Neural Network: A better Approach for Software Effort Estimation

Salma Tarannum
M.tech CSE (2nd yr)
Integral University
Lucknow, India

Md. Suaib
M.tech CSE (Assistant Professor)
Integral University
Lucknow, India

Abdul Muttalib
M.tech CSE
Integral University
Lucknow, India

ABSTRACT
Software effort estimation is to find out the development effort required by a project. Before any project is being started firstly it is required to estimate the development effort required. This is one of the most important and challenging activities that has done before. Various techniques have been proposed to calculate effort estimation, but providing accurate cost is still an unachievable goal. This paper represents different neural network techniques that have been proposed to calculate effort estimation. It describes various techniques and approach proposed by different researchers representing how neural network gives accurate effort estimation in comparison with other techniques. This paper also focuses on how researchers have used various concepts of neural network and brought it into use to estimate effort.

General Terms
Neural Network, Effort Estimation, Clustering Algorithm, Pattern Classification.

Keywords
Effort Estimation, Constructive Cost model (COCOMO), Neural Network, Radial Basis Function, Functional Link Neural Network (FLAN), Artificial Neural Network (ANN), Mean Square Relative Error (MMRE), Mean Square Error (MRE).

1. INTRODUCTION
For developing any kind of software project number of steps are executed before the beginning of actual project development. But one of the most important steps in complete development of any project is software effort estimation. This has mark a challenging task for any software developing industry. There is no particular formula to calculate development effort. Various researchers have represented different approach for estimation. Some of them were able to provide a good approach, but still for large and complex projects it's a challenging task. For any organization to complete a project successfully it is required to complete project within limited time and budget. This can only be done once the development cost has been predicted. The estimation of required effort before starting a project helps the developer to calculate the amount of budget and time that will be required and this approach leads to a better approach. Therefore various techniques have been developed just to find out the software effort estimation. Among various approach it has been examined that neural network techniques provides a better and accurate performance as compared to other methods. Neural network basically uses the concept of artificial intelligence, machine learning, and classification methods etc. Which is a very effective technique in various fields, therefore it is feasible to use neural network techniques for large and complex softwares. Kathleen Peters [1] has provided a block diagram representing different criteria and features that should be considered during estimation of software project and can also be called as software estimation tool.

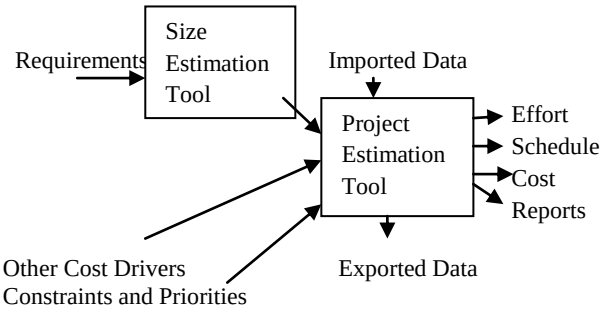

Fig 1: Estimation Tool Context

1.1 Software Effort Estimation
Software effort estimation is also called as cost estimation. One of the major costs in software is labour cost (man-power), so in the case of software it is called as effort estimation. One of the basic costs of traditional product is man-power and a successful project is the project that is completed within limited time and limited cost. There are different issues in software effort estimation

i. Algorithmic model: in case of this model problem arises in choosing the best algorithm. E.g. COCOMO Model.

ii. Size Metrics: this concept uses LOC (line of code) and FP (function point). LOC differs from technology to technology and every person will estimate in different manner. FP technique was given by Alen Albercht (IBM) in 1970. It was a solution for size measure, which shows that every functionality should have different modules: an input, outputs, enquires, internal logic files, external interface files. Each module should give weight to each and every other module.

iii. Non- Algorithmic model: under this model we have a technique called expert judgement. It does not require minute details. All the inputs are given by experts and based on expertise result modification is done.

2. NEURAL NETWORKS
Neural network has derived from the biological term (Neural). It aims to include all the features and characteristics of human neuron system. At very first it was termed as artificial neural network and was used especially for pattern classification and recognition. But, with the technological advancement neural network has a wider range of applications and intelligence. In a tutorial [2] mentioned by researchers the characteristics of

ANN that was not present in Von Neumann or a Morden parallel computer. These include: massive parallelism, distributed representation and computation, learning ability, generalization ability, adaptivity, inherent contextual information processing, fault tolerance, and low energy consumption. Some of the most efficient techniques that has been used in software fields are: categorization (clustering), function approximation, forecasting (prediction) etc. ANN is a kind of weighted directed graph where nodes are the neurons and edges depicts the connection between input neurons with a weight value attached to it and output neurons. Based on the connection links the complete architecture can be grouped into two categories:

i. Feed forward network: in which the edges of the graph moves forward and graph does not include any loop.

ii. Feed backward network (or recurrent network): in this network the edges will have backward paths connecting and hence the graph may include loops.

