International Journal of Computer Theory and Engineering, Vol. 2, No. 1 February, 2010
1793-8201

Neural Network-A Novel Technique for Software Effort Estimation

Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi

Abstract— **Estimating software development effort is an important task in the management of large software projects. The task is challenging and it has been receiving the attentions of researchers ever since software was developed for commercial purpose. A number of estimation models exist for effort prediction. However, there is a need for novel model to obtain more accurate estimations. The primary purpose of this study is to propose a precise method of estimation by selecting the most popular models in order to improve accuracy. In this paper, we explore the use of Soft Computing Techniques to build a suitable model structure to utilize improved estimation of software effort for NASA software projects. A comparison between Artificial-Neural-Network Based Model (ANN) and Halstead, Walston-Felix, Bailey-Basili and Doty models were provided. The evaluation criteria are based upon MRE and MMRE. Consequently, the final results are very precise and reliable when they are applied to a real dataset in a software project. .The results show that ANNs are effective in effort estimation.**

Index Terms—**Effort Estimation, Neural Network, Halstead Model, Walston-Felix Model, Bailey-Basili Model, Doty Model.**

I. Introduction

In the last three decades, many quantitative software cost estimation models have been developed. According to Marco, Reliable prediction of size and effort in software development projects is a necessary prerequisite to developing reliable cost and schedule estimates. The size and development effort measures, such as function points or lines of code developed per person-month, act as technical productivity and performance indicators that facilitate the tracking and control of software developments.

An empirical model uses data from previous projects to evaluate the current project and derives the basic formulae from analysis of the particular database available. An analytical model, on the other hand, uses formulae based on global assumptions, such as the rate at which developer solve problems and the number of problems available. Evaluation of many software models were presented in [1], [2], [3]. Numerous models were explored to provide better effort estimation [4], [5], [6], [7]. In [8], [9], authors provided a survey on the effort and cost estimation models.

Typical major models that are being used as benchmarks for software effort estimation are:

- Halstead,
- Walston-Felix
- Bailey-Basili
- Doty (for KLOC > 9)

These models have been derived by studying large number of completed software projects from various organizations and applications to explore how project sizes mapped into project effort. But still these models are not able to predict the Effort Estimation accurately.

As the exact relationship between the attributes of the effort estimation is difficult to establish so a Neural Network approach could serve as an automatic tool to generate model by formulating the relationship based on its training. When one designs with Neural Networks alone, the network is a black box that needs to be defined; this is a highly compute-intensive process. One must develop a good sense, after extensive experimentation and practice, of the complexity of the network and the learning algorithm to be used.

As Neural based system is able to approximate the non-linear function with more precision and non of the researcher have explored Neuro approach for the Effort Estimation and there is still scope of exploring more statistical modeling approaches. So, in this proposed study, it is tried to use Neural Network Based Approach to build a more accurate model that can improve accuracy estimates of effort required to build a software system.

In this paper, however, the main focus is on investigating the accuracy of the predictions using ANN-based model. A study was performed to examine the potential of above given approaches and neural network based approach.

II. Proposed Methodology

The following steps are used for the comparative study:

A. Preliminary Study

First, Survey of the existing Models of Effort Estimation that are discussed in the literature.

Manuscript received August 12, 2009.

Jaswinder Kaur is doing M.Tech. from department of Computer Science & Engineering & I.T. of Baba Banda Singh Bahadur Engineering College, Fateh Garh Sahib, Punjab, India.

Satwinder Singh is working as Lecturer in department of Computer Science & Engineering & I.T. of Baba Banda Singh Bahadur Engineering College, Fateh Garh Sahib, Punjab, India.

Dr. Karanjeet Singh Kahlon is working as Professor with the Computer Science & Engineering Department, Guru Nanak Dev University, Amritsar, Punjab, India.

Pourush Bassi is working as Lecturer in department of Computer Science & Engineering of Rayat Bahra Institute of Engineering & Bio-Technology, Sahauran, Mohali, Punjab, India

B. Data Collection

Collect the historical software estimation data so that the same data can be used for experimentation evaluation.

C. Calculate Effort using Different Modes

The following models are used for the data collected in the previous step and calculate the effort for each developed model:

- Halstead,
- Walston-Felix
- Bailey-Basili
- Doty (for KLOC > 9).
- Neural Network Based System.

Over the last 20 years, research efforts have focused on the development of techniques that are quantitatively based, in an effort to remove or reduce subjectivity in the estimation process. However, other techniques for the exploratory data analysis, such as clustering, case-based reasoning and ANN have been effective as a means of predicting software project effort. [11] describe the use of clustering to predict software quality. Wittig and Finnie [20] describe their use of back propagation learning algorithms on a multilayer perceptron in order to predict development effort. An overall error rate (MMRE) obtained which compares favorably with other methods

In this work, the ANN methodology is used to predicting software development effort (in man-hour) from the project size (given by the amount of source code lines).

