An investigation of artificial neural networks(ANN) based prediction systems in software project management

Authors: Iris Fabiana de Barcelos Tronto , José Demísio Simões da Silva, Nilson Sant'Anna.
Publication: Elsevier.
Affiliation: Laboratory for Computing and Applied Mathematics – LAC, Brazilian National Institute for Space Research – INPE.

Problem Domain

This problem comes under the domain of software development.

Why it’s important to Solve

The success or failure of a software project depends on the product's quality and reliability. The predictions of defects are essential since it helps direct test effort, reduce costs, and improve software quality.
If we underestimate the effort, it may increase the pressure on the employees and, in turn, lead to a compromise in developing the fully functional product(that may increase the risk of low quality deliveries),

and setting short schedule(that may result in loss of credibility as deadlines are missed).
People start padding these and tend to overestimate the effort which will lead to some other issues like:
The project is then likely to cost more than it should (a negative impact on the bottom line), take longer to deliver than necessary (resulting in lost opportunities), and delay the use of the resources in the next project.

Other Prior Solutions

The other possible solutions for software estimation are APF, SLIM, and COCOMO methods.

Adaptive Project Framework, also known as Adaptive Project Management, is a software management methodology that accommodates certain unknown factors that can arise while working on the project. Such that the team can instantly respond to it.

The core principle of APF is 'learning by doing.'

SLIM estimating model is one of the first algorithmic models for estimating the effort using various project parameters, characteristics, attributes, and KLOC in the calculation.

COCOMO (Constructive Cost Model) is a prediction model based on the number of lines(LOC). It is a widely used technique in estimating the effort of a software project and is based on the study 63 projects.

According to the cocomo model, based on the project characteristics, the project's system can be of three categories: Organic, Semi-detached, Embedded.

The types of cocomo models are:

- Basic Model
- Intermediate Model
- Detailed Model

Approach to solve

The approach followed in the paper was to use some Machine learning and Deep learning techniques, as we already know that these techniques are very good at capturing some patterns in the data. They can independently adapt to the new data, which they have never seen, without explicitly programmed.

The prediction techniques used are:

- Linear Regression(Stepwise)
- Artificial Neural Networks

- **Linear Regression**

It is a technique that tries to find a linear relationship between the dependent and independent variables, minimizing the distance between the line and all the points.

It finds an optimal line in a multi-dimensional space.

$$Y = X_1 + X_2 + X_3 + X_4 + ... + X_n$$

- **Artificial Neural Networks**

A widely used predictive technique to find non-linear patterns by finding an optimal set of weights between its predictive units(neurons).

The most commonly used ANN in the software estimation is a feedforward multilayer perceptron network, which is trained by the gradient descent backpropagation algorithm using a differentiable activation function.

- **Variable Selection**

The dependent variable is the Effort, and the independent variables are 16 cost driver variables, which are used in estimation.

- **Dataset Division**

Before model fitting, six different training and testing datasets were created by removing every sixth project starting from the first project.
After this, the data is transformed, encoded, and pre-processed(using analysis of variance(ANOVA) to establish the categories).

- **Model Selection**

In this, we are using Stepwise Linear Regression using forward regression to determine the features which will be helpful in effort prediction.
The R^2 adjusted coefficient allowed direct comparison of the models and supported the choice of the best model.
Then, we will find a set of independent variables that minimize the F-value.

The author used a ANN model, which consists of one hidden layer with 23 units in it, with a ReLU activation function.

- **Evaluation metric**

The evaluation metric, which was used in both of the model comparisons, is MMRE(Mean Magnitude of Relative Error) and R2.

- **MMRE and R^2 on some models**

Predictive Model	***MMRE***	***R^2***
Function Point Analysis	103	0.58
SLIM	772	0.89
COCOMO Basic	610	0.70

Key Learning

The results show that the predictions from linear regression and ANN have a strong relationship with the actual development effort values for all the projects.

On the R2 metrics, the performance of ANN was better than the SLIM(0.89), Function Point Analysis(0.58), and COCOMO models(0.70).

Datasets

The dataset used in this was COCOMO Dataset, which is a set of 63 projects, which has 17 feature columns, out of which 15 are categorical variables, and two are continuous variables.

Of which Lines of code(LOC) is an independent variable and Effort is the actual target.

All the categorical variables are encoded using an adjustment factor that is provided in the paper.

Other related work

We all know that Simple equations/models are easy and are more interpretable compared to complex models.

I searched for any simple models that do a similar task, rather than using these complex techniques for solving.

I Found out that the COCOMO model was one of the widely used techniques which are used in effort estimation.

There are three types of COCOMO model

- Basic Model
- Intermediate Model
- Detailed Model

The basic model only takes LOC(Lines of Code) into account and predicts the effort, which is a rough estimate of the original effort.

It is calculated using the formula $a*(KLOC)^b$ (where a, b are pre-defined contants)

The Intermediate model is an extension of the basic model, which also takes 15 other factors into consideration.

The detailed model applies this COCOMO model to each and every development phase.

- **Disadvantages of COCOMO Model**

1. COCOMO model ignores requirements and all documentation.
2. It ignores customer skills, cooperation, knowledge and other parameters.
3. It oversimplifies the impact of safety/security aspects.
4. It ignores hardware issues
5. It ignores personnel turnover levels

- **SLIM**

  SLIM model is a slight improvement of the COCOMO model which is calculated using the formula E = (LOC / (C * t *4/3)) * 3.

  Where E is effort, t is development and C is technology constant which is determined from tools, language, methods, quality features etc.

- **Disadvantages of SLIM**

1. Works best on large projects.
2. The software size needs ot be estimated in advance in order to use this model.
3. Estimates are extremely sensitive to the technology factor.
4. Model is also sensitive to size estimate.
5. Tool is considered to be fairly complex.
6. It works for Waterfall life cycle that doesn’t cover up spiral model.

____

I have also written some programs using Multiple Linear Regression and Neural Networks on COCOMO Dataset which can be found at this link: https://github.com/AnuragAnalog/ML-for-software-project-management

Future Work

We saw that linear regression is sensitive to outliers; ANN's are prone to overfitting.

We can further extend this prediction process by combining the linear regression and ANN's on other datasets, such as the ISBSG database.

Also, try some other machine learning techniques like RandomForest, Gradient Boosting, Linear Discriminant Analysis for the prediction process.

P. Anurag
17MCME13
Integrated M.Tech