

A progressive model for effort estimation for web development

Kohli PJ ,JIIT Noida 62 , D Soni JIIT Noida 128

Corresponding Author: Kohli PJ

ABSTRACT:

With the development and advancements of new areas and languages like R, python, and fields like Artificial intelligence and data learning, we want to utilize this in the field of software engineering, for predicting terms like manpower, time and effort. As worked by individuals in industry and the scholarly community read many papers in which they have used these techniques for estimating parameters like schedule, effort and like a vise. In current time it has been warmly acknowledged that coordinated advancement is a better approach than customary programming improvement with the methodologies like intelligent advancement, incremental conveyance and so forth. This development and usage of web throughout the years has become greater and greater so to accomplish it speedier in time utilizing nimble for web improvement is a decent decision. In this paper with current circumstances, customer's fulfillment is essential regardless of whether the necessities often change, and subsequently, the customary techniques feel choked or suffocated in this way these are not appropriate. As an option, python programming with improvement surfaced. It was observed to be generally acknowledged here we are using only meaningful informational index from IFPUG subjecting into a refresh exertion estimation, when contrasted with more established techniques like cocomo models, non-algometric model, for web application gauges utilizing the story point approach.

Key words

Genetic programming, Artificial Neural Networks, Relu function, PReLU, deep leaning, scrum, story point

Date of Submission: 22-06-2019

Date of acceptance: 08-07-2019

I. INTRODUCTION

The estimation has been a very vital activity for software development when we were just little so to being mature, estimation has been important to our growth, same is the case for projects. Every aspect of developing a project is vital. Estimation in software has been an area of research for quite some time.

With the coming of the existence of a new language or a new platform, it is a topic explored by researchers. As it is a crucial phase in software development, the quantity of workforce and time is required to build software is predicted in this stage.

Planning and estimation are two very important phases. one has to be completed until the final post-production phase. These estimates are important, for scheduling activities, assigning costs and dates for the project. We want these estimates to be as close as actual values.

There are some methods like such as experience judgment, algorithmic estimation, parametric estimation, machine learning are used for effort estimation.

Among all the various models we have Artificial Neural Networks (ANN) which is one of the new and promising options. In which a combination of COCOMO (1+2) ANN can show various complex non-linear relationships which can be used for evaluating these quantities. Many devopres and people from academia and researchers prefer these to extract and compute and various directly or indirectly related parameters from effort estimates.

Apart from the tradition Software model(TSM), we have agile Software Development (ASD) models which have some striking differences. In agile the estimation is done in a very popular but ad-hoc manner way to get the requirements for the task at hand which are the user story .which is a very casual way of getting from the user /customer wants/ what he is expecting for the development team. many agile popular methods as given below are listed below, 1)Extreme Programming (XP), 2)Scrum, 3)Feature-Driven 4)Development (FDD), 5)Dynamic System Development Method (DSDM), 6) Adaptive Software Development (ASD), 7) Crystal and 8) Lean Software Development (LD). One popular way the capture user perception is a user story (US) the US are a technique to capture requirements. They are very vital parts of an agile approach that the entire concentration is on talking and getting to know what customer wants this will help us to understand what the customer wants, in a describer format as in table1.

Format	As A	I Need	So That
Meaning	Who	What	Why
Example	HR manager	I want to create a quiz	I understand a candidate knowledge

Table 1: Format of a user story, example for agile

The user story is a very popular technique used in agile to capture user understanding, for what he expects the system over the years. A lot of work has been done in this area and gained maturity. There are many researchers working on effort estimation over a few years. lately, years Analysis of various domains has exploited the benefits of ANN given. Many researchers have applied Artificial neural network to compute effort as against the well discussed and popular techniques like COCOMO [7, 8,]. Most of these techniques use a combination of ANN with back propagation or use feed forward network. These models as in ANN have demonstrated the capacity to deliver outcomes about like , done and measured by the human way of thinking and experts .So use this capacity. of ANN frameworks

In these models, working on ANN has demonstrated, with playing around or updating weights and forward propagation techniques can be seen as good results, as a black box [7]. Nobody can explain how and what happens and one may vaguely express how ANN works, it can exactly describe what and how it happens in the weights and layers and. thus these, modifications not only try to better the performance, but they also give some sustainable explanations of what is happening inside the model. and experimental designs. Few of well-documented works by authors Kaushik, Kumar give a feed-forward neural network model based on COCOMO and shows that the output of the model is close to the actual effort value [8]. COCOMO 81, Masaki data-set is used in this work. We use evaluation criteria used[14]. Outputs are modified through a logarithmic version of COCOMO and comparisons are made with that value over perception value. Unfortunately, the estimates have done in the past time which relies wholly on COCOMO versions, time is not much accurate. As mostly rely on expert opinion. According to the Molokken and Jorgensen report [1], Reading the literature available for agile-based web development we read many works on web-based ANN for effort estimation. Seems to be more promising, innovative and challenging and more directed to get in the results of the effort. Keaveney et al. [15] investigated the applicability of conventional estimation techniques.

