Comparative Analysis of Neural Network Techniques for Estimation

Amrinder Singh Grewal
M.E, CSE
University institute of Engineering and Technology, Panjab University, Chandigarh

Vishal Gupta
Assistant Professor, CSE
University Institute of Engineering and Technology, Panjab University, Chandigarh

Rohit Kumar
Assistant Professor, CSE
University Institute of Engineering and Technology, Panjab University, Chandigarh

ABSTRACT

Software cost estimation is the process of predicting the effort required to develop a software system. Accurate cost estimation helps us complete the project within time and budget. For completing the project in time and budget, one must have efficient estimation technique for predicting project efforts. Artificial neural network is a promising technique to provide efficient and good results when dealing with problems where there are complex relationship between inputs and outputs. Researchers proved better estimation using back propagation techniques like RBP and Bayesian regulation. In this paper further discussion will be about the study and the efficiency of Neural based one step secant back propagation based cost estimation model, Powell-Beale conjugate gradient model and Fletcher-reeves conjugate gradient model. Result is concluded with the best effort predicting model.

Keywords: software estimation; artificial neural networks; one step secant BP; Powell-beale conjugate gradient; Fletcher-powell conjugate gradient.

1. INTRODUCTION

Neural Network consists of large number of highly interconnected elements called nodes, where each node produces a non-linear function of its input [11].Each node is connected to the number of nodes in the other layer. The main inspiration for the field of Neural Networks (NN) originated from the desire to produce artificial systems capable of sophisticated, perhaps "intelligent", computations similar to the biological neurons in brain structures. Although there are a number of learning algorithms to train a neural network, the back propagation (back-prop) paradigm has emerged to be the most popular learning mechanism for prediction and classification problems. When the relationship between the input and output variables is nonlinear, the hidden layer helps in extracting higher level features and facilitate generalization. Connections between neurons have numerical weights associated with them; the weights are adjusted in the training process by repeatedly feeding examples from the training set.

2. BACKGROUND

In the last few years' research, there are many software cost estimation methods available including algorithmic methods, estimating by analogy, expert judgment method, price to win method, top-down method, and bottom-up method. No one method is necessarily better or worse than the other, in fact, their strengths and weaknesses are often complimentary to each other.[1] Numerous researchers and scientists are constantly working on developing new software cost estimation techniques [2, 3, 4].

These techniques may be grouped into two major Categories: (1) parametric models, which are derived from the statistical or numerical analysis of Historical projects data [5], and (2) non-parametric models, which are based on a set of artificial Intelligence techniques such as artificial neural networks, analogy-based reasoning, regression trees, Genetic algorithms and rule-based induction [6][7][8]. More recently, attention has turned to a variety of machine learning (ML) [9] methods to predict software development effort. Artificial neural nets (ANNs)[10][11],genetic algorithms[12], case based reasoning (CBR)[13] and rule induction (RI)[15],estimating by analogy[14],clustering techniques are examples of such methods. Many researchers have applied the neural networks approach to estimate software development effort [16][17][18].

3. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Network is a promising techniques to build predictive models, because they are capable of modeling non linear relationships ANNs posses' large number of highly interconnected processing elements called neurons, which usually operate in parallel and are configured in regular architectures. Each neuron connected with the other by a communication link and each connection link is associated with weights which contain information about the input signal. The neuron computes a weighted sum of its inputs and generates an output if the sum exceeds a certain threshold.

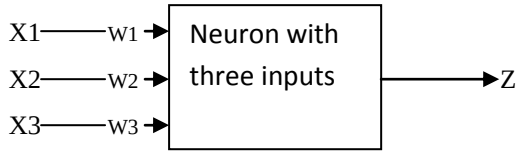

Figure 1 Neuron Model with 3 Inputs

Each neuron has an activation level, specified by continuous or discrete values. The internal activation is modified by a transfer function and becomes an output, which in turn may become an input to one or more neurons. For classification problems, a sigmoid transfer function is typically used to transform the input signals into output signals. The sigmoid transfer function is represented by F (I) = 1/ (l+e-l), where I represents the internal activation. The process continues until one or more outputs are generated. These are estimation models that can be "trained" using historical data to produce ever better results by automatically adjusting their algorithmic parameter values to reduce the delta between known actual and model predictions.

