I J C T A, 9(18) 2016, pp. 8919-8925
© International Science Press

Software Development Time Estimation Using Neuro- Fuzzy Approach: A Review

Megha Sharma*, Vineet Sharma* and Parita Jain*

ABSTRACT

If we have to make a project efficiently and effectively, it is essential for any organization that the project should be completed under the budget, on time and the project should have mandatory excellence. Software processes should be managed efficiently and that requires a measurement of quantification and modelling. The pitiable causes for any software failure may be related to the inefficient estimation of resources and the resources may be like effort, cost , development time etc. So it should be managed effectively and efficiently. In this paper, we deals with a hybrid approach by combining neural networks and fuzzy i,e The Adaptive Neuro-Fuzzy approach for the development time estimation of a software . The proposed approach is based on the estimation of development time of a software and the results will be compared with the existing system.

Keywords: software estimation, ANFIS, development time estimation, cost and effort estimation.

I. INTRODUCTION

There are many forms of estimations for a software, for making it effective and efficient. the types of estimations for a software are cost estimation, effort estimation, development time estimation. Here, we will estimate the development time of the software by knowing the fact that:-

what does the developmnt time of a software?

what does the development time estimation refers to?

what is the need to estimate development time for a software?

Development time:- The development time of a software is the time that a software takes for its successful completion in terms of cost, effort, maintainence. [1]

Need to estimate the development time of a software:- The development time of a software needed to be estimated so that the software can be completed on time and within the budget and before the given deadline by the customer. If any software is not completed on time, the software may suffer from many failures. So,to satisfy the customer, the completion of the software should takes place on time.

The development time in making or developing a software project plays a vital and necessary role. The goals and objectives of any software can be meet when it is completed on time with a proper budget. The software engineering deals with the many areas like: Requirement design, construction, testing and management. Here, we represent the Software Development Time Estimation using the neuro-fuzzy models. Many of the projects in software engineering fails due to the bad estimation of resources like effort in terms of money and the labours required. When the estimation of these resources is incorrect the software project faces a failure. In this era of growing software industries, the development of a software should take place on time and with a proper budget so as to achieve the goals and set objectives of the organization.

* Department of Computer Science and Engineering, Krishna Institute of Engineering and Technology, Ghaziabad, U.P, *E-mail: sharmamegha027@gmail.com*; *vineet.sharma@kiet.edu; paritajain23@gmail.com*

The software development time estimation should be done with statistics, and the data should be presented in a proper form. With the present and past statistics we can come to know that how much improvement should be done in the previous data so that the results are In the favor of the organization. The reason that the development time should be estimated is that sometimes the customer gives the deliverable deadline to the customer and so to satisfy the customer requirements the project should be completed and delivered on time.

The fuzzy logic was started in 1965 by Zadeh for the representation or manipulation of data and the information that possess non statistical uncertainties. Fuzzy systems has two main characteristics that provides some better performance for specific applications.

- These systems work better in case of uncertain of approximate facts or reasoning, and especially for the system that has mathematical model that is difficult to be functioned or operate.
- It allows decision making with estimated values in case of incomplete or uncertain data.

The neural networks were started by Warren McCulloch and Walter Pitts in 1943.They created a mathematical model and algorithms for the neural networks. There are two approaches in which the neural networks can be split:-

- The first approach deals with the biological processes in the brain.
- And the second one focus on the neural networks applications in artificial intelligence. The approach that is used here is the combination of the fuzzy systems and neural networks that is called the neuro-fuzzy approach. It takes the advantage of both, the fuzzy systems and the neural networks to achieve the maximum efficiency.

Neural networks are used to tune membership functions of fuzzy systems that are employed as decision-making systems for controlling equipment. Although fuzzy logic can encode expert knowledge directly using rules with linguistic variables. The different approaches by which the time can be estimated are- 1) Top-down estimation .2) Bottom-up estimation. 3) Comparative estimation and 4) Three point estimation. The hybrid approach that is used for estimation provides the strength to the incorrect and uncertain inputs.

