International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC) 2011
Proceedings published by International Journal of Computer Applications® (IJCA)

Application of Feed-Forward Neural Network in Estimation of Software Effort

Vachik S. Dave
Research scholar, CSE Dept.
National Institute of Technology,
Hamirpur (H.P.), India.

Dr. (Mrs.) Kamlesh Dutta
Associate Professor, CSE Dept.
National Institute of Technology,
Hamirpur (H.P.), India.

ABSTRACT
In current scenario of software industries, Software effort estimation is very important task for software manager for successful completion of the project. Prediction is always challenging task and in recent days effort estimation take many researcher's attention. Prediction with more accuracy is also an important for prediction models. We use Feed-Forward Neural Network for software development effort estimation. In this paper we have shown that, Feed-Forward Neural Network gives much better results than other prediction models. The simulated results shows that, simple Neural network model predict software development effort more accurately.

Keywords
Feed-Forward Neural Network, Software Develop Effort, Effort Estimation, Neural Network.

1. INTRODUCTION
In current situation of software industries, successful project completion within time is most important task for any industry. The management point of view effort prediction is complicated task. The average effort overrun appears to be 30-40 percent. Similar findings are reported for schedule overruns, with 65-80 percent of all projects facing overruns of the delivery date [6]. Effort overrun directly proportional to cost overrun, so accurate effort prediction is also important.

There are a lot many models available for prediction of software development effort and cost. COCOMO (COnstructive COst MOdel) is most commonly used model. But machine learning methods for software prediction are more appropriate, because they are more adaptable [2]. When we are talking about specifically software development effort prediction problem, the output (effort value) of the system is very complexly dependent on input parameters, such as size of the problem, experience and many other. Now this complex relationship cannot be described or expressed using simple mathematical equations. In such situations neural network is more suitable to use, the reason can easily understand if we see at architecture of neural networks shown in the next section.

Regression Analysis is more frequently used as prediction model from last two decades. Here in this paper authors have compared simulated results for software development effort estimation using regression analysis approach and Neural Network (NN) model. The evaluation criteria used for comparison are Mean Magnitude Relative Error (MMRE), Relative Standard Deviation (RSD) and Relative Mean Squared Error (MSE).

This paper contains 6 sections. Second section gives brief overview of Feed-Forward NN (FFNN). Third section contains Regression analysis introduction. Fourth section is experimental study in which we have mentioned conducted experiment, its results and evaluation of related models. Fifth section is conclusion and future work. Finally we have referances.

2. FEED-FORWARD NEURAL NETWORK
An Artificial Neural Network (ANN) is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. An ANN can be configured for a specific application, such as pattern recognition or data classification, through a learning process. ANNs incorporate the two fundamental components of biological neural networks:

- Neurons (nodes)
- Synapses (weights)

A neuron has a set of n synapses associated to the inputs. Each of them is characterized by a weight.

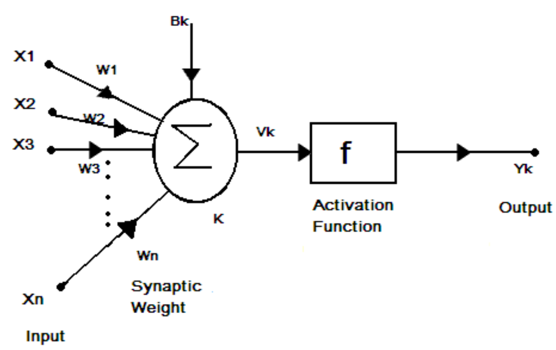

Fig 1: An Artificial Neuron

An artificial neural network is composed of many artificial neurons that are linked together according to specific network architecture. The objective of the neural network is to transform the inputs into meaningful outputs. The architecture of simple Feed-Forward Neural Network (FFNN) is shown in figure 2. It does not contain any self loop or any backward feed.