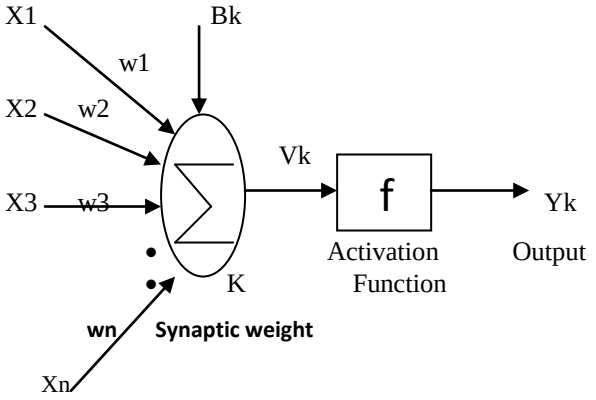

Figure 2: An Artificial Neuron with Activation function

The back-propagation learning algorithm is one of the most important developments in neural networks (Bryson and Ho, 1969; Werbos, 1974; Lecun, 1985; Parker, 1985; Rumelhart, 1986 ;) In back-prop network paradigm, all the connection weights are assumed to be responsible for the output error. Error is defined to be the difference between a network's estimated output or predicted value and the corresponding observed output value. The error values are calculated at the output layer and propagated to previous layers and used for adjusting the connection weights. The training process consists of repeatedly feeding input and output data from empirical observations, propagating the error values, and adjusting the connection weights until the error values fall below a user-specified tolerance level.

4. NEURAL NETWORK TECHNIQUES USED

In this paper, Three Neural Network based cost estimation models for predicting best estimates using NASA dataset are used. The comparison between these three models for finding best among them for predicting cost estimations. Comparison of Neural networks with traditional Cocomo, Halstead, Walston-Felix, Bailey-Basili and Doty models has been done by several authors proving that Neural Network provides much better results than these models.

Here we compared Neural Network techniques with one another. These techniques are Neural based One step secant back propagation based cost estimation model, Powell-Beale conjugate gradient model and Fletcher-Reeves conjugate gradient model.

These models can train any network as long as its weight, net input, and transfer functions have derivative functions. Back propagation is used to calculate derivatives of performance perf with respect to the weight and bias variables X. Each variable is adjusted according to the following:

$$X = X + a*dX;$$

Where dX is the search direction. The parameter a is selected to minimize the performance along the search direction. The line search function searchFcn is used to locate the minimum point. The first search direction is the negative of the gradient of performance. In succeeding iterations the search direction is computed from the new gradient and the previous steps and gradients, according to the formula:

For One step secant back propagation:

$$dx = -gX + Ac*X\_step + Bc*dgX;$$

Where gX is the gradient, X_step is the change in the weights on the previous iteration, and dgX is the change in the gradient from the last iteration.

For Conjugate gradient back propagation with Powell-Beale restarts:

$$dX = -gX + dX\_old*Z;$$

Where dX is the gradient. The parameter Z can be computed in several different ways. The Powell-Beale variation of conjugate gradient is distinguished by two features. First, the algorithm uses a test to determine when to reset the search direction to the negative of the gradient. Second, the search direction is computed from the negative gradient, the previous search direction, and the last search direction before the previous reset.

For Conjugate gradient back propagation with Fletcher-Reeves updates:

$$dX = -gX + dX\_old*Z:$$

For the Fletcher-Reeves variation of conjugate gradient it is computed according to

Z = normnew_sqr/norm_sqr;

Where norm_sqr is the norm square of the previous gradient and normnew_sqr is the norm square is the norm square of the current gradient.

Training stops when any of these conditions occurs:

- The maximum number of epochs is reached.
- The maximum amount of time is exceeded.
- Performance is minimized to goal.
- The performance gradient falls below min_grad.

5. PERFORMANCE CRITERIA

In this paper MATLAB R2012a is used with Neural Network Toolbox for the development of neural based models. Perform the comparison of the models on basis of Mean Magnitude of Relative Error (MMRE).

We considered the MMRE as the main performance measure. 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.

Mean magnitude of relative error:

(MMRE)= (1/T)*(MRE1+MRE2+…..MRET)

Where T is total number of projects.MRE is the magnitude of the relative error.

The objective of the study is deducing conclusions at the end that which one is better on the basis of MMRE.

