



# International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

[www.iasir.net](http://www.iasir.net)

## Cost Estimation with Neural Network and Affect of Different Input parameters on Cost Estimation

Amrinder Singh Grewal<sup>1</sup>, Vishal Gupta<sup>2</sup>

<sup>1</sup>M.E CSE, 2Assistant Professor, UIET,  
Panjab University, Chandigarh, INDIA

**ABSTRACT:** Software estimation is a process of estimating the effort, size and cost of the software, before the actual development of the software. Realistic estimation helps the software developers and managers in finding the required resources, man-hours, time and cost required in developing software. There are various types of estimation methods such as algorithmic, regression, machine learning, function pts etc, but still there is need of an accurate and precise software estimation method. Some models don't include all the parameters, some are extremely vulnerable to misclassification or error deviation is high. In this paper neural network training functions are used for software estimation as neural network is good at discovering relations (linear or non linear). Results of neural network are compared with cocomo 2 efforts as it is the most used model in the world for estimation. Input to neural network training functions is Cocomo 2 parameters which consist of 17 cost multipliers and 5 scale factors. After analyzing the results, a hybrid set of parameters are given to neural network training functions. Results from these hybrid parameters shows that neural networks techniques shows lowest mean error as compared to cocomo 2 parameters.

**Keywords:** software estimation; artificial neural networks; cocomo2; hybrid parameters.

### I. INTRODUCTION

Software estimation is a science of determining the how many hours and how many workers are required to complete the software product. Cost of the project largely depends upon how many resources and workers are required. Accurate estimation leads to successful completion of software project because it is used to determine what resources are better matched to real needs. In recent years, software becomes the most expensive part of our computer system. So when project managers or developers estimate the project resources and time inaccurately, it leads to the great loss for the organization. Therefore it is desirable to estimate accurately for proper budgeting and successful completion of project in time. Industries los opportunities when budget overruns or project takes too much time for completion. Neural networks are those information systems, which are constructed and implemented to model the human brain. The main objective of the neural network research is to develop a computational device for modeling the brain to perform various computational tasks at a faster rate than the traditional systems. Artificial neural networks perform various tasks such as pattern matching and classification, optimization function, approximation, vector quantization and data clustering etc. These tasks are very difficult for traditional computers, which are faster in algorithmic computational tasks and arithmetic operations. Back propagation technique is widely used for estimating various projects.

In this paper neural network training functions are used for software estimation. Efforts are calculated with different neural network functions and cocomo2, predicting the most successful estimation model. Comparison is done on the basis of MMRE (mean magnitude of relative error), which calculates the error deviation. For best results MMRE must be least and our goal is to find the model with lowest MMRE. This comparison is done with cocomo2 dataset of 50 projects, which consists of 17 cost drivers and 5 scale factors. After that a hybrid dataset of 50 projects is used which consists of 21 input parameters. Predicting the efforts using hybrid parameters for same neural network training functions and comparison is done on the basis of MMRE.

### II. BACKGROUND

Accurate estimate means better planning and efficient use of project resources such as cost, duration and effort requirements for software projects especially space and military projects. Efficient software project estimation is one of the most demanding tasks in software development. A large software industry growth is taking place in recent years and failure of different software projects is the matter of worry for researchers. According to Molokken and jorgenson nearly 30% to 40% of the projects are accomplished and rests are failing. There is lot of work have been conducted by several authors[1] [2] [3] in the field of model based estimation techniques, expertise based, learning oriented, dynamic based[4], regression based[5][6] and composite Bayesian such as cocomo 2. Common model based techniques are SLIM, cocomo, checkpoint and SEER. Delphi and rule based are come under expertise based estimation techniques. From few years researchers concentrate on various machine learning (ML) [7] methods to predict software development effort. Artificial neural (ANNs) [8] [9], genetic algorithms [10] [11], case based reasoning (CBR) [12] [13] and rule induction (RI), estimating by

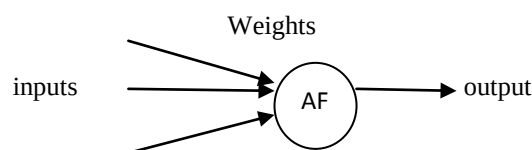
analogy, clustering techniques [14] are examples of such methods. Several researchers have applied the neural networks approach to estimate software development effort [15] [16] [17]. Wittig and Finnie [18] describe their use of back propagation learning algorithms on a multilayer perceptron in order to predict development effort and cost. They consider ANNs as promising techniques to build predictive models, because they are capable of modeling non linear relationships. No one method is necessarily better or worse than the other, in fact, their strengths and weaknesses are often complimentary to each other.