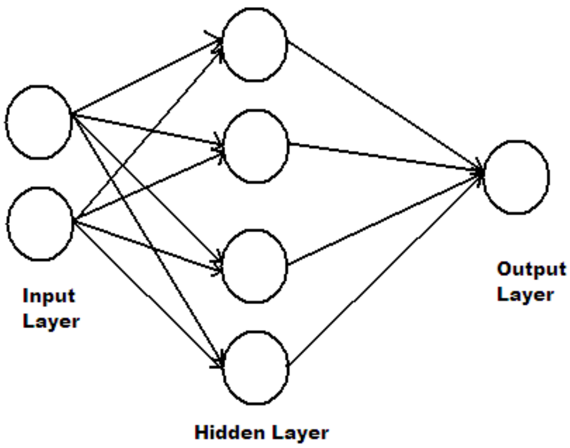

Fig 2: Feed-Forward Neural Network Architecture

3. REGRESSION ANALYSIS

Regression Analysis is very often used as prediction model. It simply finds mathematical relationship between inputs and output. The basic formula for regression model is shown in eq.1.

$$Y_{est} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n \quad (1)$$

Where, Y_{est} is estimated output, $X_1, X_2, \ldots, X_n$ are n independent inputs. This is the basic linear regression model, which is used for linear relationship between inputs and output. While prediction problem is more complex and not linear, we can use logarithmic or non linear regression model. The formula for logarithmic regression model is shown in eq.2.

$$Y_{est} = \beta_0 + \beta_1 \ln X_1 + \beta_2 \ln X_2 + \ldots + \beta_n \ln X_n \quad (2)$$

We have used both of these models for prediction of software effort.

4. EXPERIMENT STUDY

4.1 Data collection

Here for experiment we have used NASA[10] dataset, which contain size in term of KDLOC, methodology and effort value of 18 different NASA projects. The dataset is shown in table1.

Table 1. NASA[10] Dataset

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	100.8	34	138.3
14	2.1	28	5
15	5	29	8.4
16	78.6	35	98.7
17	9.7	27	15.6
18	12.5	27	23.9

4.2 Estimation Model

Here, in the experiment we have used two prediction models: regression model and Feed-Forward Neural Network model.

4.2.1 Regression model

Linear regression is traditional prediction model used for prediction of many fields. We have used linear regression model for comparison to logarithmic regression model. Logarithmic regression is more suitable or effort prediction. The reason is that here input to output relations are not linear.

4.2.2 Feed-Forward Neural Network Model

For this experiment we want to prove that NN model is more preferable for effort prediction, so we used very simple feed-forward NN model with three layers: Input Layer, Hidden Layer and Output Layer. The model we used is shown in figure 3.

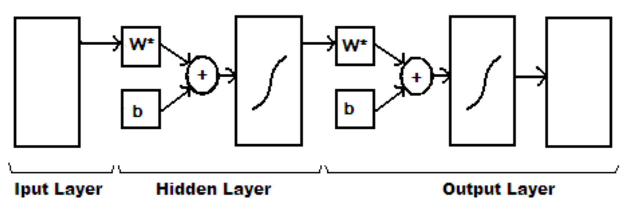

Fig 3: Feed-Forward Neural Network Model

Here, we have used 3 hidden layers model.

4.3 Training and Testing

For training of prediction models we have used first 13 data projects of NASA[10] Dataset. Remaining 5 project data is used for testing of the models.

We have train FFNN using standard Back-propagation training algorithm. As we have very less data for learning, we have used high learning rate. Learning rate we assumed was 0.85. The learning iterations we implemented were 1500. Using this model we found effort for test set.

We found co-efficient of the both regression equations (linear and Logarithmic) using training dataset. Using those equations we found estimated effort value for testing dataset.

4.4 Result of Experiment

The results of estimated values for test dataset for all three models are shown in table 2. The comparison chart between actual effort and estimated values is shown in figure 4.

Table 2. Experimental Result

Actual Effort	Estimated Effort		
	Linear Regression	Logarithmic Regression	Feed-Forward NN
5	6.327	3.906772	5.023160037
8.4	10.3	8.64812	8.344414857
98.7	111.132	107.8629	99.97681567
15.6	16.739	15.86895	14.35436601
23.9	20.575	20.01865	17.75815711

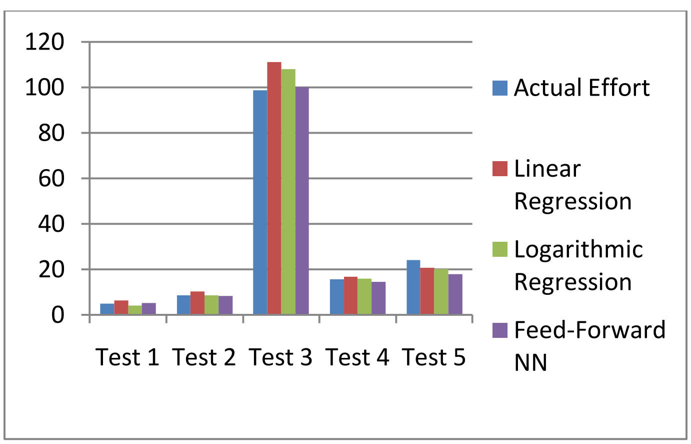

Fig 4: Comparison chart of Actual Effort with Estimated Effort Values

4.5 Evaluation

We have evaluated different prediction models with three evaluation criteria: Mean Magnitude Relative Error (MMRE), Relative Standard Deviation (RSD) and Root Mean Squared Error (RMSE).