6. RESULTS AND CONCLUSIONS

Table1. Data of Actual cost comparison with others

Project	Efforts of OSSBP	Efforts of CGBPBR	Efforts of CGBFRU	Actual Efforts
1	**124.22**	**122.18**	**125.45**	**117.6**
2	**112.40**	**113.81**	**116.28**	**117.6**
3	**31.98**	**25.57**	**25.92**	**31.2**
4	**32.89**	**27.55**	**27.73**	**36**
5	**35.98**	**33.73**	**33.48**	**25.2**
6	**24.79**	**6.28**	**9.05**	**8.4**
7	**26.09**	**10.44**	**12.55**	**10.8**
8	**358.60**	**351.49**	**338.74**	**352.8**
9	**81.68**	**71.04**	**65.91**	**72**
10	**35.29**	**128.75**	**158.73**	**72**
11	**19.82**	**29.61**	**42.70**	**24**
12	**354.41**	**359.64**	**358.90**	**360**
13	**19.26**	**27.99**	**34.27**	**36**
14	**214.39**	**215.30**	**199.10**	**215**
15	**20.63**	**44.37**	**58.49**	**48**

Table2. Comparison On the basis of MMRE

Model Used	OSSBP	CGBPBR	CGBFRU	Actual
MMRE	**0.393**	**0.163**	**0.233**	**0**

The performances of the Neural Network based effort estimation system are compared for effort dataset available with NASA. The results show that Neural Network based conjugate gradient back propagation Powell-Beale restarts has the lowest MMRE to Actual i.e. 0.163 and second best performance is shown by conjugate gradient back propagation Fletcher-Reeves updates i.e. 0.233. Hence, the proposed Neuro based system is able to provide good estimation capabilities. By using Neural Network techniques most accurate estimates can be made in future which are used in formulating complex relationship between the variables. It is suggested to use Neuro based techniques for estimating the all types of projects.

7. ACKNOWLEDGEMENT

Many thanks to Mr. Vishal Gupta Assistant Professor and Rohit kumar Assistant Professor in UIET, Pan jab University Chandigarh, for their efforts.

8. REFERENCES

[1] Murali Chemuturi "Analogy based Software Estimation" Chemuturi Consultants

[2] J .P. Lewis, "Large Limits to Software Estimation," Software Engineering Notes, Vol. 26, No. 4, July 2001

[3] Stamelos, etal, "Estimating the development cost of Custom software", Information and Management, 2003

[4] R. W. Jensen, "Extreme Software Cost Estimating", Crosstalk, Journal of defense software Eng, Jan 2004

[5] 0 W Boehm, Software Engineering Economics Prentice-Hall, 1981

[6] A.Idri, A.Abran and T.M. Khoshgoftaar, "Estimating Software Project Effort by Analogy

based on Linguistic values," 8th IEEE International Software Metrics Symposium, Ottawa, Canada

[7] K. Srinivasan and D. Fisher "Machine Learning Approaches to Estimating Software Development Effort," IEEE Transactions on Software Engineering, vol. 21, no. 2, February, 1995

[8] S. Vicinanza, and M.J. Prictolla, "Case-Based Reasoning in Software Effort Estimation," Proceedings of the 11th Int. Conf. on Information Systems, 1990

[9] Jianfeng, Wen, Shixian, Li, Changqin Huang," Systematic literature review of machine learning based software development effort estimation models," Information and Software Technology (2012)

[10] Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi," Neural Network-A Novel Technique for Software Effort Estimation," International Journal of Computer Theory and Engineering 2010

[11] Manpreet Kaur, Sushil Garg," Analysis of Neural Network based Approaches for Software effort Estimation and Comparison with Intermediate COCOMO," International Journal of Engineering and Innovative Technology June 2012

[12] Collin j. burgess and martin lefly," Can genetic programming improve software estimation. A review," Information and software tech. 2001

[13] Aarmodt and Plaza (1994)," Case-Based Reasoning: Foundational issues, Methodical Variations and System Approaches." AI Communications

[14] Stephen G. MacDonell, Martin J. Shepherd," Combining techniques to optimize effort predictions in software project management," The Journal of Systems and Software (2003)

[15] Carolyn Mair, Gada Kadoda, Martin Lefley," An Investigation of Machine Learning Based Prediction Systems."

[16] Ali idri, taghi, M. khoshgoftaar, Alain abran,"can neural network be easily interpreted in software cost estimation?

[17] Ch.Satyananda Reddy and KVSVN Raju," An Optimal Neural Network Model for Software Effort Estimation," DENSE Research Group

[18] Jagannath Singh and Bibhudatta Sahoo," Software Effort Estimation with Different Artificial Neural network," 2nd National Conference- Computing, Communication and Sensor Network,"2011