Comparison of Artificial Neural Network and Regression Models in Software Effort Estimation

I.F. Barcelos Tronto, J.D. Simões da Silva, N. Sant'Anna

[1]Laboratory for Computing and Applied Mathematics - LAC

Brazilian National Institute for Space Research - INPE

C. Postal 515 – 12245-970 – São José dos Campos - SP

BRAZIL

E-mail: {iris_barcelos, demisio, nilson}@lac.inpe.br

Keywords: software effort, artificial neural network, regression analysis, software development estimate

Abstract: Estimating development effort remains a complex problem attracting considerable research attention. Improving the estimation techniques available to project managers would facilitate more effective control of time and budgets in software development. In this paper, predictive Artificial Neural Network and regression based models are investigated, comparing the performance of both methods. The results show that ANNs are effective in effort estimation.

Introduction: A quality level and international productivity on software development can be achieved through the use of effective software management process, focalizing people, product, process, and project. The project requires planning and accompaniment supported by a group of activities, among which the estimates (effort, resources, time, etc.) are fundamental, because they supply a guide for the other activities.

Predicting software development effort with high precision is still a great challenge for project managers. Consequently, there is an ongoing, high level activity in this research field in order to build, to evaluate, and to recommend prediction techniques Mirtveit et al (2005). A large number of different predictive models have been investigated over the last years. They range from mathematical functions: regression analysis (Sentas et al., 2005) and COCOMO (Boehm, 1981) to machine learning models: estimation by analogy (Barcelos Tronto et al., 2006a); clustering techniques (Zong et al., 2004); artificial neural artificial (Barcelos Tronto et al., 2006b). In contrast to a regression model which is defined by a mathematical formula, ML models are not defined by a mathematical formula but may take on many different shapes. Despite the number of research activities, there is still a doubt to advise practitioners as to what prediction models they should select, because the studies have not converged to similar answers. There are a number of factors that should be considered in the selection of a prediction technique, and it is likely that trade-offs will need to be made in the process. Technique selection should be driven by both organizational needs and capability. In terms of need, the most common aim is to maximize the accuracy in prediction; however, other issues may also need to be considered. Different error measurements have been used by various researchers to determine the accuracy, but for this project the main measure for model accuracy is the Mean Magnitude of Relative Error (MMRE). MMRE is the mean of absolute percentage errors:

$$MMRE = \frac{\left(\sum_{i=1}^{n} \left| \frac{M_{est} - M_{act}}{M_{act}} \right| * 100 \right)}{n}$$

where there are *n* projects; M_{act} is the actual effort; and M_{est} is the predicted effort. In this paper, however, the main focus is on investigating the accuracy of the predictions using ANN-based and regression models. A case study was performed to examine the potential of two approaches: a multi-layer perceptron neural network and a linear regression model, using the COCOMO database (Boehm, 1981).

The prediction techniques: In the last years, a great interest on the use of ANNs has grown. ANNs have been successfully applied to several problem domains, in areas such as medicine, engineering, geology, and physics, in general to design solutions for estimate problems, classification, control, etc. They can be used as predictive models because they are modeling techniques capable of modeling complex functions. In this work, the artificial neural networks methodology is used to predicting software development effort (in man-hour) from the project size (given by the amount of source code lines). A comparative analysis was accomplished between a regression model and an ANN model that were calibrated and tested in this study.

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. There exist many different learning algorithms. Feed-forward Multilayer Perceptrons are the most commonly used form of ANN, although many more sophisticated neural networks have been proposed. Multi-layer architectures are mostly trained by the error back propagation algorithm that requires a differentiable activation function. The ANN is initialized with random weights and gradually learns the relationships implicit in a training data set by adjusting its weights when presented to these data. Among the several available training algorithms the error back propagation is the most used by software metrics researchers. In general the studies concerned with the use of ANNs to predict software development effort have focused mostly on the accuracy comparison of algorithmic models rather than on the suitability of the approach for building software effort prediction systems.

Linear regression attempts at finding linear relationship between one or more predictor parameters and a dependent variable, minimizing the mean square of the error across the range of observations in the data set. Some researchers have tried building simple local models, using this type of approach. The philosophy is essentially one of solving local prediction problems before attempting at constructing universal models. A disadvantage with this technique is its vulnerability to extreme outlier values although robust regression techniques, that are less sensitivity to such problems, have been successfully used. Another potential problem is the impact of co-linearity – the tendency of independent variables to be strongly correlated with one another – upon the stability of a regression type prediction system.

