the network are shown in Fig. 1. Each of these inputs is multiplied by a connection weight. These weights are represented by $w(n)$. In the simplest case, these products are simply summed, fed through a convey function to generate a result, and then delivered as output. This process lends itself to physical achievement on a large scale in a small

Fig. 1 A simple artificial neural net



package. This electronic implementation is still possible with other network structures, which utilize different calculation functions as well as different transfer functions [2].

4 Related Work

Researchers in effort estimation models have developed multiple software. Artificial neural network is capable of generating good information and modeling complex nonlinear relationships. For the calculation of software effort estimation, researchers all across the world have used the artificial neural network approach. Moreover, Boehm's COCOMO dataset is also used. Tadayon [3] reports the use of neural network with a backpropagation. Kaushik [1] also research on multilayer neural network using perceptron learning algorithm. COCOMO [4] is the most effective and widely used software for effort estimation model, which arranges beneficial expert knowledge. For getting appropriate calculations, COCOMO model is one of the most imperative tools that produce capacity for developing effort estimation models with better analytical accuracy. In this paper, multilayer feedforward neural network using perceptron learning algorithm and COCOMO dataset has been used. Using this approach, an effort estimation model for software cost evaluation has been proposed.

5 Proposed Neural Network Model

Figure 1 shows the simple artificial neural network [5]. Many of the network architectures have been developed for different applications and performance. Neural network configuration depends on its design and their performance constraint settings.

In neural network, several parameters are used such as:

- Number of nodes in input layer and hidden layer,
- Training algorithm parameters,
- Weights—Neurons are connected to each other by directed communication links, which are associated with weights.

In proposed model, 20 inputs are taken in which 15 are effort multipliers and 5 scale factors. We have initialized bias as 1.00, weight $w = 1.08$, and learning rate = 0.1 in our proposed network and used identity activation function to calculate the desired output of the network.

This describes about experimenting networks and calculating new set of weight. The proposed algorithm is as follow:

- Step 1. Initialize the bias ($b = 1$), weights ($W_i = 1$ for $i = 1$ to 15), set learning rate ($\text{lr} = 0.1$) and Threshold θ (Θ) value = 4. Set Learning Rate ($0 < N \leq 1$).
- Step 2. Execute steps 3-8 until condition is false.
- Step 3. Execute step 5-7 for each training pair.
- Step 4. The input layer gets information flag and transmit it to concealed layer by executing identity activation function on the inputs from $x = 1$ to 20.
- Step 5. Calculate the response of each response per unit is as follow:
- 5.1 Calculate effort multiplier hidden layer Using $\text{Hiddenem} = b + \text{input} * \text{weightem}$ for $\text{input}(x) = 1$ to 15.
 - 5.2 Calculate scale factor hidden layer using $\text{Hiddensf} = b + \text{input} * (\text{weightsf} + \text{size})$ for $\text{input}(x) = 16$ to 20.
- Step 6. Calculate estimated output i.e. effort using $\text{Estimated Output} = \text{Hiddenem} * \text{Weightem} + \text{Hiddensf} * \text{Weightsf}$. where $\text{Weightem} = \text{effort multiplier weight}$ and $\text{Weightsf} = \text{scale factor weight}$.
- Step 7. Weights are not updated. Go to step 10.
- Step 8. If the difference is acceptable limit the output is considered else weights are modified using $\text{weight}(\text{new}) = \text{weight}(\text{old}) + \text{learning rate} * \text{input}(x)$.
- Step 9. Repeat step 5 to 6.
- Step 10. Calculate Magnitude Relative Error (MRE) $\text{MRE} = ((O_{\text{act}} - O_{\text{est}}) / O_{\text{act}}) * 100$
- Step 11. Stop.

6 Evaluation Criteria and Results

In this area, we depict the procedure utilized for figuring endeavors and the outcomes get when actualizing proposed neural system model to the COCOMO information set [6]. COCOMO information set is open-source cost evaluating apparatus which comprises of 63 undertakings. The tool does the logical appraisal among the exactness of the assessed exertion with the genuine exertion. For examining programming exertion estimation, we have calculated error using Magnitude of Relative Error (MRE) which is characterized as follows:

$$\text{MRE} = (\text{actual effort} - \text{estimated effort} / \text{actual effort}) * 100 \quad (3)$$

14 experimental values are shown in Table 1 which were tested. Actual effort of the model has been compared with these values. The comparison reflects us about the efficiency of our network. Table 2 contains the estimated effort, actual effort, and Mean Magnitude of Relative Error (MRE) values for 14 experimented projects (Table 3).

Table 1 Cost drivers

Cost drivers			
Product attributes	Computer attributes	Personnel attributes	Project attributes
RELY	TIME	ACAP	MODP
DATA	STOR	AEXP	TOOLS
CPLX	VIRT	PCAP	SCED
	TURN	VEXP	
		LEXP	