4.5.1 Mean Magnitude Relative Error (MMRE)

MMRE is most commonly used evaluation criterion for comparing different prediction models. MMRE is mean (average) value of Magnitude Relative Error (MRE). MRE can be calculated using equation 3.

$$MRE_i = \frac{|E_i - \widehat{E_i}|}{E_i} \tag{3}$$

Where, $\hat{E}$ is Estimated Effort and E is actual effort value.

MMRE is calculated using equation 4.

$$MMRE(\%) = \sum_{i=1}^{n} MRE_i \times 100 \tag{4}$$

Now Tron Foss et. al. [12] shown that, MMRE is an unreliable criterion when used to select between competing prediction models. There is a high probability that MMRE will prefer a model with a bad fit to a model with a good fit to the data. Tron Foss et. al. [12] have suggested RSD is more appropriate for comparison of prediction models rather than MMRE. For this database MMRE shows same priorities of model as per other criteria.

4.5.2 Relative Standard Deviation (RSD)

As we mentioned before RSD is better evaluation criterion than MMRE for prediction Model. Equation 5 shows the formula to calculate RSD value.

$$RSD = \sqrt{\frac{\sum \left(\frac{E_i - \widehat{E_i}}{X_i}\right)^2}{n-1}} \tag{5}$$

Where, X is input value, E is actual effort value and $\hat{E}$ is estimated Effort.

Root Mean Squared Error (RMSE)

This is traditional error measure used for identify error in any observation or prediction. RMSE can be find using formula shown in equation 6.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(E_i - \widehat{E_i})^2} \tag{6}$$

Table 3 shows the calculated values of evaluation criteria for related prediction Models. Figure 5, figure 6 and figure 7 contains chart for comparison of prediction models using different evaluation criteria.

Table 3. Evaluation of Models

Evaluation Criteria	Linear Regression	Logarithmic Regression	Feed-Forward NN
MMRE (%)	16.59364165	10.4131951	7.220296442
RSD	0.404123994	0.309937587	0.254176138
RMSE	5.869897767	4.480030102	2.860339926