The case study: The aim of the case study was to compare two different prediction techniques: ANN and regression models. The analysis undertaken in this study deals with a set of measures taken from COCOMO dataset, a public available data set consisting of a total of 63 projects at the time of this study. It was used for describing and testing one of the most important effort estimative methods: the COCOMO model, implemented by Boehm (1981). Furthermore, various methods have been already applied on it (Sentas et al., 2005; Barcelos Toronto et al., 2006a e 2006b). The effort is represented by the variable EsforcoIT (the amount of man-hour for the software integration and test phase). The systems are mainly written using the programming languages COBOL, PLI, HMI and FORTRAN. The area types are mainly business, scientific and system software. TOTKDSI is the independent variable used to build the ANN and regression model. The learning dataset was constructed by removing projects 6, 12,18,24,30,36,42,48,54 and 60. Since we used all 63 projects of the COCOMO database in order to build our models, the learning dataset contained 53 projects and the test dataset contained 6 projects.

The neural network was implemented with 1 input, 9 units in the first hidden layer, 4 units in the second layer, and 1 output neuron, using the logistic function. The input variable was TOTKDSI and the neural network was trained to estimate effort IT. The training phase was repeated 15 times, in a search for the best network to solve the problem. Besides, different neural network architectures were tried. But, the results presented in this paper correspond to the neural network with the best generalization performance. The linear regression model was calibrated using stepwise backward method. After a number of experiments, we achieved a final regression model. Table 1 summarizes the MMRE and R^2 values resulting from a linear regression of M_{est} and M_{act} values for the stepwise backward regression and the ANN models, and results obtained by Kemerer (with COCOMO-Basic, Function Points and SLIM models.

The predictions obtained from the ANN and the regression model (after training on the COCOMO data) using the test dataset indicate that stepwise regression's and ANN's predictions show a strong linear relationship with the actual development effort values for the ten test projects. On this dimension, the performance of the ANN model is less then SLIM's performance in Kemerer's experiments, but better than the stepwise regression models. In terms of MMRE, the ANN performs strikingly well compared to the other approaches, and regression model.

Table 1. The predictive accuracy

	Regress. Eq.	R- square	MMRE
ANN	-1,68+1,676*x	0,85	420
Regression	-1,71+1,623*x	0,83	462
FPA	-37 +0,96x	0,58	103
COCOMO	27,7 + 0,156x	0,70	610
SLIM	49,9 +0,082x	0,89	772

Conclusion and future works: This experiment illustrates two points. In an absolute sense, none of the models perform particularly well at estimating software development effort, particularly along the MMRE dimension, but in a relative sense ANN approach is competitive with traditional models. In general, even though MMRE is high in the case of all models, a high R^2 suggests that by "calibrating" a model's prediction in a new environment, the adjusted model prediction can be reliably used. Along the R^2 dimension, the ANN method provides significant fits to the data. Consequently, new experiments are being led in order to combine the ANN and regression techniques to training and testing a software effort prediction, on other datasets.

REFERENCES

Barcelos Tronto, I.F., Simões da Silva, J.D., Sant'Anna, N., Petit, J.M. (2006a), *Melhorando as Estimativas de Esforço de Software Através de um Modelo Baseado em Analogia*, in *XXVI Congresso da Sociedade Brasileira de Computação, XXXIII Seminário Integrado de Software e Hardware – SEMISH*, Campo Grande, July 14 – July 20, 2006, pp. 217-230.

Barcelos Tronto, I.F., Simões da Silva, J.D., Sant'Anna, N. (2006b), *Uma Investigação de Modelos de Estimativas de Esforço em Gerenciamento de Projeto de Software*, in *Simpósio Brasileiro de Engenharia de Software* – SBES, Campo Grande, October 16 – October 20, 2006.

Boehn, B.W. (1981), *Software engineering economics*, Prentice-Hall, Englewood Cliffs, NJ.

Kemerer, C.F. (1987), *An empirical validation of software cost estimation models,* Comunication of ACM, .30, 416-429.

Myrtveit, I., Stensrud, E. and Shepperd, M.(2005), *Reliability and Validity in Comparative Studies of Software Prediction Models*, IEEE Transaction on Soft. Engineering,, 31(5), 46-62.

Sentas, P.,Angelis, L., Stamelos, I. and Bleris, G.(2005), *Software productivity and effort prediction with ordinal regression*, Journal Information and Software Technology, 47, 17-29.

Zhong, S., Khoshgoftaar, T.M., and Seliya, N.(2004), *Analysing Software Measurement Data with Clustering Techniques*, IEEE Intelligent Systems, 20-27.