OPTIMIZATION TECHNIQUE FOR SOFTWARE COST ESTIMATION USING NEURAL NETWORK

S.P.SAM DHANA SEKAR AND A.ASKARUNISA

1Department of CSE, Veeramall engineering college, K.Singarakottai (P.O), Dindugal. 2 HOD/CSE, Vickram College of Engineering, Enathi (P.O), Madurai. Emails: sam.lakshia@gmail.com

Received: 12 September 2014, Revised and Accepted: 15 October 2014

ABSTRACT

Last few decade software accomplishment admiration models developed, authentic estimates of the software activity beneath development is still unachievable goal. Recently advisers are alive on the development of new models and the advance of the absolute ones application bogus intelligence techniques. Designing of ANN (Artificial Neural Network) to archetypal a circuitous set of accord amid the abased capricious (effort) and the absolute variables (cost drivers) makes an apparatus for estimation. This cardboards presents an achievement assay of Multi ANNs in accomplishment estimation. We accept apish Back propagation ANN created by MATLAB Neural Network Apparatus application NASA dataset.

Keywords: Effort, Drivers, Back propogation, Matlab, NASA dataset.

INTRODUCTION

Software bulk and accomplishment admiration is one of the lots of arduous issues in software activity management. Several estimates are complex to finer administer the software cost. It has become cold of every software engineering association to advance advantageous models that can accurately appraisal the software effort.

One of the a lot of broadly acclimated address is Cocomo (constructive bulk model) alien by Barry Boehm in 1981, and is still in use by software engineering community. Software development efforts admiration is the action of admiration the lot of astute use of accomplishment accomplishment appropriate to advance or advance software base on incomplete, ambiguous and/or blatant input. Accomplishment estimates may be acclimated as ascribe to activity plans, abundance plans, budgets, investment analyses, appraisement processes and behest rounds. A lot of the assay has focused on the architecture of academic software cost models.

The aboriginal models were about based on corruption assay or mathematically acquired from theories from added domains. Cocomo consists of a bureaucracy of three more abundant and authentic forms. The aboriginal level, Basic Cocomo is acceptable for quick, early, asperous adjustment of consequence estimates of software costs, but its accurateness is bound due to its abridgement of factors to annual for aberration in activity attributes (Cost Drivers). Average Cocomo takes these Bulk Drivers into annual and Abundant Cocomo additionally accounts for the access of alone activity phases.

Basic Cocomo computes software development accomplishment (and cost) as an action of affairs size. Affairs admeasurements is bidding in estimated bags of antecedent curve of cipher (SLOC), Cocomo applies to three classes of software projects:

- Amoebic projects - "small" teams with "good" acquaintance alive with "less than rigid" requirements.
- Semi-detached projects - "medium" teams with alloyed acquaintance alive with a mix of adamant and beneath than adamant requirements.
- Embedded projects - developed aural a set of "tight" constraints. It is as well aggregate of amoebic and semi-detached projects [Hardware, software, operational].

The basic Cocomo equations are:

Effort Activated \( E = ab \times (KLoC) \]

Development Time \( D = cd \times \text{[Effort Applied]} \times \text{[months]} \]

People appropriate \( P = \text{Accomplishment Activated / Development Time} \)

Where, KLoC is the estimated bulk of delivered curve (expressed in thousands) of cipher for project.

Intermediate Cocomo computes software development accomplishment as action of affairs admeasurements and a set of "cost drivers" that cover abstract appraisal of product, hardware, cadre and activity attributes. This addendum considers a set of four "cost drivers", anniversary with a bulk of accessory attributes:-

Product attributes
- Required software reliability
- Size of application database
- Complexity of the product

Hardware attributes
- Run-time performance constraints
- Memory constraints
- Volatility of the virtual machine environment
- Required turnabout time

Personnel attributes
- Analyst capability
- Software engineering capability
- Applications experience
- Virtual machine experience
- Programming language experience

Project attributes
- Use of software tools
- Application of software engineering methods
- Required development schedule

The Intermediate Cocomo formula is: \( E=a_i \times (KLoC) \times EAF \)

Where \( E \) is the effort applied in person-months, KLoC is the estimated number of thousands of delivered lines of code for the project, and EAF is the factor calculated above. Detailed Cocomo incorporates all characteristics of the average adaptation with an appraisal of the bulk driver's appulse on anniversary footfall [analysis, design, etc.] of the software engineering process. The
abundant archetypal uses altered accomplishment multipliers for anniversary bulk discriminant attribute. These Appearance Sensitive accomplishment multipliers are anniversary to actuate the bulk of accomplishment appropriate to complete anniversary phase.

