# BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT

Yelahanka, Bengaluru. 560064

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

## PROJECT SYNOPSIS

**Name of Guide:** *Prof. Ravi Kumar* **Batch No:** *C16* **Date of Submission:** *5-11-2022*

| NAME OF STUDENT | USN | Email-Id/Group Mail Id | Contact No. |
|---|---|---|---|
| Samuel Sampath Kumar | 1BY19IS144 | 1by19is144@bmsit.in | 9353366895 |
| Sudeep Rangan D R | 1BY19IS162 | 1by19is162@bmsit.in | 8217024560 |
| Apurv Jha | 1BY19IS028 | 1by18is028@bmsit.in | 6204882572 |
| Anamika Sharma | 1BY18IS129 | 1by18is129@bmsit.in | 9945136946 |

| | |
|---|---|
| **Project Title:** | **Software Cost Estimation Using Back Propagation Algorithm.** |
| **Project Execution Place** | Inhouse |
| **Project Category/Area** | Application Project |

## Abstract:

Software cost estimation is one of the most challenging tasks in software engineering. Over the past years the estimators have used parametric cost estimation models to establish software cost, however the challenges to accurate cost estimation keep evolving with the advancing technology.This project uses Back-Propagation neural networks for software cost estimation. A model based on Neural Network has been proposed that takes the KLOC of the project as input, uses COCOMO model parameters and gives cost as output. Artificial Neural Network represents a complex set of relationships between the effort and the cost drivers and is a potential tool for estimation. The proposed model estimates the software cost and helps project manager to provide fast and realistic estimates for the project effort and development time that in turn gives software cost.

## Introduction:

**Software Cost basically consists of:**

- ***Manpower*** i.e. number of engineering and management personnel allocated to the project as a function of time.

- ***Duration*** i.e. the amount of time required to complete the project.
- ***Effort*** i.e. the engineering and management effort required to complete a project.

Software cost estimation usually fails to accurately predict the actual costs or the time needed to develop the system. Software cost estimation models have two problems. Estimating software development costs with accuracy is very difficult. The most common approach for improving software cost estimates is to use empirical models. It becomes more challenging to predict the costs and schedule at the beginning of the project. Early prediction of completion time is necessary for proper advance planning and to avoid any type of risk of a project.

The purpose of this project is also to identify how many hidden layers of Neural networks are needed in order to have a good accuracy. Our results show that the Neural networks using backpropagation algorithms having two hidden layers gives more accurate results in comparison to other layers. That's why we also trained networks with three, four and five hidden layers and studies have shown that the two hidden layers are the most accurate in comparison with single, three, four and five hidden layers.

## Objectives:

- **Designing a Software cost estimation model for Object Oriented Systems:**
- **Applying ANN techniques for the basic model to improve the accuracy:**
- **Comparing with existing techniques and to prove the propose Software Effort Estimation model is more Efficient:**

## Problem Statement:

Software cost estimation is the set of procedures and techniques with a set of inputs that an organization uses to achieve a software cost estimate in terms of effort, manpower, duration etc. Generally, inaccurate estimates are because of:

- **More Hidden Layers:** Usually assumed that more layers means faster processing and results but in the case of back propagation algorithm 2 layers are proven to be more effective than 3,4,5 hidden layers.
- **Increasing Complexities:** As there are many factors involved in creating a software it becomes highly complex in predicting the accurate cost of the software therefore the applications regarding software cost estimation also becomes complex.
    - Problem with requirements
    - System Size
    - Maintenance Issues
    - Software Process and Process Maturity
    - Project Monitoring and Control  Lack of historical data
    - Lack of application domain expertise
- **Time Delays:** Using other machine learning models and algorithms the expected accuracy is not Reached and hence as these are also complex and comprise more layers, time delays are inevitable.

## Proposed System:

The existing system is too bulky and complex with noticeably more accuracy for more cost and also slower due to the complexities, hence the proposed system is less complex, easy to understand and affordable. This system is to be developed with the help of the back propagation algorithm from Neural Networks.

- **Neural Networks:**

A neural network is a non-linear type of model that receives its inspiration from the neural architecture of the human brain. A single neuron from the brain has three basic components - dendrites that provide for input from neighboring neurons, a soma for processing and axons for output to other neurons (Beale and Jackson, 1990. Each neuron may be connected to thousands of neighboring neurons via this network of dendrites and axons. This network allows the brain to achieve speed and power through massive parallel processing. Similarly, an artificial neural network contains inter- connections analogous to dendrites, processing nodes similar to somas in many regards and output connections that represent the axons. One of the prime advantages to neural network modeling is that there are no previous assumptions made about the relationship being modeled. Rather, the neural network examines data involving the phenomena in question and maps an approximation of the relationship (i.e., the underlying function) using a learning or training algorithm.

