Functional Link Artificial Neural Networks for Software Cost Estimation

B. Tirimula Rao, Anil Neeruknoda Institute of Technology and Sciences, India
Satchidananda Dehuri, Fakir Mohan University, India
Rajib Mall, Indian Institute of Technology Kharagpur, India

ABSTRACT

Software cost estimation is the process of predicting the effort required to develop a software system. Software development projects often overrun their planned effort as defined at preliminary design review. Software cost estimation is important for budgeting, risk analysis, project planning, and software improvement analysis. In this paper, the authors propose a faster functional link artificial neural network (FLANN) based software cost estimation. By means of preprocessing, i.e., optimal reduced datasets (ORD), the authors make the functional link artificial neural network faster. Optimal reduced datasets, which reduce the whole project base into small subsets that consist of only representative projects. The representative projects are given as input to FLANN and tested on eight state-of-the-art polynomial expansions. The proposed methods are validated on five real time datasets. This approach yields accurate results vis-à-vis conventional FLANN, support vector machine regression (SVR), radial basis function (RBF), classification, and regression trees (CART).

Keywords: Classification and Regression Trees (CART), Functional Link Artificial Neural Network (FLANN), Radial Basis Function (RBF), Software Cost Estimation, Support Vector Machine Regression (SVR)

1. INTRODUCTION

Software engineering measurement and analysis specifically, cost estimation initiatives have been in the center of attraction for many firms. The concept of software cost estimation has been growing rapidly due to practicality and demand for it. Software cost estimation involves the process to foresee the total costs spent during the development of a software product based on several factors, called ‘cost drivers’, and mostly relate with the product to be developed, the engineering process followed and the people engaged in the process. During, last few decades the main cost driver attracting the research interest is development effort (typically measured in person-months). Software cost estimation techniques fall into following six categories: parametric models including COCOMO (Constructive Cost Model)
(Boehm, 1981; Huang et al., 2007), SLIM (Software Life cycle Management) (Putnam & Myers, 1992), and SEER-SEM (Software Evaluation and Estimation of Resources–Software Estimating Model) (Jensen, 1983); expert judgment including Delphi technique (Helmer, 1966) and work breakdown structure based methods (Tausworthe, 1980; Jørgensen, 2004); learning oriented techniques including machine learning methods (Heiat, 2002; Shin & Goel, 2000; Oliveira, 2006) and analogy based estimation (Shepperd & Schofield, 1997; Auer et al., 2006; Huang & Chiu, 2006); regression based methods including ordinary least square regression (Mendes et al., 2005; Costagliola et al., 2005) and robust regression (Miyazaki et al., 1994); dynamics based models (Madachy, 1994); composite methods (Chulani et al., 1999; MacDonell & Shepperd, 2003).

Functional Link Artificial Neural Network (FLANN) based software cost estimation (FBE), which is essentially a machine learning technique, was introduced by Rao et al. (2009). Due to its conceptual simplicity and empirical competitiveness, FLANN has been extensively studied and applied (Jodpimai et al., 2010; Papatheocharous et al., 2010; Abhishek et al., 2010; Khatibi et al., 2011). FLANN is basically a flat net and the need of the hidden layer is removed and hence, the BP learning algorithm used in this network becomes very simple, originally proposed by Pao et al. (1992). The functional expansion effectively increases the dimensionality of the input vector and hence the hyper planes generated by the FLANN provide greater discrimination capability in the input pattern space. FLANN architecture for predicting software development effort is a single-layer feed forward neural network consisting of an input and output layer. FLANN generates output (effort) by expanding the initial inputs (cost drivers) and then processing to the final output layer. Each input neuron corresponds to a component of an input vector. The output layer consists of one output neuron that computes the software development effort as a linear weighted sum of the outputs of the input layer (Rao et al., 2009).

However, some difficulties are still confronted by FLANN based methods. Such as the non-normal characteristics (includes skewness, heteroscedasticity and excessive outliers) of the software engineering datasets (Pickard et al., 2001) and the increasing sizes of the datasets (Shepperd & Kadoda, 2001). The large and non-normal datasets always lead FLANN based methods to low prediction accuracy and high computational expense. In order to minimize these fluctuations and trade-offs while training the network we introduced a new preprocessing step before training the FLANN. The integration of preprocessing steps and model design using FLANN generates a faster FLANN named it as ORD-FLANN. When given an unknown tuple, the algorithm searches the pattern space for the k training tuples that are closest to the unknown tuple. So we get ORD for training and prediction using neural networks. This validation method gives accurate results than other methods. For predicting the output in Person-Months we used ORD-FLANN.

The rest of this paper is organized as follows: Section 2 presents a brief overview of the literature concerning software cost estimation. Overview on conventional FBE is briefly discussed in Section 3. ORD is used to optimize the projects and fetch reduced similar data point is described in Section 4. In Section 5 datasets and experimental design are illustrated. The experimental results are summarized in Section 6. Conclusion and future work is given in Section 7.

2. RELATED WORK

In the field of software effort estimation, the effort required to develop a new software project is estimated by taking the details of the new project into account. The specific project is then compared to a historical data set (i.e., a set of past projects) containing measurements of relevant metrics (e.g., size, language used, and experience of development team) and the associated development effort.
Adaptable QoS Management Communication Architectures for User Perception
[www.igi-global.com/chapter/adaptable-qos-management-communication-architectures/4220?camid=4v1a](www.igi-global.com/chapter/adaptable-qos-management-communication-architectures/4220?camid=4v1a)

[www.igi-global.com/chapter/creating-adaptive-web-sites-using/4223?camid=4v1a](www.igi-global.com/chapter/creating-adaptive-web-sites-using/4223?camid=4v1a)