In abundant COCOMO, the accomplishment is affected as action of affairs admeasurements and a set of bulk drivers accustomed according to anniversary appearance of software activity cycle.

An Abundant activity agenda is never static. The 5 phases of abundant COCOMO are:
- Plan and requirement.
- System Design.
- Detailed Design.
- Module code and test.
- Integration and test.

In order to make accurate estimates cost estimation techniques are divided into two main categories
- Parametric Models or Algorithmic Models that are derived from numerical analysis of historical projects data
- Non Parametric or Non Algorithmic Models based on set of artificial intelligence techniques like neural networks, genetic Algorithm, rule based induction, etc.

This paper discusses Neural Network non parametric cost estimation technique.

LITERATURE REVIEW

ANN in Effort Estimation

Artificial Neural Arrangement is acclimated in accomplishment admiration due to its adeptness to apprentice from antecedent data. It is as well able to archetypal circuitous relationships amid the abased (effort) and absolute variables (cost drivers). In addition, it has the adeptness to generalize from the training abstracts set appropriately enabling it to aftermath adequate aftereffect for ahead concealed data. Most of the plan in the appliance of neural arrangement to accomplishment admiration fabricated use of Back-propagation algorithm.

Artificial Neural Network (ANN) is a massively alongside adaptive arrangement of simple nonlinear accretion elements alleged Neurons, which are advised to abstruse and archetypal some of the functionality of the animal afraid arrangement in an attack to partially abduction some of its computational strengths.

An artificial neural network comprises of eight basic components: (i) neurons, (ii) activation function, (iii) signal function, (iv) pattern of connectivity, (v) activity aggregation rule, (vi) activation rule, (vii) learning rule and (viii) environment.

After an ANN is created it accepts to go through the action of learning or training. The action of modifying the weights in the access amid arrangement layers with the cold of accomplishing the accepted achievement is alleged training a network. There are two approaches for training supervised and unsupervised. In supervised training, both the inputs and the outputs are provided.

The network afresh processes the inputs, compares its consistent outputs adjoin the adapted outputs and absurdity is calculated. In unsupervised training, the network is provided with inputs but not with adapted outputs. The network itself accept to afresh adjudicate what appearance it will use to accumulation the ascribe data.

Most of the plan in the appliance of neural network to accomplishment admiration fabricated use of Back-propagation algorithm and Back propagation. Abounding altered models of neural nets accept been proposed for analytic abounding circuitous absolute activity problems.

The 7 steps for effort estimation using ANN can be summarized as follows:

1. Abstracts Collection: Collect abstracts for ahead developed projects like LOC, adjustment used, and added characteristics.
2. Division of dataset: Divide the amount of abstracts into two locations training set & validation set.
3. ANN Design: Design the neural arrangement with amount of neurons in input layers aforementioned as the amount of characteristics of the project.
4. Training: Feed the training set aboriginal to alternation the neural network.
5. Validation: After training is over afresh validate the ANN with the validation set data.
6. Testing: Finally analysis the created ANN by agriculture analysis dataset.
7. Absurdity calculation: Check the achievement of the ANN. If satisfactory afresh stop, abroad afresh go to step (3) accomplish some changes to the arrangement ambit and proceed.

RESEARCH METHOD

Back-Propagation Learning rule: Back-Propagation Learning (BPL) algorithm was invented in 1969 for learning in multilayer network. The back-propagation algorithm trains a accustomed augmented advanced multilayer neural arrangement for a accustomed set of ascribe patterns with accepted classifications. If anniversary access of the sample set is presented to the network, the arrangement examines its achievement acknowledgment to the sample ascribe pattern. The achievement acknowledgment is again compared to the accepted and adapted achievement and the absurdity amount is calculated. Based on the error, the affiliation weights are adjusted. The back-propagation algorithm is based on Widrow-Hof delta learning rule in which the weight aclimation is done through mean square error of the achievement acknowledgment to the sample input. The set of these sample patterns are again presented to the arrangement until the absurdity amount is minimized. The back-propagation neural network has input layer, hidden band and one achievement (output) layer. Ascribe signals transmitted from ascribe (input) to hidden and hidden to achievement band and absurdity arresting from achievement to hidden and hidden to input.

Back-propagation Learning algorithm uses training abstracts to acclimatize the weights and beginning of neurons so as to abbreviate the error. It is based on the differences amid the absolute and the adapted output. It works by applying the activity coast aphorism to feed-forward network. The algorithm involves two phases, the advanced appearance that occurs if the inputs (external stimuli) are presented to the neurons of the ascribe band and are broadcast advanced to compute the achievement and the astern phase, if the algorithm performs modifications in the backward direction.