- **Use of neural Networks:**

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques (Ward et al., 1995). A trained neural network can be thought of as an expert in the category of information it has been given to analyze. This expert can then be used to provide projections given new situations of interest and answer "whatif" questions.

- **Back Propagation Algorithm:**

Back Propagation (BP) is a method for training multilayer feedforward networks (Fadalla et al., 2001). It works by training the output layer and then propagating the error calculated for these output neurons, back though the weights of the net, to train the neurons in the inner (hidden) layers. To understand how BP works consider the network shown in Fig 3.

- **COCOMO MODEL:**

The COCOMO model can be categorized into basic, intermediate, and detailed model. The basic model represents quick, early, and rough estimates of effort. The intermediate and detailed model includes more information in the form of cost drivers. The inputs of COCOMO model are:

(1) The estimated size of the software product in thousands of Delivered Source Instructions (KDSI) adjusted for code reuse.

(2) The project development mode given as a constant value B (also called the scaling factor).

(3) 15 cost drivers. The project development mode depends on one of the three categories of software development modes: organic, semi detached, and embedded. Each rating has a corresponding real number (effort multiplier), based upon the factor and the degree to which the factor can influence productivity [23, 26, 29]. The estimated effort in person-months (PM) for the intermediate COCOMO is given as:

$$Effort = A \times [Size]^{B} \times \prod_{i=1}^{15} EM_i$$

## <u>Methodology:</u>

### Learning By Human Brain:

The basic building block of the brain and the neural network is the neuron. The Basic human neuron adapted from (Beale and Jackson, 1990) is shown in Fig. .1

As described by (Beale and Jackson ,1990), all inputs to the soma (cell body) of the neuron arrive along with dendrites. Dendrites can also act as outputs interconnecting interneurons.

Mathematically, the dendrite's function can be approximated as a summation. Axons, on the other hand, are found only on output cells. The axon has an electrical potential. If excited past a threshold it will transmit an electrical signal. Axons terminate at synapses that connect it to the dendrite of another neuron. When the electrical input to a synapse reaches a threshold, it will pass the signal through to the dendrite to which it is connected. The human brain contains approximately 10 ° interconnected neurons creating its massively parallel computational capability.

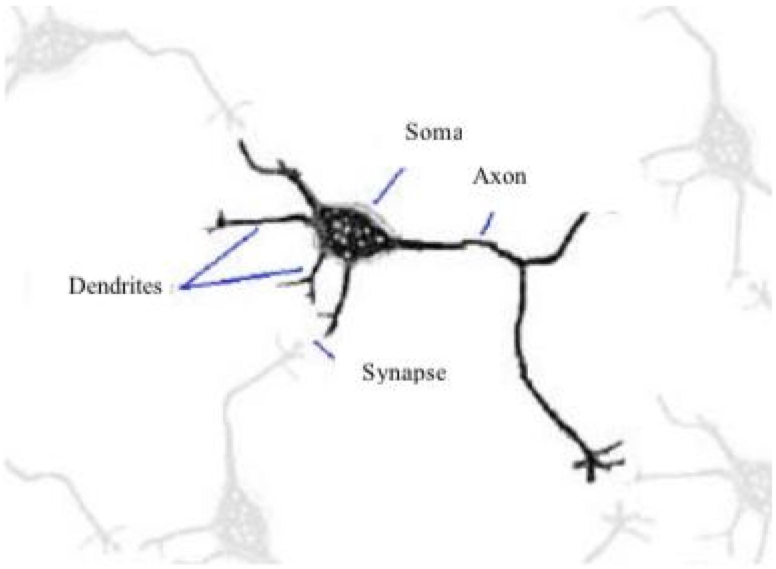


**Fig.1: Components of Human Brain cell(Neuron)**

**From Human Neurons To Artificial Neurons:**

The artificial neuron was developed in an effort to model the human neuron. The artificial neuron depicted in Fig. 2 was adapted from (Kartalopoulos and Stamatios, 1996) and (Haykin and Simon, 1994). Inputs enter the neuron and are multiplied by their respective synaptic weights. They are then summed and processed by an activation function. The activation function dampens or bound's the neuron's output. Figure 3 represent common activation functions which also happened to be used by the network tested during this research. The logistic or sigmoid function f(x) = 1/ (I+exp (-X)).