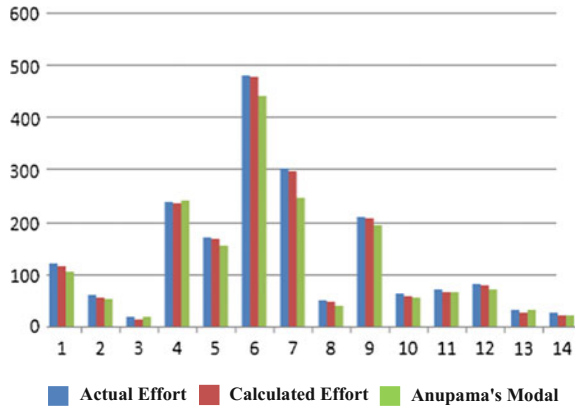
Table 2 Assessment of calculated effort

Project No.	Actual effort	Kaushik [1] model	Our proposed model
P1	120	104.17	116.33
P2	60	53.64	56.3
P3	18	18.69	14.33
P4	239	240	235.58
P5	170	154.87	166.52
P6	480	439.17	476.33
P7	300	245.39	296.33
P8	50	38.42	46.33
P9	210	194.64	206.32
P10	62	54.22	58.58
P11	70	66.74	66.58
P12	82	71.13	78.6
P13	31.2	30.14	27.3
P14	25.2	20.55	21.33

Table 3 Assessment of MRE (magnitude relative error)

Project No.	Actual effort	Calculated effort our proposed model	MRE using Kaushik [1] model	MRE using our proposed model
P1	120	116.33	13.19	3.1
P2	60	56.3	10.6	6.1
P3	18	14.33	3.83	20.4
P4	239	235.58	0.41	1.6
P5	170	166.52	8.9	2.3
P6	480	476.33	8.5	0.8
P7	300	296.33	18.20	1.2
P8	50	46.33	23.16	7.3
P9	210	206.32	7.31	1.8
P10	62	58.58	12.54	6.1
P11	70	66.58	4.65	5.4
P12	82	78.6	13.25	4.6
P13	31.2	27.3	3.39	12.8
P14	25.2	21.33	18.45	15.4

Fig. 2 Graphical representation of calculated effort



Tables 1, 2 and Fig. 2 show that the described neural network model gives the most proficient effort estimation results as compared to other models.

7 Conclusion

Having a dependable and precise estimate of software development has never been an easy task, and this is where has always lied the problem for many scholarly and industrial conglomerates since ages. Talking about anticipating the future programming improvement exertion, there are a number of product exertion estimating models that can be used for the purpose. This paper shows how a cost estimation model is built based on artificial neural network. The neural network that is used to estimate the software improvement effort is multilayer feedforward neural network with identity activation function. Accurate value is attained through neural network. In future, for software cost estimation, we will put our focus on neurofuzzy approach.

Acknowledgements I would like to express my deep gratitude and thanks to Dr Nidhi Mishra, Associate Professor, Department of Computer Engineering, Poornima University, IT Developer Devesh Arora and Prof. Pramod Choudhry for giving me an opportunity to work under his guidance for preparing the paper. Finally thanks to my family members Mr. Gopal Das (Father), Anju Arora (Mother), Abhishek (Brother), Shivani, Neeraj Munjal and my friend Hridya Narang for their constant encouragement and support throughout the research.

References

1. Kaushik, A., Chauhan, A., Mittal, D.: COCOMO estimates using neural networks. *Int. J. Intell. Syst. Appl.* 4(9), (2012)

2. Sivanandam, S.N., Deepa, S.N.: Introduction to Neural Networks using MATLAB 6.0. Tata McGraw-Hill Education, US (2006)
3. Tadayon, N.: Neural network approach for software cost estimation. In: IEEE International Conference on Information Technology: Coding and Computing (ITCC'05), vol. 2, pp. 815–818,4–6 (2005)
4. Sharma, T.: A comparative study of COCOMO II and Putnam models of software cost estimation. *Int. J. Sci. Eng. Res.* **2**(11), (2011)
5. Krenke, A., Bešter, J., Kos, A.: Introduction to the artificial neural networks. In: *Methodological Advances and Biomedical Applications* (2011)
6. <http://promise.site.uottawa.ca/SERepository/datasets/cocomo81.arff>
7. Reddy, C., Raju, K.V.S.V.N.: A concise neural network model for estimating software effort. *Int. J. Recent. Trends. Eng.* **1**(1), (2009)
8. Bayindir, R., Colak, I., Sagirolu, S., Kahraman, H.T.: Application of adaptive artificial neural network method to model the excitation currents of synchronous motors. In: 11th International Conference on in Machine Learning and Applications (ICMLA), vol. 2, pp. 498–502 (2012)
9. Kumari, S.: Performance analysis of the software cost estimation methods: a review. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **3**(7), (2013)
10. Sharma, T.: Statistical analysis of various models of software cost estimation. *Int. J. Eng. Res. Appl.* **2**(3), 683–685 (2012)
11. Kaushik, A., Soni, A.K., Soni, R.: A simple neural network approach to software cost estimation. *Glob. J. Comput. Sci. Technol. Neural Artif. Intell.* **13**(1), (2013) (Version 1.0)
12. Hamza, H., Kamel, A., Shams, K.: Software effort estimation using artificial neural networks. A Survey of the Current Practices. In: Tenth International Conference on Information Technology: New Generations (ITNG), pp. 731–733, 15–17 (2013)
13. Mukherjee, S.: Optimization of project effort estimate using neural network. In: IEEE International Conference on Advanced Communication Control and Computing Technologies, pp. 406–410 (2014)
14. Jodpimai, P., Lursinsap, P.C.: Estimating software effort with minimum features using neural functional approximation. In: International Conference on Computational Science and Its Applications (ICCSA), pp. 266–273, 23–26 (2010)
15. Ghose, M.K., Bhatnagar, R., Bhattacharjee, V.: Comparing some neural network models for software development effort prediction. In: National Conference on Emerging Trends and Applications in Computer Science (NCETACS), pp. 1–4, 4–5 (2011)