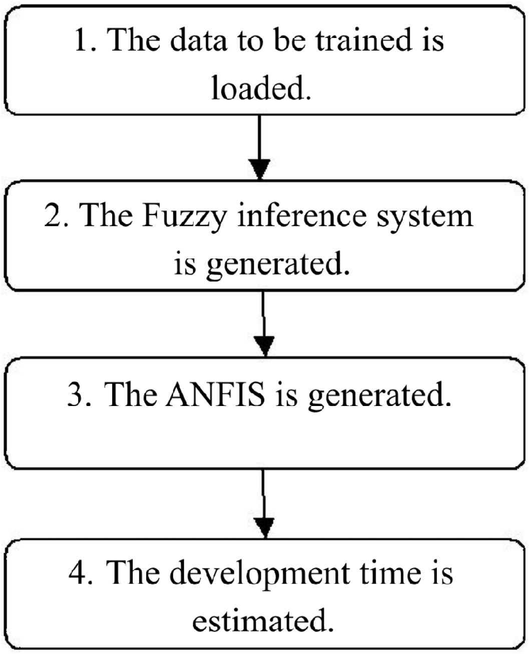
Here, an Adaptive Neuro Fuzzy Inference System(ANFIS)[2] is used for the software development time estimation. The comparison of the results of various neural networks model will be compared using the different types of parameters of the relative errors. Adaptive Neuro Fuzzy Inference System (ANFIS) ia a grouping of Fuzzy Logic and Neural Network so ANFIS performs the best operation taking the best from both neural networks and fuzzy logic. A fuzzy inference system is constructed by ANFIS with the help of training data set that is given and its membership function parameters are adjusted by least square or back propagation algorithm or the combination of both.

II. RELATED WORK

Shina Dhingra[1] presented an adaptive Neuro fuzzy approach for software development time estimation[1]. Lopez Martin et al.[9] proposed a fuzzy logic model for development time estimation. Ting su et al.[20] described an enhanced fuzzy logic model for the estimation of software development effort which had the similar capabilities as the previous fuzzy logic model in addition to enhancements in empirical accuracy in terms of MMRE. The main goal of this paper is to evaluate software development time using an adaptive Neuro fuzzy approach by applying some parameters that are different from the existing or applied approach and getting more accurate results.

III. METHODOLOGY

We have to estimate the development time of a software using an adaptive neuro fuzzy approach. The learning algorithm is used for training the network. The learning algorithm is the combination of the back propagation and the least mean square algorithm. The procedure of training has the four steps involved in it. The steps are:



1. The data to be trained is loaded.

2. The Fuzzy inference system is generated.

3. The ANFIS is generated.

4. The development time is estimated.

3.1 ANFIS Method

The proposed method accurately estimates the software development time by the Adaptive Neuro Fuzzy Inference System(ANFIS)[1][8]. The ANFIS was developed in early 1990s. It has the potential to combine the benefits of the neural networks and the fuzzy systems in a single framework. ANFIS can be considered as a universal estimator.

ANFIS is a class of of adaptive networks that are equivalent functionally to fuzzy inference systems. This hybrid learning algorithm is used in ANFIS.

The ANFIS structure is presented below consisting of five layers performing their corresponding functions