3. NEURAL NETWORK TECHNIQUES

Table 1. Various techniques used in neural network

Name Neural Network Techniques	Description
Learning [2]	Learning method helps ANN to update (adjusts) weights and architecture according to the parameters. Learning characteristics helps the ANN to learn automatically from examples.
RBFN (Radial basis function) [3]	RBFN is a feed forward network which consist of three layers an input layer, hidden layer and output layer. Output value is calculated based on RBF function. Each neurons in the hidden layers employees a radial basis function (Gaussian function).
ANN (feed forward network)	This is the simple structure of neural network, in which the weighted sum of n inputs produced a desired output. The output result depends on the threshold value.
FLAN (Functional link artificial neural network)	This method consists of one input layer and one output layer. This is called as FLAN as it depends on functional approximation. Learning approach helps in expanding input layer (cost drivers), therefore this network works faster in case of complex projects.
Machine learning techniques	It comes under Artificial neural network (subfield) of ANN. This technique is also based on learning approach. SVM (support vector machine) also comes under machine learning.
ANFIS (adaptive neuro fuzzy inference system)	ANFIS is the combination of artificial neural network and fuzzy logic. This is one of the good approach in development estimation
Identity Spoofing	The attacker pretends to be the originator of the message and tries to gain access to legitimate resources.

4. RELATED WORK

The paper by Srichandan,[3] have discussed that, with advancement in time number of software effort estimation models has been developed, but still there is no such model that can calculate the exact effort required. So, artificial intelligence techniques are being used such as neural network, fuzzy logic, genetic algorithm, case based reasoning etc., which explains the radial basis function of neural network (RBFN) to calculate effort estimation that basically concentrates on the hidden layer neurons and basis functions width and evaluation criteria are based on MRE and MMRE which uses two different datasets tukutuku datasets and COCOMO datasets for empirical study.

Jaswinder Kaur [4] has mentioned that, as the software has started being largely developed for commercial purpose. There is a huge requirement of software effort estimation. Different researchers has developed different model to estimate the cost estimation, yet no effective method has been developed and also specifies the comparison between ANN (Artificial Neural Network) and other models to calculate the results using evaluation criteria based on MRE and MMRE. Finally, the main result is reliable and precise when applied to software datasets and the result obtained from ANN was more effective as compared to any other models.

The paper by B. Tirimula Rao, [5] has focused basically on the FLAN techniques of neural network. It is one of the most important tasks to calculate the software effort. Since, there is no particular formula to calculate effort estimation. FLAN (Functional Link Artificial Neural Network) technique has been used to calculate the effort estimation using the COCOMO datasets. Neural network techniques have been proposed to calculate effort estimation to minimize the computational complexity. These methods are suitable for complex problems but the complexity of software increases when number of inputs becomes large. So, FLAN techniques have been used in effort estimation as it reduces the computational complexity and hence is suitable for online applications. FLAN architecture is very simple as compare to others as it uses no hidden layers and COCOMO model is used as base model, to compute the cost of the software. This method provides more accurate results as compared to others because it uses proper train data and also trains the network architecture providing better results.

In Ref. [6] the author has explained that one of the biggest challenges for the software industry is to select the best approach to calculate the effort estimation cost of the software. Neural techniques have proved very effective in software effort estimation. ANN (Artificial Neural Network) and SVM (Support Vector Machine) techniques are combined together to calculate the performance indices Mean-Magnitude-Relative-Error (MMRE), Relative-Root-Square-Error (RRSE), Correlation Coefficient (CC), Root-Mean-Square-Error (RMSE), Mean-Square-Error (MSE), Mean-Absolute-Error (MAE), Relative-Absolute-Error (RAE), Sum-

Square-Error (SSE) and PRED have been used to compare the results obtained from these two methods, uses china dataset as the base model

The paper by Dr. Finnie, [7] has done a research study which compares three estimation techniques using function points. It was based on artificial neural network, regression analysis and case-based reasoning. After performing the analysis various points were observed with respect to these models. Regression techniques appeared to give poor result as comparison to neural network and case-based reasoning. Case –based reasoning was the best method because of its expert judgment approach that provides the result in support of human judgment.

The paper by Gerhard witting, [8] has mentioned the most effective requirement in terms of software is to find the best method for software effort estimation and Artificial Neural Network is considered as a good approach because of their capabilities to give good results in terms of large and complex software projects. Back-propagation neural network technique has been used for cost estimation. This neural network model was tested on actual data and stimulated data of commercial projects under typical software development conditions. The results were encouraging, with the network showing an ability to estimate development effort within 25% of actual effort more than 75% of the time for one large commercial data.

The paper by Abbas Heiat, [9], compares the neural network approach and old regression analyses in terms of mean value percentage error. The comparison was done in terms of multilayer perceptron and radial basis function based neural network to that of regression analyses which shows that neural network gives the best results and improved performance in terms of effort estimation.