Khoshgoftaar [13] presented a case study considering real time software to predict the testability of each module from source code static measures. They consider ANNs as promising techniques to build predictive models, because they are capable of modeling non linear relationships.

ANNs are massively parallel systems inspired by the architecture of biological neural networks, comprising simple interconnected units (artificial neurons). The neuron computes a weighted sum of its inputs and generates an output if the sum exceeds a certain threshold. This output then becomes an excitatory (positive) or inhibitory (negative) input to other neurons in the network. The process continues until one or more outputs are generated.

Figure 1 shows an artificial neuron that computes the weighted sum of its n inputs, and generates an output of y. The neural network results from the arrangement of such units in layers, which are interconnected one to another. The resulting architectures solve problems by learning the characteristics of the available data of related to the problem

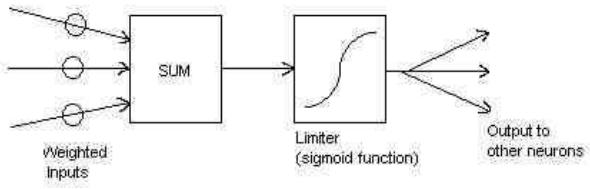

Figure1 Computations of a Neuron

For this study, backpropagation artificial neural network models were used. Backpropagation networks are the most generalized neural networks currently in use (Nelson & Illingworth, 1991). The backpropagation network requires data from which to learn. To learn the network calculates the error, which is the difference between the desired response and the actual response, and a portion of it is propagated backward through the network. At each neuron in the network the error is used to adjust weights and threshold values of the neuron, so that at the next epoch the error in the network response

D. Performance Criteria

Perform the comparison of the models on basis of:

- Mean Magnitude of Relative Error (MMRE)
- Root Mean Square Error (RMSSE)

RMSSE is frequently used measure of differences between values predicted by a model or estimator and the values actually observed from the thing being modeled or estimated. It is just the square root of the mean square error as shown in equation given below:

$$RMSE = \sqrt{\frac{(a_1-c_1)^2+(a_2-c_2)^2+\ldots+(a_n-c_n)^2}{n}} \quad (1)$$

The mean-squared error is one of the most commonly used measures of success for numeric prediction. This value is computed by taking the average of the squared differences between each computed value and its corresponding correct value. The root mean-squared error is simply the square root of the mean-squared-error.

The literature considered the mean magnitude of relative error (MMRE) as the main performance measure. The value of an effort predictor can be reported many ways including MMRE. MMRE is computed from the relative error, or RE, which is the relative size of the difference between the actual and estimated value.

Given a data set of size "D", a "Training set of size "(X=|Train|) <= D", and a "test" set of size "T=D-|Train|", then the mean magnitude of the relative error, or MMRE, is the percentage of the absolute values of the relative errors, averaged over the "T" items in the "Test" set.

In other words, *RMSSE* is frequently used measure of differences between values predicted by a model or estimator and the values actually observed from the thing being modeled or estimated. It is just the square root of the mean square error as shown in equation given below:

$$RMSSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2} \quad (2)$$

Where yi represents the ith value of the effort and is the estimated effort.

MMRE is another measure and is the percentage of the absolute values of the relative errors, averaged over the N items in the "Test" set and can be written as:

$$MMRE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i-\hat{y}_i\right| \Big/ y_i \quad (3)$$

III. Results and Discussion

The dataset of NASA [10] is used for the comparison of different models. In this dataset, there is empirical data in terms of DKLOC, Methodology and Effort values of 18 projects as shown in table I.

TABLEI. NASA DATA [10] OF EFFORT ESTIMATION

Project No.	KDLOC	Methodology	Actual Effort
1	90.2	30	115.8
2	46.2	20	96
3	46.5	19	79
4	54.5	20	90.8
5	31.1	35	39.6
6	67.5	29	98.4
7	12.8	26	18.9
8	10.5	34	10.3
9	21.5	31	28.5
10	3.1	26	7
11	4.2	19	9
12	7.8	31	7.3
13	2.1	28	5
14	5	29	8.4
15	78.6	35	98.7
16	9.7	27	15.6
17	12.5	27	23.9
18	100.8	34	138.3