### III. ARTIFICIAL NEURAL NETWORKS

An artificial neural network simulates the behavior of brain which consists of biological neurons. It has large count of neurons which are highly interconnected in the same way as biological neurons. This adaptive system keeps improving its structure based on the input data during the training process. Neural network architecture is recognized for its remarkable property i.e. "Learn by example."

Artificial neural network has high number of largely interconnected processing elements called neurons. A connection link is used to connect each neuron with other neuron and has weights (contain information about the input signal) associated with it. Information contained in the weights is used to solve a particular problem problems are solved by using information held in the weights. Parallel operation of neurons is characterized by their ability to recall, learn and generalize training patterns. Each processing element of network has its activation function which is derived from the input received by neuron.

Artificial neural network usually operate in parallel. The parallel nature of neural network enables it to achieve remarkable computation speeds not attainable by conventional sequential systems. The main objective of neural network research is to develop a computational device for modeling the brain to perform various tasks at a faster rate than the conventional systems. Neural network is an energetic tool that represents complex input/output relationships very efficiently. Particular target output is achieved from a specific input by training the neural networks. Neural networks have enlarged applications in many fields such as medical and engineering. As artificial neural networks are wonderful to identify patterns, it is used in recognizing diseases.



Where, AF is activation function.

**Figure 1 a simple neural network**

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.

### IV. NEURAL NETWORK TECHNIQUES USED

In this paper, we have used two Neural Network based cost estimation models for predicting best estimates using Cocomo2 dataset. We compared these two models for finding best among them for predicting cost estimations and compare with the Cocomo2. These techniques are Neural based Conjugate gradient back propagation with polak-ribiere updates (CGBPRU) and Bayesian regulation back propagation (BRBP).

CGBPRU (Conjugate gradient back propagation with polak-ribiere updates) is used to train network until transfer function, inputs and weights have derived functions. Derivatives of performance are computed by back propagation and every variable is set as:

$$X = x + a \times dx$$

Performance towards the search direction is minimized by adjusting the parameter a. minimum point is located by using line search function. Search direction can be computed by using old search direction and new gradient as:

$$Dx = -gx + dx\_old \times z$$

Where, gx is the gradient. Z is conjugat direction and can be computed as:

$$Z = ((gX - gX\_old)' \times gX) / \text{norm\_sqr};$$

Gx\_old is gradient of previous iteration.

BRBP (Bayesian regulation back propagation) is used to train network until its inputs, weights and bias have derived functions. it restricts the linear relationship between errors and weights. It is also responsible for competent generalization property. It minimizes a combination of squared errors and weights and then determines the correct combination so as to produce a network that generalizes well.

BRBP works as Levenberg-Marquardt algorithm works. It has to compute jacobian of performance by using back propagation and every variable is set as it is adjusted in Levenberg-Marquardt algorithm. Same equations are used as LMBP. Speed of the algorithm largely depends upon the memory function mem\_reduc. If we set mem\_reduc 1, then it speeds up using large memory space and if we set mem\_reduc2 then speeds down and have less memory requirements.

Network training obstruct when min\_grad reached, time exceeds, mu increased from max, repetitions exceeds or goal reached.

## V. PERFORMANCE CRITERIA

Calculate the efforts using COCOMO2 effort equation:

$$\text{Effort in PM} = 2.94 (\text{SIZE})^E * \text{EM}$$

Where EM= effort multipliers which is the product of 17 cost driver attribute.

$$E = B + 0.01 * \sum_{J=1}^5 SF$$

Where SF is scale factors, B is constant having value 0.91.