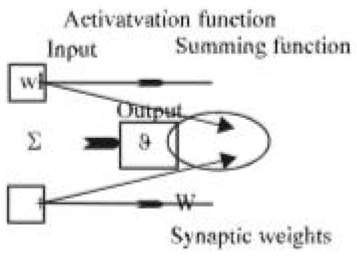


**Fig.2:Artificial Neuron Model**

To show the general technique of BP training, consider the neuron in the hidden layer H. This neuron has three weights feeding into it, WH.Wh2 and WH3. The training technique applied to this one neuron can be applied to all the other neurons throughout the network.

First, an input is applied to the network and the output is calculated. The output is then compared with the target and an error is calculated. The error is then multiplied by the derivative of the activation function (for example the sigmoid function).

$$\frac{\partial O}{\partial S} = O(1-O) \qquad (1)$$

so:

$$\delta_A = O_A(1-O_A)(T_A - O_A) \qquad (2)$$

This can be used to calculate Ab y multiplying learning rate ,I then calculate the new for neuron A, as follows

$$\Delta_A = l\,\delta_A \qquad (3)$$

$$w_A^{+} = w_A +\_\Delta_A \qquad (4)$$

Then propagate the value of 8 calculated for the output neurons, back through the weights, to the neurons of the hidden layer and hence calculate a value of 8 for these. This is done by multiplying the

$$\delta_H = O_H(1-O_H).(\delta_A.w_{A,H} + \delta_B.w_{B,H}) \qquad (5)$$

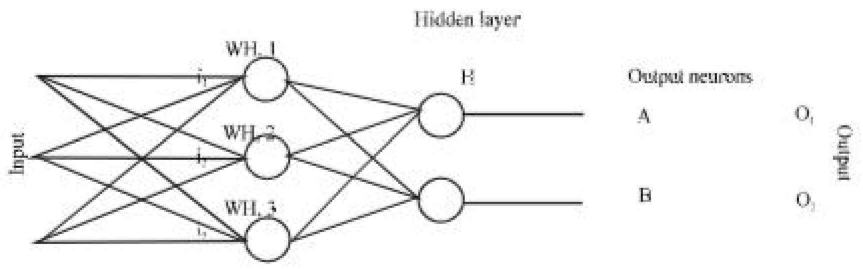
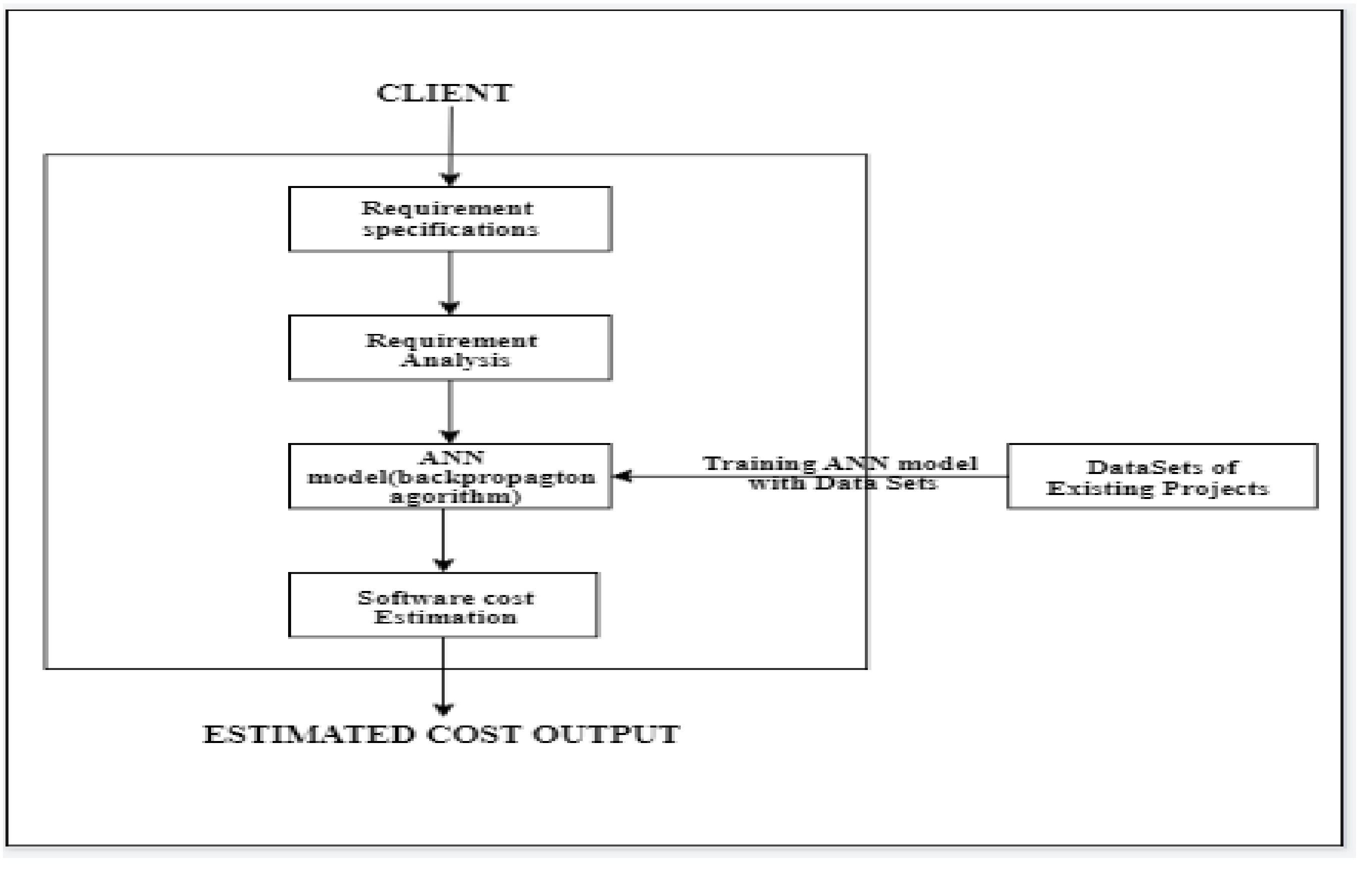