Researching through literature creates the impression that ANNs is a viable alternative to other old methods are used in many cases for effort estimation purposes. Along these lines, a survey on the usage of ANNs for effort estimation has been prepared. Wen et al. [10] assessed distinctive machine learning methods based programming effort, estimation models. It was seen by the examination that machine learning-based effort estimation models perform better than non-machine learning models. Idri et al. [11] suffocated henceforth these are not fitting. As a choice, swift programming change surfaced and is seen to be for the most part recognized here we are tackling an enlightening file from IFPUG subjecting into an invigorating effort estimation, when appeared differently in relation to more settled methods like ace supposition, non-almometric model, for web application checks using the story point approach.

Have outlined another RBFN organize based model with the end goal of programming improvement exertion estimation. Results demonstrated that the evaluations delivered by RBFN arrange the dis*-play, are enormously enhanced utilizing a satisfactory equation for width. A FLANN [12] was proposed by Rao et. al. for exertion estimation and to decrease the computational complexities with the goal that the neural net ends up plainly helpful for on-line applications. Parag C. Pendharkar [14] have proposed a PNN approach for anticipating a product.

development parameter and a probability measure which denotes the chances of the actual value of the parameter being less than its estimated value at the same time. This PNN approach was then compared with CHAID. Results indicate that PNN performs similar to the CHAID, but provides superior probability estimates. Adriano L. Oliveira et al. [15] have proposed and investigated the application of the GA for selecting an optimized feature subset and improving SVR parameters all the while aiming at enhancing the exactness of the software effort estimates towards agile development approaches by underscoring on the case studies of agile methods utilized within diverse organizations. Andreas Schmietendorf et al. [14]We use ReLU which is rectified linear unit) has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. ReLU's machinery is more like a real neuron in your body [15]

With a list of various activation functions being liner and non-liner both each has its advantaged and pitfalls .these being liner with are Identity function, Binary step function. Bipolar step functions are some of the linear functions. Whereas few non-linear are sigmoidal function, Binary sigmoid function bipolar sigmoid functions, tanh. arc tah ,soft sign [16][17]

Problem definition

We are in this paper working in the data sets as published in ISGSG, more in detail look for china data set

We performed radial base Ann, back propagation network and finally, ReLU (rectified linear unit) has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. ReLUs' machinery is more like a real neuron in your body.

Sample of the of data investigated in the paper is like is of the form in tab1

Tab 1: a view of the data set

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S		
1	ID	AFP	Input	Output	Enquiry	File	Interfac	Added	Change	Deleted	PDR	AF	PDR	UF	NPDR	ANPDU	Resourc	Dev.Typ	Duration	N effort	Effort
2	1	1587	774	260	340	128	0	1502	0	0	4.7	5	4.7	5	4	0	4	7490	749		
3	2	260	9	4	3	193	41	51	138	61	16	16.6	16	16.6	2	0	17	4150	415		
4	3	152	25	33	28	42	35	163	0	0	4.4	4.1	4.4	4.1	1	0	9	668	66		
5	4	252	151	28	8	89	0	69	153	4	12.8	14.3	15.5	17.3	1	0	4	3901	303		
6	5	292	93	0	194	20	0	0	307	0	10.3	9.8	12.4	11.7	1	0	13	3607	299		
7	6	83	63	0	24	0	0	0	87	0	16.1	15.3	19.3	18.5	1	0	4	1606	133		
8	7	79	24	0	23	30	0	0	77	0	20.3	20.9	24.5	25.1	1	0	6	3936	160		
9	8	97	0	108	7	0	5	120	0	0	11.9	9.7	11.9	9.7	2	0	7	1158	115		
10	9	116	0	23	58	14	20	81	34	0	10.7	10.8	12.9	13	1	0	6	1498	124		
11	10	52	39	7	0	0	0	0	46	0	54.8	73.3	78.1	88.3	1	0	7	4063	327		
12	11	465	209	129	24	83	15	460	0	0	21.9	22.2	21.9	22.2	1	0	9	10200	1020		
13	12	67	32	5	16	7	0	25	35	0	25.4	28.4	30.6	34.2	1	0	7	2053	170		
14	13	199	0	115	57	0	42	214	0	0	13.3	12.3	15.2	14.2	2	0	7	3034	264		
15	14	176	13	54	54	40	7	168	0	0	19	19.9	19	19.9	1	0	26	3348	334		
16	15	391	208	26	81	25	0	38	302	0	1.7	2	2.1	2.4	1	0	7	814	57		
17	16	263	65	45	101	42	10	176	87	0	3.5	3.5	3.5	3.5	1	0	3	911	91		
18	17	42	12	15	3	7	15	52	0	0	59.4	48	59.4	48	1	0	6.4	2496	249		
19	18	190	98	20	16	63	5	160	42	0	6.2	5.8	6.2	5.6	1	0	10	1171	117		
20	19	245	105	28	18	58	0	19	190	0	14.4	16.9	14.4	16.9	1	0	13	3532	353		
21	20	77	28	0	42	0	0	0	70	0	5.7	6.2	6.8	7.5	1	0	1	525	43		
22	21	355	278	0	73	0	0	0	351	0	2.6	2.6	3.1	3.1	1	0	4	1095	90		
23	22	3156	2075	525	97	0	0	28	12	2657	2.9	3.4	3.5	4.1	1	0	6	10957	908		
24	23	46	0	28	0	25	0	28	25	0	7.5	6.5	7.5	6.5	1	0	6	344	34		
25	24	56	14	12	15	7	5	53	0	0	5.3	5.6	5.3	5.6	4	0	3	296	29		
26	25	106	65	4	12	14	0	35	53	7	33	36.9	39.8	44.4	1	0	13	4220	350		
27	26	71	31	28	9	7	0	31	23	21	3.5	3.3	4.2	3.9	1	0	1	296	24		