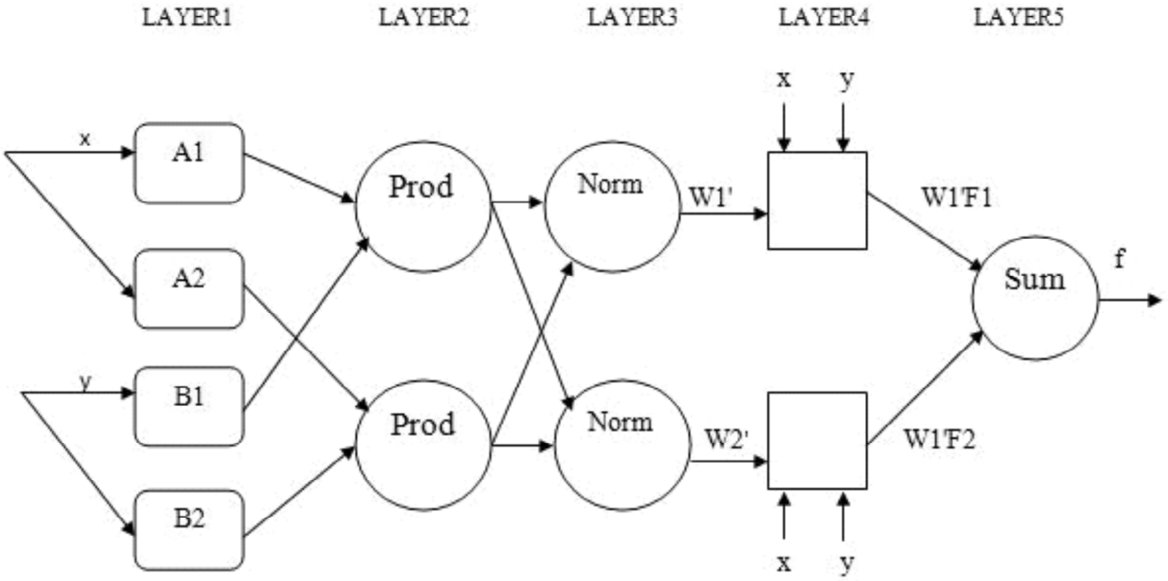


Figure1: ANFIS Architecture

LAYER1:

a) The output of the ith node of the layer1 is $O_{1,i}$.

b) Each node i of the layer1 is having a node function and is called an adaptive node.

$O_{1,i}$ = $\mu Ai(x)$ for i = 1, 2….. or
$O_{1,i}$ = $\mu B_{i”2}(x)$ for i = 3, 4….

c) The input node i is x or y and $_{Ai}$ is a linguistic label associated with this node.

d) So, the fuzzy set (A1,A2,B1,B2) has the membership grade $O_{1,i}$.

e) The typical membership function is:

$$\mu_a(x) = \frac{1}{1 + \frac{|x - c_i|}{|a_i|} * 2b_i}$$

a_i, b_i, c_i is the parameter set.

f) Parameters are called as the Premise parameters.

LAYER2

a) In this layer, every node is a fixed node with name “prod”. b)Product of all the incoming signals is the output of this layer.

$O_{2,i}$ = Wi= $\mu Ai(x).\mu Bi(y)$, i = 1,2….

c) Fire strength of the rule is represented by each node.

d) Any other T-norm operator that perform the AND operator can be used.

LAYER 3

a) In this layer, every node is a fixed node with name “norm”.

b) The ith node calculates the ratio of the i^{th} rule’s firing strength to the sum of all rule’s firing strengths.

c) $$O_{3,i} = Wi' = \frac{W_i, i = 1,2}{W_1 + W_2}$$

d) Outputs are called normalized firing strengths.

LAYER 4

a) Each node i of the layer4 is having a node function and is called an adaptive node. $O_{4,i}$ = W_i’F_i= Wi’ (P_x + Q_{iy} + R_i)

b) Wi’ is the normalized firing strength from layer 3.

c) {Pi, Qi, Ri} is the parameter set of this node.

d) And called as consequent parameters.

LAYER 5

a) There is a single and fixed node labelled as “sum” that evaluates the total output as the summation of all the output signals coming from each layer as:

total output = $O_{5,i} = \Sigma_i W_i' F_i = \dfrac{\Sigma i Wi Fi}{\Sigma i Wi}$

3.2. Performance Evaluation Metrics

The evaluation metrics[10] that can be used to evaluate the performance of the estimation methods are as follows:

I. Magnitude of relative error :-

MRE= $\dfrac{|\text{Actual Time- Estimated Time}|}{\text{Actual Time}}$ *100

II. Mean Magnitude of relative Error(MMRE):-

$$MMRE = 1/n * \Sigma \frac{|\,Actual\,Time - Estimated\,Time\,|\;(for\,i = 1\,to\,n)}{Actual\,Time}$$

III. Balanced Relative Error(BRE):-BRE(%)=

$\dfrac{|\text{Estimated Time-Actual Time}|}{\min(t,t')}$ *100

Where, t=estimated time and t'=actual time

IV. Prediction level(Pred):-

PRED(l)= k/n *100

Where, l is the maximum MRE of a selected range, n is the total number of projects

k is the number of projects in a set of n projects.