Steps of the algorithms are the following:

Step 1: Initialize weights with small, accidental values
Step 2: While endlessly action is not true
For each training pair (input/output):
1. Each input unit broadcasts its value to all hidden units.
2. Each hidden unit sums its input signals & applies activation function to compute its output signal.
3. Each hidden unit sends its signal to the output units.
4. Each output unit sums its input signals & applies its activation function to compute its output signal.
Step 3: Each output computes its error term; its own weight correction term and it bias (Threshold) correction term & sends it to layer below
Step 4: Each hidden unit sums its delta inputs from above & multiplies by the derivative of its activation function; it also computes its own weight correction term and its bias correction term
Step 5: Each output unit updates its weights and bias
Step 6: Each hidden unit updates its weights and bias:
  a. Each training cycle is called an epoch. The weights are updated in each cycle.
The proposed anatomy of neural arrangement is customized to board the COCOMO II post architectural model. There are 5 calibration factors denoted by SF and 17 accomplishment multipliers denoted by EM. The use of neural all the inputs of calibration factors and accomplishment multipliers are provided through the neurons of ascribe band as apparent in amount 2 with bias. The net ascribe of weights and provided to the network. The weights associated with scale factors \( v_j \) are the multipliers are initialized as \( w_i = 1 \) for \( i = 1 \) to 17, learning rate \( \alpha = 0.001 \) and bias 1 =log (A). The inputs are accustomed and accumulate to the weights and provided to the network. The weights associated with scale factors \( v_j = 0 \) for \( j = 1 \) to 5 and bias 2 is 1.01.

**Abbreviations used:**
- PM : Person per month
- SIZE : Line of Code in KLOC
- SF : Scale factors
- EM : Effort Multipliers
- QO : Initial weight associated with scale factors
- P0 : Initial weight associated with scale factors

**Step 1:** Calculate PM according to COCOMO II model of Berry Boehm

\[
P_M = A \cdot (\text{SIZE})^{0.4} \cdot (\text{SIZE})^{a} \cdot \sum_{i=1}^{3} (\text{EM})^i
\]

**Step 2:** Calculate output of hidden layer neuron as:

Net input to hidden layer node 1 (for scale factors \( w_i \) is the weights) = \( N_1 \)

\[
((gP + \log (\text{size}))Bias1 + \sum_{j=1}^{5} (Wj + \log (\text{size}))(\text{SF}j)) = P
\]

\( F(\text{net}) = F(N_1) \) i.e. output of hidden layer node 1 (for scale factors)

\( F(N_1) = 1 / 1 + \exp (-N_1) = S \)

Net input to hidden layer node 2 (for effort multiplier \( v_j \) are the weights) = \( N_2 \)

\[
Po \ Bias = \sum_{j=1}^{5} (Vj \ log (EMj))
\]

\( F(\text{net}) \) i.e. output of hidden layer node 2 (for effort multiplier) = \( F(N_2) \)

\[
F(N_2) = 1 / 1 + \exp (-N_2) = T
\]

**Step 3:** Calculate Net input to output layer node as:

\[
PM_o = S \cdot P + TQ \quad \text{Where P and Q are weights from hidden layer nodes to output layer node. P=1 and Q=1}
\]

**Step 4:** Check if \( PM_o < PM \) then output =1 and exit

Else output =0 and go to step 5

**Step 5:** weights are updated as.

\[
Wt(\text{new}) = Wt(\text{old}) + (\text{desired o/p} - \text{actual o/p}) \cdot \text{input}
\]

**Data Collection:**
Results in neural networks will be affected by demography actual abstracts of 50 projects which are disconnected into three parts: 20 projects abstracts for training the network, 10 projects for acceptance the arrangement and 10 projects for testing the network.