Proposed model :

We are working on python 3.2 version of python using the NumPy,SciPy and Scikit libraries with anaconda package

Created ANN with Relu /PReLU model for deep learning .the data sets used for it consists of 80 columns and 60 rows

step 1: test n train(check for alternative)

Step 2: with the data set relevant columns issues . story point .LOC (the yellow marked are inputs)

Step3: added columns like staff, duration ,Effort which will be output columns (the effort marked as pink is output)

step4: clean data remove the outlier values

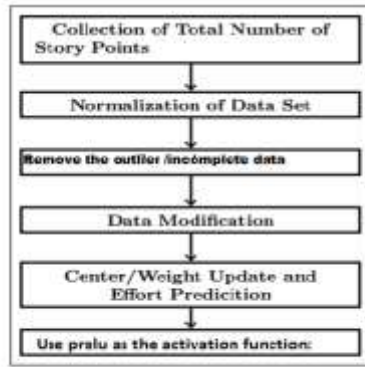
Step 5 random nos as values to k to give weights ()

Step6 :activation function is ReLU , Prelu , to obtain more promising results

Step :7 apply fuzzification

Step 8: Result obtained as given below

Results are shown in a table grid and a graphical representation of results. Also parameters of the system, system configuration, like ANN epoch count, learning rate, layers and neurons in these layers, K-Means set fullness ratio, set count are shown in order to give user information about These system parameters



Proposed Steps to Estimate Effort using Various Neural Networks

Experimental details

Diagram for the model depicting steps to be used

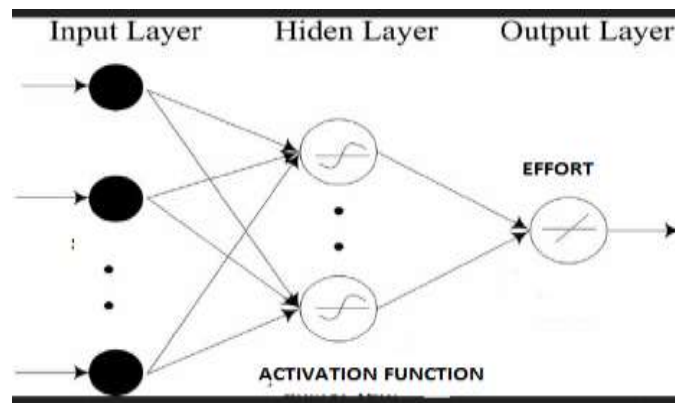


Diagram1: ANN effort estimation method

For our model

1. we have 4 inputs (Input /Output,Enquiry,File)
2. We have tried many activation function but the best rest was obtained with Prelu the activation function
3. using 70 :30 train and test scheme
4. with 10 hidden layers
5. Executing 50-100 epochs
6. Output is Duration /N effort/ Effort