Cocomo2 parameters:

The values of the constants and scale factors are similar for three stages of Cocomo2 model. Size is estimated in KLOC. Effort multiplier is the products of 17 cost drivers that include subjective assessment of various technological factors are:

- Reliability of software
- Size of the database
- Reusability
- Documentation
- Time constraint during execution
- Storage requirement
- Volatility of platform
- Capability of analyst
- Capability of programmer
- Continuity in personal
- Applications Experience
- Experience of platform
- Experience of tools and language
- Software tools
- Multisite Development
- Complexity of software
- Development Schedule

Cocomo2 has 5 scale factors. Scale factors are used to describe the interrelated economies or diseconomies for the projects which are different in size and other measures. These parameters have their affect according to the size of the project i.e. more impact on large projects and less impact on small projects. 5 scale factors are given below:

- Precedentedness
- Flexibility during development
- Risk resolution
- Cohesion
- Maturity of process

In this paper we have used MATLAB R2012a 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).

**Mean magnitude of relative error:**

$$(\text{MMRE}) = (1/T) * (\text{MRE1} + \text{MRE2} + \dots + \text{MRE}_T)$$

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

Main objective is to find the best effort estimating model on the basis of MMRE (mean magnitude of relative error)

## VI. HYBRID PARAMETERS

This study is done to build an aid to the developed algorithms. So far we have done effort estimation using cocomo2 dataset. This dataset contain 17 cost drivers and 5 scale factors. Parameter used in this dataset contains attributes of product, personal and computer etc. In spite of number of various types of effective parameters

used in cocomo2 dataset, we can say that some important factors are there which can be very useful in estimating efforts such as technological factors, development phase and experience factors etc. Here we used 34 different factors from various categories and created an artificial dataset of 40 projects. In this study some parameters of cocomo2 are also included as they are important for estimating effort along with the parameters included from various research studies.

- Documentation
- Availability of resources(human, financial, raw, facilities)
- Productivity
- Analyst experience
- Programmer experience
- Required software quality
- Emergent technology
- Required software reliability
- Database size
- Complexity of software
- Execution time constraint
- Main storage constraint
- Analyst capability
- Distributed data processing
- Programmers capability
- Use of software tools
- Developed for reusability
- Platform volatility
- Size of development team
- Business attributes
- Computer hardware
- Personal continuity
- Risk resolution
- Cohesion
- Cost of networking and communication
- Development phase
- Structure of organization
- Adequate funding to completion
- Planning and control techniques
- Customer consultation and involvement
- Implementation problems
- Design of software(complex or simple)
- Technical training
- Environment

Each of the attributes receives a rating on a six point scale that ranges from “very low” to “extra high” (in importance or value). Efforts are calculated using neural network training algorithms. Error is the difference between the actual efforts and estimated efforts. Lower the MMRE (mean magnitude of relative error), higher is the performance of the training models.

## VII. RESULTS AND DISCUSSION

Estimated efforts are calculated using neural based models and cocomo2 equations.

**Error = Actual efforts – Estimated efforts**

**RE = AE – EE**

Where RE is Relative error, AE is Actual efforts and EE is estimated efforts.

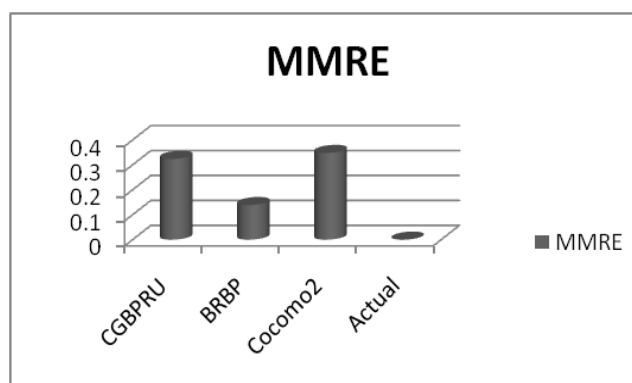
**MRE = Magnitude of relative error**

**MMRE =  $1/T \times \text{sum of MREs}$**

Where T is total number of projects

Table 1 comparison on the basis of MMRE

| Model Used | Cocomo2 | CGBPRU | BRBP  | Actual |
|------------|---------|--------|-------|--------|
| MMRE       | 0.348   | 0.324  | 0.139 | 0      |



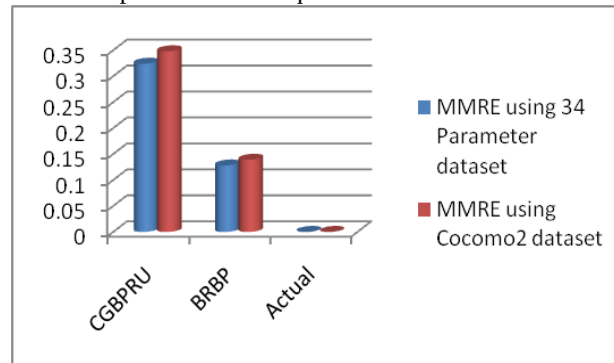
### A. Results using hybrid parameters

Now calculate efforts and errors by using neural network algorithms over a hybrid dataset of 34 different parameters and compared with the above developed MMRE values.