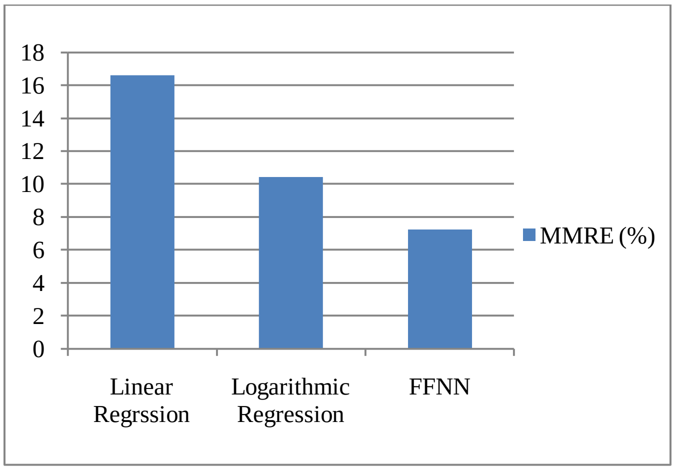

Fig 5: MMRE based comparison of prediction models

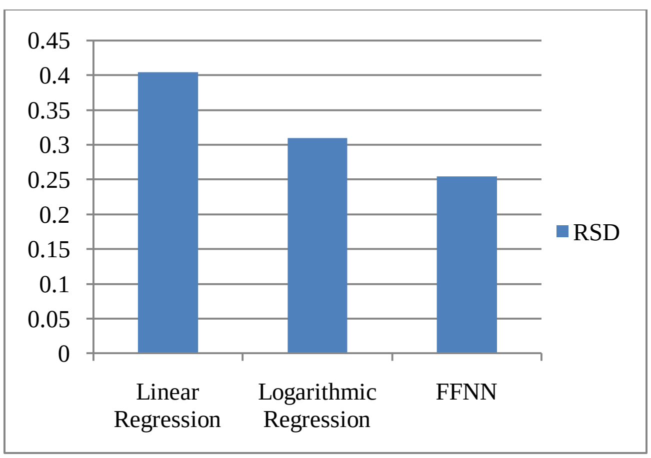

Fig 6: RSD based comparison of prediction models

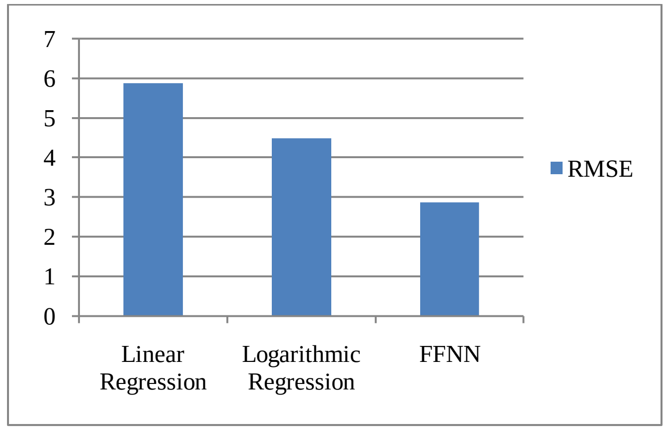

Fig 7: RMSE based comparison of prediction models

5. CONCLUSION AND FUTURE WORK

The similar studies have been done by I. F. de Barcelos Tronto et. al. [13] and R. Bhatnagar et. al.[14]. They have concluded that the Neural Network approach for effort estimation is more preferable over regression model. Here in this study we have got results which prove the same conclusion. Moreover in this paper with the help of experimental study, we have observed that logarithmic regression is more appropriate for estimation of software development effort. Experiment results and evaluation also proved that FFNN predict more accurately compare to other regression models.

For future work we can suggest to study about more independent parameters for estimation of effort, which effect effort value indirect way.

6. REFERENCES

[1] Gray, A. R. and MacDonell S. G. 1997. A comparison of techniques for developing predictive models of software metrics. Information and Software Technology 39.

[2] Srinivasan, K. and Fisher, D. 1995. Machine Learning Approaches to Estimating Software Development Effort. IEEE transactions on software engineering.

[3] Senyard, A., Dart, P. and Sterling L. 2000. Towards the Software Engineering of Neural Networks: A Maturity ModelIn proceeding of Software Engineering Conference, IEEE Xplore.

[4] Moløkken, K. and Jørgensen, M. 2003. A Review of Surveys on Software Effort Estimation. In proceeding of International Symposium on Empirical Software Engineering,IEEE Xplore.

[5] Idri, A., Abran, A. and Khoshgoftaar T. M. 2002. Estimating Software Project Effort by Analogy Based on Linguistic Values. In proceedings of the Eighth IEEE Symposium on Software Metrics (METRICS'02).

[6] Moløkken, K. and Jørgensen, M. 2005. A Comparison of Software Project Overruns—Flexible versus Sequential Development Models. IEEE transactions on software engineering.

[7] Idri, A., Khoshgoftaar T. M. and Abran, A. 2002. Can Neuaral Networks be easily interpreted in Software Cost Estimation? World Congress on Computational Intelligence, Honolulu, Hawaii.

[8] Jørgensen, M. and Shepperd, M. 2007. A Systematic Review of Software Development Cost Estimation Studies. IEEE transactions on software engineering.

[9] Idri, A., Abran, A. and Mbarki S. 2004. Validating and Understanding Software Cost Estimation Models based on Neural Networks. In proceeding of International Conference on Information and Communication Technologies:From Theory to Applications, IEEE Xplore.

[10] Bailey, J. W. and Basili V. R. 1981. A meta model for software development resource expenditure. In proceedings of the International Conference on Software Engineering.

[11] Tadayon, N. 2005. Neural Network Approach for Software Cost Estimation. Proceedings of the International Conference on Information Technology: Coding and Computing, IEEE Computer Society.

[12] Foss, T., Stensrud, E.,Kitchenham, B. and Myrtveit, I. 2003. A Simulation Study of the Model Evaluation Criterion MMRE. IEEE transactions on software engineering.

[13] I. F. de Barcelos Tronto, J. D. Simoes da Silva, N. Sant'Anna. 2007. Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation. In proceeding of IEEEInternational Joint Conference on Neural Networks.

[14] Bhatnagar, R., Bhattacharjee, V. and Ghose M. K. 2010. Software Development Effort Estimation – Neural Network vs. Regression Modeling Approach. International Journal of Engineering Science and Technology.