TABLEII. ACTUAL AND CALCULATED EFFORT USING DIFFERENT EFFORT ESTIMATION MODELS

Actual Effort	NN System Effort	Halstead Model Effort	Walston-Felix Model Effort	Bailey-Basili Model Effort	Doty Model Effort
8.4	7.9455	7.8262	22.494	10.222	28.518
98.7	98.9744	487.79	275.95	120.85	510.27
15.6	14.6092	21.147	41.112	15.685	57.074
23.9	19.3829	30.936	51.783	19.169	74.431
138.3	99.5649	708.42	346.06	189.43	662.09

TABLEIII. ERRORS IN CALCULATED EFFORT USING DIFFERENT EFFORT ESTIMATION MODELS

Perfor-mance Criteria	Model Used				
	NN System	Halstead Model	Walston -Felix Model	Bailey-Basili Model	Doty Model
MMRE	11.7896	175.655	155.559	20.2885	302.502
RMSSE	17.4475	308.7097	123.457	25.0224	299.474

The data of first 13 projects is used as training data for the Neural Network and data of last 5 projects is used as testing data of the trained Neural Network. The neural network used is backpropagation based Neural Network that consists of two neurons in input layer, two neurons in the hidden layer and one neuron in the output layer. In the testing phase the calculated efforts and errors using different models is shown in table II and table III respectively. As evident from the table II, the predicted values of the efforts is very close to the expected or actual values e.g. in the second test case using Neural Network approach the predicted value of the effort is 98.9744, which is very close to the actual value i.e. 98.7. This thing is evident with very less values of MMRE and RMSSE as error values in prediction of test cases.

IV. Conclusion

The performance of the Neural Network based effort estimation system and the other existing Halstead Model, Walston-Felix Model, Bailey-Basili Model and Doty Model models is compared for effort dataset available in literature [10]. The results show that the Neural Network system has the lowest MMRE and RMSSE values i.e. 11.7896 and 17.4475 respectively. The second best performance is shown by Bailey-Basili software estimation system with 20.2885 and 25.0224 as MMRE and RMSSE values. Hence, the proposed Neuro based system is able to provide good estimation capabilities. It is suggested to use of Neuro based technique to build suitable generalized type of model that can be used for the software effort estimation of all types of the projects.

References

[1] M. Boraso, C. Montangero, and H. Sedehi, "Software cost estimation: An experimental study of model performances," tech. report, 1996.
[2] T. Menzies, D. Port, Z. Chen, J. Hihn, and S. Stukes, "Validation methods for calibrating software effort models," in ICSE '05: Proceedings of the 27th international conference on Software engineering, (New York, NY, USA), pp. 587–595, ACM Press, 2005.
[3] O. Benediktsson, D. Dalcher, K. Reed, and M. Woodman, "COCOMO based effort estimation for iterative and incremental software development," Software Quality Journal, vol. 11, pp. 265–281, 2003.
[4] S. Devnani-Chulani, "Modeling software defect introduction," tech. report.
[5] B. Clark, S. Devnani-Chulani, and B. Boehm, "Calibrating the COCOMO II post-architecture model," in ICSE '98: Proceedings of the 20th international conference on Software engineering, (Washington, DC, USA), pp. 477–480, IEEE Computer Society, 1998.
[6] S. Chulani and B. Boehm, "Modeling software defect introduction and removal: Coqualmo (constructive quality model)," tech. report.
[7] S. Chulani, B. Boehm, and B. Steece, "Calibrating software cost models using bayesian analysis," IEEE Trans. Software Engr., July-August 1999, pp. 573–583, 1999.
[8] M. Shepper and C. Schofield, "Estimating software project effort using analogies," IEEE Tran. Software Engineering, vol. 23, pp. 736–743, 1997.
[9] G. Witting and G. Finnie, "Estimating software development effort with connectionist models," in Proceedings of the Information and Software Technology Conference, pp. 469–476, 1997.
[10] J. W. Bailey and V. R. Basili, "A meta model for software development resource expenditure," in Proceedings of the International Conference on Software Engineering, pp. 107–115, 1981.
[11] P. Sentas, L. Angelis, I. Stamelos, and G. Bleris, "Software productivity and effort prediction with ordinal regression," Journal Information and Software Technology, 2005, no. 47, pp.17-29.
[12] G. Witting, and G. Finnie, "Using Artificial Neural Networks and Function Points to Estimate 4GL Software Development Effort", J. Information Systems, 1994, vol. 1, no. 2, pp. 87-9
[13] T. M. Khoshgoftaar, E.B. Allen, and Z. Xu, "Predicting testability of program modules using a neural network," Proc. 3rd IEEE Symposium on Application-Specific Systems and Sof. Eng. Technology, 2000, pp. 57-60.