**Fig.3: General Multilayer Feedforward Network**


CLIENT
Requirement specifications
Requirement Analysis
ANN model(backpropagton agorithm)
Training ANN model with Data Sets
DataSets of Existing Projects
Software cost Estimation
ESTIMATED COST OUTPUT

## Steps Involved in estimating the software cost through this project:

- **Step 1:** Getting the requirement specification from the client
- **Step 2:** Analyzing the processing the requirement specifications given by the client
- **Step 3:** Feeding it the created software which is trained with datasets(These data sets contain the already created software details which include the overall cost of that software)
- **Step 4:** The software processes the given data with the back propagation algorithm and the cost is estimated and given as output

## System Requirement Specifications:

### Hardware Requirements:

- 4 GB RAM
- 5 GB free disk space
- x86 64-bit CPU (Intel / AMD architecture)

### Software Requirement:

Modern Operating System:

- IDE – Microsoft Visual Studio Code
- Web Browser – Firefox , Google Chrome
- Operating system – Windows 7 / Ubuntu 16.04
- JDK 19

## Conclusion:

This study has compared the forecasting accuracy of neural networks with back propagation algorithm using one, two, three, four and five hidden layers. Results showed that for this software cost estimation problem, the neural networks with back propagation algorithms using two hidden layers has given most accurate prediction in comparison to other layers.Software development cost estimation is a challenging task for both the industrial as well as academic communities. The accurate predictions during the early stages of development of a software project can greatly benefit the development team. There are several effort estimation models that can be used in forecasting software development effort.

## References

- Beale, R. and T. Jackson, 1990. Neural Computing: An Introduction, Adam Hilger, Bristol, England. Fadalla, A., Lin and Chien-Hua, 2001. An Analysis of the Applications of Neural Networks in
- Finance. Interfaces, pp: 112-122.
- Gately and J. Edward, 1996. Neural Networks for Financial Forecasting, John Wiley and Sons,
- New York.
- Haykin and Simon, Neural Networks: A Comprehensive Foundation, Macmillian College Publishing
- Company, New York, New York.
- Kanas, A., 2001. Neural network linear forecasts for stock returns. Int. J. Finance Econom.,
- 6: 245-254.
- Kartalopoulos and V. Stamatios, 1996. Understanding Neural Networks and Fuzzy Logic: Basic
- Concepts and Applications, IEEE Press, New York.
- Kryzanowski, L., M. Galler and D.W.Wright, 1993. Using artificial networks topick stocks. Financial
- Analyst's J., pp: 21-27.

**Signature Of Guide** **Signature of Coordinator** **Signature of HOD**