5. COMPARATIVE ANALYSES

Table 2. Comparative analyses among different models of neural network to evaluate software effort estimation

Technique used	Description	Advantage
Radial Basis Function Neural Network (RBFN)	RBFN describes methods to calculate the effort estimation. For this he uses k-means clustering algorithm to find the structure of hidden neurons. RBFN network consist of three layers: input layers (cost drivers), hidden neurons and output layer. The basic dataset used is the COCOMO and tukutuku dataset.	This paper examines the radial basis width using different formulas and RBFN provides a better accuracy on effort estimation when tested on different projects.
ANN (Artificial Neural Network)	ANN architecture consists of input neurons that is attached to some weighted value and calculates the sum of n inputs and generates an output. Output is compared with a threshold value to determine whether the output is positive or negative.	It was noted that ANN gives the least MMRE when compared with other models (such as Halstead, Walston-Felix, Bailey-Basili) and has more accuracy than others.
Functional Link Artificial Neural Network (FLAN)	FLAN is also a technique of Neural Network that uses single layer feed forward mechanism. This mechanism consists of input neurons (cost driver) and generating resultant output (effort) by expanding the cost drivers. The development effort is calculated by the linear weighted sum of the outputs of the input neurons.	FLAN focuses on functional approximation and also reduces the complexity. It also uses the learning concept which makes it faster than other neural network technology which leads to better approximation.
Artificial neural Network and Support vector Machine	ANN (Artificial Neural Network) and SVM (Support vector machine) mechanism can be combined to calculate effort estimation. ANN uses the concept of error-back- propagation mechanism for learning and training. During the forward phase weights are fixed and during backward phase weights are updated according to the error calculated. SVM is a supervised learning mechanism that is based on classification and regression analysis. SVM is basically used to improve the accuracy.	It focuses on the learning techniques to compute the error. It gives a better result as weights can adjusted according to the training dataset.
Using function points with neural network, case- based reasoning and regression Analyses	(case-based Reasoning, Regression Analyses and Neural network) has been used to calculate the system size. Evaluation criteria are based on MRE and MMRE.	Case- based reasoning which uses expert judgement mechanism and neural network gives better result as compared with regression analyses.

Connectionist model using neural network techniques of back propagation	Comparative study on actual projects and commercial projects was done. The term connectionist model is being used because there are inputs that are in distorted form which leads to a complex relation between inputs and output.	It was calculated that the effort was within 25% of actual projects more than 75% of the time on large commercial data sets.
Artificial neural network and regression model	Instead of using function points individually to calculate development effort, neural network can be used that can evaluate various elements of development environment. And hence this model was named as network based software development estimation model	Gives a better performance as compared with the tradition method using only COCOMO or function – point (FP).

6. CONCLUSION

Different methods of neural network have been used to calculate effort estimation. Each and every technique focuses on providing best software effort estimation. Neural network is a good approach in estimating development effort. It was suggested that for complex and computationally large projects it's better to use neural network approach. But there is a need to examine accuracy of methods which mostly required in software effort estimation. It was analyzed that neuro based models have better estimation capability and hence can be used to calculate software effort estimation of all kinds of project.

7. FUTURE SCOPE

Even though, number of methods has been proposed for cost estimation, project management, finding complex input cost drivers. But the dynamic nature of project arises several problems in computing cost estimation and hence the complete method highly depends on several complex factors. Neuro model has provided better accuracy and optimization, but to have a better optimized result other emerging technique of artificial intelligence (eg. Genetic algorithms[10], fuzzy logic[11, 12] etc.) Can be used for manipulating real- world problems and provides better accuracy using these techniques.

8. REFERENCES

[1] Kathleen Peter, 2012. Software Project Estimation.

[2] Anil K.Jain, Jianchang Mao, KM. Mohiuddin, 1996. Artificial Neural Network: A Tutorial http://www.*csc.lsu.edu/~jianhua/nn.pdf*

[3] Sriman Srichandan, 2012. A new approach of software effort estimation using Radial basis function neural networks.

[4] Jaswinder Kaur, Satwinder Singh, Dr. Karanjeet Singh Kahlon, Pourush Bassi, 2010. Neural Network- A Novel Technique for Software Effort Estimation.

[5] B. Tirimula Rao, B. Sameet, G.Kiran Swathi, K.Vikram Gupta, Ch. Ravi Teja, S. Sumana , 2009. A Novel Neural Network Approach For Software Cost Estimation Using Functional Link Artificial Neural Network.

[6] Prabhakar, Maitreyeea Dutta**,** 2013. Prediction of Software Effort Using Artificial Neural Network and Support Vector Machine.(2013)

[7] Dr. G. R. Finnie, G. E. Wittig, J.M Desharnars, 1997. A Comparison of Software Effort Estimation Techniques: using Function Points with Neural networks, case-based reasoning and regression model.

[8] Gerhard Witting, Gavin Finnie, Estimating Software development effort with connectionist models.

[9] Abbas Heiat, 2002. Comaprison of artificial neural network and regression models for estimating software development effort.

[10] Li, Y.F. Xie, M. Goh, T.N (2007). A study of genetic algorithm for project selection for analogy based software cost estimation. IEEE International Conference on Industrial Engineering and Engineering Management, 1256-1260.

[11] Pedrycz, W. Peters, J.F. Ramanna, S (1999). A fuzzy set approach to cost estimation of software projects. IEEE Canadian Conference on Electrical and Computer Engineering, 2, 1068-1073.

[12] Shina Dhingra, Palvinder Singh Mann (2013). An Adaptive Neuro Fuzzy Approach for Software Development Time Estimation.