V. Median Magnitude of Relative Error (MdMRE):

MdMRE= median (MRE)

VI. Root Mean Square Error(RMSE):

It is just the square root of the mean square error which measures the average magnitude of errror.

$$RMSE = \sqrt{1/n * \Sigma(E_t - A_t)^2}\;\;(For\,t = 1\;to\,n)$$

VII. Mean Absolute Relative Error (MARE):

It usually expresses accuracy as a percentage, and is defined by the formula:

$$\frac{1}{n}\sum_{t=1}^{n}\frac{A_t - E_t}{A_t}$$

where A_t is the actual value and E_t is the forecast value.

3.3. Algorithm Used In Training

The ANFIS system uses the two methods or two learning algorithm either the Back Propagation Algorithm or Least Square Method or may be the combination of both the algorithms i.e, The hybrid algorithm of back Propagation and the Least Square Method.

Mostly we use the hybrid approach i.e the combination of the Back Propagation and the Least Square method for training the data set in the ANFIS. This approach is used for training because it minimizes the approximation error and to overcome the certain problems.

There are two passes in the algorithm:

1) Forward Pass
2) Backward pass

- In, Forward Pass the Least Square method is used for identifying the consequent parameters in layer4.
- During the backward pass, the Gradient descent updates the premise parameters and the errors are propagated backward.

In this hybrid training approach, to create a training stopping criterion, the error tolerance is used, and error size is also related to it. When the training data error remains under the tolerance, after that the training process stops. The model parameters that are associated with the minimum checking error, are choosen by the anfis method.

Table 1
Passes in the Algorithm

	Forward Pass	*Backward Pass*
Precise parameters	Fixed	Gradient Descent
Consequent parameters	Least-Square estimators	Fixed
Signals	Node outputs	Error Signals

IV. PERFORMANCE EVALUATION FOR THE DEVELOPMENT TIME ESTIMATION

Author Name	*Project Name*	*Problem Statement*	*Conclusion*
Shina Dhingra and Palvinder Singh Mann	Design and Implementation of Neuro- Fuzzy model for Software Development Time Estimation[1]	The approach is meant at the making and evaluation of a Neuro-fuzzy model using three membership functions for software development time.	The conclusion or the result that is observed that the best results were obtained from the Neuro fuzzy model using the trapezoidal membership function.
Cuauhtémoc López Martín, Jérôme Leboeuf, Cornelio YanezM. and Agustin Gutierrez T.	Software Development Effort Estimation Using Fuzzy Logic[11]	It describes an application where the results of application are compared to the multiple regression results.	The results of this application shows that fuzzy logic can be used as an alternative for estimating the software development effort .
Moon Ting Su, Teck Chaw Ling, Keat Keing Phang , Chee Sun Liew, Peck Yen Man	Enhanced Software Development Effort and cost estimation using fuzzy logic model[20]	The paper describes an enhanced fuzzy logic model called FLECE that has same capabilities as the previous fuzzy logic model in addition that, the enhancement that is done using this model improve the empirical accuracy of the previous models in terms of MMRE and PRED.	The results shows that FLECE obtains more accurate results in software development effort estimation when it is compared to the previous fuzzy logic model.

V. CONCLUSION AND FUTURE WORK

The software development time estimation can be done by the use of various parameters such as MRE, MMRE,BRE, Prediction, etc and including some appropriate membership functions that will give the best results for estimation of development time. Our future work deals with the comparison of results got from applying various membership functions with the help of Neuro-fuzzy approach and calculating the more accurate results from the previous observations.