**TABLE 3.1 DATA USED FOR TRAINING THE NEURAL NETWORK**

<table>
<thead>
<tr>
<th>Project No</th>
<th>Size</th>
<th>Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.20</td>
<td>9.00</td>
</tr>
<tr>
<td>2</td>
<td>5.00</td>
<td>8.40</td>
</tr>
<tr>
<td>3</td>
<td>7.80</td>
<td>7.30</td>
</tr>
<tr>
<td>4</td>
<td>9.70</td>
<td>15.60</td>
</tr>
<tr>
<td>5</td>
<td>12.50</td>
<td>23.90</td>
</tr>
<tr>
<td>6</td>
<td>12.80</td>
<td>18.90</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>73.0</td>
</tr>
<tr>
<td>8</td>
<td>24</td>
<td>49.3</td>
</tr>
<tr>
<td>9</td>
<td>28</td>
<td>65.8</td>
</tr>
<tr>
<td>10</td>
<td>29</td>
<td>40.1</td>
</tr>
<tr>
<td>11</td>
<td>30</td>
<td>32.2</td>
</tr>
<tr>
<td>12</td>
<td>31.10</td>
<td>39.60</td>
</tr>
<tr>
<td>13</td>
<td>35</td>
<td>52.6</td>
</tr>
<tr>
<td>14</td>
<td>39</td>
<td>72.0</td>
</tr>
<tr>
<td>15</td>
<td>40</td>
<td>27.0</td>
</tr>
<tr>
<td>16</td>
<td>41</td>
<td>95.5</td>
</tr>
<tr>
<td>17</td>
<td>46.60</td>
<td>96.00</td>
</tr>
<tr>
<td>18</td>
<td>46.50</td>
<td>79.00</td>
</tr>
<tr>
<td>19</td>
<td>52</td>
<td>58.6</td>
</tr>
<tr>
<td>20</td>
<td>57</td>
<td>71.1</td>
</tr>
</tbody>
</table>

**TABLE 3.2 DATA USED FOR VALIDATING THE NEURAL NETWORK**

<table>
<thead>
<tr>
<th>Project No</th>
<th>Size</th>
<th>Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>31</td>
<td>2.10</td>
<td>5.00</td>
</tr>
<tr>
<td>32</td>
<td>3.10</td>
<td>7.00</td>
</tr>
<tr>
<td>33</td>
<td>21.50</td>
<td>28.50</td>
</tr>
<tr>
<td>34</td>
<td>22</td>
<td>19.1</td>
</tr>
<tr>
<td>35</td>
<td>54</td>
<td>138.8</td>
</tr>
<tr>
<td>36</td>
<td>54.50</td>
<td>90.80</td>
</tr>
<tr>
<td>37</td>
<td>62</td>
<td>189.5</td>
</tr>
<tr>
<td>38</td>
<td>67.50</td>
<td>98.40</td>
</tr>
<tr>
<td>39</td>
<td>318</td>
<td>692.1</td>
</tr>
<tr>
<td>40</td>
<td>450</td>
<td>1107.3</td>
</tr>
</tbody>
</table>

**TABLE 3.3 DATA USED FOR TESTING THE NEURAL NETWORK**

<table>
<thead>
<tr>
<th>Project No</th>
<th>Size</th>
<th>Effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td>10.50</td>
<td>10.30</td>
</tr>
<tr>
<td>42</td>
<td>42</td>
<td>78.9</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>23.2</td>
</tr>
<tr>
<td>44</td>
<td>48</td>
<td>84.9</td>
</tr>
<tr>
<td>45</td>
<td>50</td>
<td>84.9</td>
</tr>
<tr>
<td>46</td>
<td>78.60</td>
<td>98.7</td>
</tr>
<tr>
<td>47</td>
<td>130</td>
<td>673.7</td>
</tr>
<tr>
<td>48</td>
<td>165</td>
<td>246.9</td>
</tr>
<tr>
<td>49</td>
<td>200</td>
<td>130.3</td>
</tr>
<tr>
<td>50</td>
<td>214</td>
<td>86.9</td>
</tr>
</tbody>
</table>

**Experimental parameters**
Parameters used for performing the operation in neural networks are as follows

**TABLE 3.4 OPERATION TABLE**
Future plan on these capacity should cover application an accomplishment achievement abstracts set for which the bulk of abstracts accessible is not a constraint. A neural arrangement accomplished on this abstracts set would accommodate an added reliable appraisal of neural networks adeptness to aftermath a superior accomplishment estimate. In adjustment to accretion added acumen to the tremendous box attributes of the neural arrangement amount estimate, accommodation copse could be examined.

The accommodation copse could appearance the point at which an attributes amount changes the software’s cost, and that attributes accept the better appulse on the Software’s cost.

The abstracts conception adjustment could as well be advised in adjustment to advance its usefulness. The adjustment as currently constituted provides added babbie in the actualize abstracts than is desired. By introducing added constraints and Precedent relationships in the abstracts conception process, the bulk of babbie present would be reduced. This would allow the abstracts conception action to be acclimated on networks that accomplished an acceptable after effect amount if application their abject abstracts set.

REFERENCES