**Table 2 comparison of estimated MMRE using different datasets**

| Model used                  | CGBPRU | BRBP  | Actual |
|-----------------------------|--------|-------|--------|
| Using 35 parameters dataset | 0.287  | 0.128 | 0      |
| Using cocomo2 dataset       | 0.324  | 0.139 | 0      |

The neural network model shows the better results as compared to cocomo2 that is being extensively used for the software effort estimation. As we have seen that neural network models having least MMRE (mean magnitude of relative error) than COCOMO2. So, neural network training functions are able to provide good estimations about the software development efforts required.



Therefore it is suggested to build a model structure for software effort by using neural network techniques which has more accuracy than the other models. It is also suggested to use all the important parameters of the software project that are critical for software estimation. Results show that if we consider more aspects of the project that has affect on software development, our estimate becomes more accurate. Use of various important parameters helps in considering all the aspects of the software projects. Results shows neural network is able to provide better effort estimation than other models, so neural network is promising technique in software estimation field.

## VIII. ACKNOWLEDGMENTS

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

## REFERENCES

- [1] A.Idri.,A.Abran and T.M.Khoshgoftaar, "Estimating Software Project Effort by Analogy based on Linguistic values," 8th IEEE International Software Metrics Symposium," 4-7 April, Ottawa, Canada, 2002, pp. 21 -30
- [2] Ali idri, taghi, M. khoshgoftaar, Alain abran,"can neural network be easily interpreted in software cost estimation?
- [3] Murali Chemuturi "Analogy based Software Estimation" Chemuturi Consultants
- [4] Madachy, "A Software Project Dynamics Model for Process Cost, Schedule and Risk Assessment," Ph.D. Dissertation, University of California
- [5] Roheet Bhatnagar, Vandana Bhattacharje,"Software Development Effort Estimation – Neural Network vs. Regression Modeling Approach," International Journal of Engineering Science and Technology 2010
- [6] Ali Bou Nassifa, Danny Hob," Towards an early software estimation using log-linear regression and a multilayer perceptron model," The Journal of Systems and Software (2012)
- [7] Jianfeng Wen,Shixian Li,Changqin Huang," Systematic literature review of machine learning based software development effort estimation models," Information and Software Technology (2012)
- [8] Jaswinder Kaur, Satwinder Singh, Dr. KaranjeetSingh Kahlon, Pourush Bassi," Neural Network-ANovel Technique for Software Effort Estimation," International Journal of Computer Theory and Engineering 2010
- [9] Heejun Park, Seung Baek," An empirical validation of a neural network model for software effort estimation," Expert Systems with Applications(2008)
- [10] Kavita Choudhary," GA Based Optimization of Software Development Effort Estimation," IJCST, September 2010
- [11] Collin j. burgess and martin lefly," can genetic programmin improve software estimation. A review," information and software tech. 2001
- [12] Aarmodt and Plaza (1994)," Case-Based Reasoning: Foundational issues, Methodical Variations and System Approaches. AI Communications
- [13] Ch.Satyananda Reddy and KVSVN Raju," An Optimal Neural Network Model for Software Effort Estimation," DENSE Research Group
- [14] Anita Lee, Chun Hung Cheng, Jaydeep," Software development cost estimation: Integrating neural network with cluster analysis." Information & Management 1998
- [15] Ali idri, taghi, M. khoshgoftaar, Alain abran,"can neural network be easily interpreted in software cost estimation?
- [16] Jagannath Singh and Bibhudatta Sahoo," Software Effort Estimation with Different Artificial Neural network." "2nd National Conference- Computing, Communication and Sensor Network,"2011
- [17] Jitendra, Vikas, Kuldeep, Samiksha," Cost prediction using Neural Network Learning Techniques," IJCSMS International Journal of Computer Science and Management Studies, 2011
- [18] G. Witting and G. Finnie, "Using Artificial Neural Networks and Function Points to Estimate 4GL Software Development effort", J. Information Systems.