REFERENCES

[1] Shina Dhingra and Palvinder Singh Mann. Article: Design and Implementation of Neuro Fuzzy model for Software Development Time Estimation. *International Journal of Computer Applications* 86(5):7-12, January 2014.

[2] Pillai, Krichnakumar, and VS Sukumaran Nair. "A model for software development effort and cost estimation." *IEEE Transactions on Software Engineering* 23.8 (1997): 485-497.

[3] Agrawal, Vidisha, and Vishal Shrivastava. "Performance Evaluation of Software Development Effort Estimation Using Neuro-Fuzzy Model." (2015).

[4] Borade, Jyoti G., and Vikas R. Khalkar. "Software Project Effort and Cost Estimation Techniques." *International Journal of Advanced Research in Computer Science and Software Engineering* 3.8 (2013): 730-739.

[5] Marza, Venus, Amin Seyyedi, and Luiz Fernando Capretz. "Estimating development time of software projects using a neuro fuzzy approach." *World Academy of Science, Engineering and Technology* 46 (2008): 575-579.

[6] Dave, Vachik S., and Kamlesh Dutta. "Comparison of regression model, feed-forward neural network and radial basis neural network for software development effort estimation." ACM SIGSOFT Software Engineering Notes36.5 (2011): 1-5.

[7] Boehm, Barry, Chris Abts, and Sunita Chulani. "Software development cost estimation approaches—A survey." Annals of software engineering 10.1-4 (2000): 177-205.

[8] Loganathan, C., and K. V. Girija. "Hybrid learning for adaptive neuro fuzzy inference system." Research Inventy: International Journal Of Engineering And Science 2.11 (2013): 06-13.

[9] Martin, Cuauhtemoc Lopez, et al. "Software development effort estimation using fuzzy logic: a case study." Sixth Mexican International Conference on Computer Science (ENC'05). IEEE, 2005.

[10] Malathi, S., and S. Sridhar. "Analysis Of Size Metrics And Effort Performance Criterion In Software Cost Estimation." 2012 (2012).

[11] Cuauhtémoc López Martín, "Software Development Effort Estimation Using Fuzzy Logic: A Case Study", Proceedings of the Sixth Mexican International Conference on Computer Science (ENC'05), IEEE 2005.

[12] Moataz A. Ahmed, Moshood Omolade Saliu and Jarallah AlGhamdi, "Adaptive fuzzy logic-based framework for software development effort prediction", Information and Software Technology, 2005.

[13] Divya Kashyap, Ashish Tripathi,Prof. A.K.Mishra, "Software Development Effort and Cost Estimation: Neuro-Fuzzy Model", IOSR Journal of Computer Engineering (IOSRJCE), Vol. 2, pp. 12-14, July-Aug. 2012.

[14] A.Heiat, "Comparison of artificial neural network and regression models for estimating software development effort", Information and Software Technology, Vol.44,Issue 15,2002, pp. 911-922.

[15] Urvashi Rahul Saxena, S.P.Singh, "Software Effort Estimation Using Neuro-Fuzzy Approach", CSI 6th International Conference, pp.1-6, Sept.2012.

[16] J. Jantzen, "Neuro- Fuzzy modeling", Reort no 98-H-874, 1998.

[17] Xishi Huang et al., "A Neuro-Fuzzy Model for Software Cost Estimation", Proceedings of the Third International Conference on Quality Software (QSIC'03), IEEE, 2003.

[18] V. Khatibi, Dayang and N. A. Jawawi, " Software Cost Estimation Methods: A Review", Journal of Emerging Trends in Computing and Information Sciences, CIS Journal, Vol. 2, no. 1, ISSN 2079-8407, 2011.

[19] N. Sharma1, A. Bajpai and M. R. Litoriya, "A Comparison of Software Cost Estimation Methods: A Survey", The International Journal of Computer Science and Applications (TIJCSA), Vol.1, no. 3, ISSN – 2278 – 1080, May 2012.

[20] Ting, Su Moon. "Enhanced Software Development Effort and Cost Estimation Using Fuzzy Logic Model." Malaysian Journal of Biochemistry and Molecular Biology 20.2 (1970).