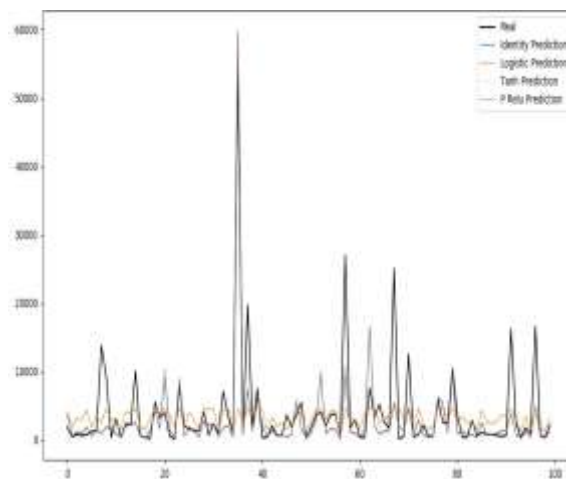
Experimental step-up

We are working on python 3.2 version of python using the NumPy and SciPy,Scikit libraries with anaconda package working on Ubuntu open source software form laptop ,desktop to cloud

Results

We have trained the model for an 20 -80 percentage of the data-set ,as in test n train ratio for the

Activatio nFunction	No of epocs	No of I/P columns	No of Hidden Layers	Fidelity count
Relu	100	3	10	0.953
PRelu	50	3	10	100.002



II. CONCLUSION

Effort estimation is a very important phase of software development. It needs to be done Very specifically in order to hand over the desired product at decided time, without incurring any additional cost. Currently, agile methods in software development for web-based requirements has gained much popularly. ANN models are an obvious choice . For effort prediction and sufficiency to handle missing or noisy input there are various work on ANN with many activation function like sigmoid, linear, softmax none was done for relu which is Rectified linear unit, Leaky rectified linear unit (Leaky ReLU) Randomized leaky rectified linear unit (RReLU). and its variation to give promising results

REFERENCE

- [1]. A. Stellman and J. Greene, Applied Software Project Management. Sebastopol, CA: O'Reilly, 2006.
- [2]. B. Hughes and M. Cotterell, Software Project Management, 4th Ed. New York, NY: McGraw Hill, 2004.
- [3]. Ch. S. Reddy and KSVN Raju, "A Concise Neural Network Model for Estimating Software Effort", International Journal of Recent Trends in Engineering, Issue. 1, vol. 1, pp. 188-193, May 2009.
- [4]. Project Management Institute, PMBOK - A Guide to the Project Management Body of Knowledge, 4th Ed. Newtown Square, PA: Project Management Institute, 2008.
- [5]. B. Boehm, Software Engineering Economics. Englewood Cliffs, NJ: Prentice Hall, 1994.
- [6]. J. Han and M. Kamber, Data Mining Concepts and Techniques, 2nd ed. Waltham, MA: Morgan Kauffman Publishers, 2006.
- [7]. Ch. S. Reddy and KSVN Raju, "A Concise Neural Network Model for Estimating Software Effort", International Journal of Recent Trends in Engineering, Issue. 1, vol. 1, pp. 188-193, May 2009.
- [8]. A. Kaushik, A. Chauhan, D. Mittal, and S. Gupta, "COCOMO Estimates Using Neural Networks", IJISA, 9, pp. 22-28, August 2012.
- [9]. A. Idri, T.M. Khoshgoftaar and A. Abran, "Can neural networks be easily interpreted in software cost estimation?," Proceedings of the IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'02, vol. 2, pp. 1162-1167, May 2002.
- [10]. Y. Kultur, B. Turhan, A. Bener, "Ensemble of neural networks with associative memory (ENNA) for estimating software development costs", Knowledge-Based Systems, vol. 22, pp. 395-402, August 2009.
- [11]. M. Jørgensen and M. Shepperd, "A Systematic Review of Software Development Cost Estimation Studies", IEEE transactions on software engineering, vol. 33, pp. 33-53, January 2007.
- [12]. S. Aljhadali and A. Sheta, "Software effort estimation by tuning COOCMO model parameters using differential evolution", ACS/IEEE International Conference on Computer Systems and Applications - AICCSA 2010, pp. 1-6, May 2010
- [13]. V. S. Dave and K. Dutta, "Application of Feed-Forward Neural Network in Estimation of Software Effort", IJCA, 2011.
- [14]. L.C. Briand, I. Wieczorek, K.E. Emam and D. Surmann, "An Assessment and Comparison of Common Software Cost Estimation Modeling Techniques," ISERN-1998-27, pp.313-323, 1999.
- [15]. PROMISE COCOMO Data Sets, <http://promise.site.uottawa.ca/SERespository/datasets-page.html> (accessed on May 10, 2013)

Kohli PJ" A progressive model for effort estimation for web development" International Journal of Research in Engineering and Science (IJRES), vol. 07, no. 2, 2019, pp. 01-05