

Chapter 9

Analysis and Comparison of Neural Network Models for Software Development Effort Estimation

Kamlesh Dutta

Department of CSE, National Institute of Technology, Hamirpur, India

Varun Gupta

Amity School of Engineering and Technology, Amity University, Noida, India

Vachik S. Dave

Department of CSE, National Institute of Technology, Hamirpur, India

ABSTRACT

Prediction of software development is the key task for the effective management of any software industry. The accuracy and reliability of the prediction mechanisms used for the estimation of software development effort is also important. A series of experiments are conducted to gradually progress towards the improved accurate estimation of the software development effort. However, while conducting these experiments, it was found that the size of the training set was not sufficient to train a large and complex artificial neural network (ANN). To overcome the problem of the size of the available training data set, a novel multilayered architecture based on a neural network model is proposed. The accuracy of the proposed multi-layered model is assessed using different criteria, which proves the pre-eminence of the proposed model.

1. INTRODUCTION

An accurate estimation is the key objective of any prediction model. Software development effort estimation is one of the important research domains for software organizations. However, there are several problems associated with making an accurate estimation. Stutzke (1996) a) attributed ambiguity about the meaning of estimation; b) Confliction of project goal; c) Lack of requirement information; d) Inclusion of Reusable code; e) New development procedures and tools; and f) Relation of inputs to output. Software development effort prediction depends on several factors such as software size, people involved in software development, use of technology, software complexity, and many more. Further, these factors are correlated to each other and influence the software development process directly or indirectly. The effort estimate can be derived from the size of software products, if it is estimated reliably. Defining software development lifecycle and development process that is followed to specify, design, develop, and test the software is necessary for the estimation of the software development effort from software size. Developing new software is not just a task of coding the software functionalities. Rather, coding the software is just a single part of the whole software development process and its effort. Designing the software or deliverables, implementation of the prototypes, writing & reviewing documentation, and reviewing & testing the code take up the larger portion of overall project effort. The project effort estimate requires identifying and estimate, and then sum up all the activities, one must perform to build a product of the estimated size.

The main objective of this paper is to develop neural network based software effort estimation model. A series of experiments are conducted to explore the suitable model and reported in the subsequent sections. The paper is organized as follows. In Section 2 we have enlighten the portion related to the neural networks based estimations of software development effort as reported in the literature. Section 3 gives an overview of various criteria's used for assessing the suitable prediction model for software development estimation. Section 4 presents a series of experiments conducted to analyze the performance of various models. Most of the researchers emphasized the benefit or dominance of the ANN model over other traditional models. Here, the experimental studies are conducted to find out the best suitable ANN model for software effort estimation. Based on the analysis, a multilayered ANN model for the software effort estimation task is proposed in section 5. The model makes use of three categories for input parameters. The categories are based on the impact factor of the inputs. The suitability of the proposed model is further tested and verified through a series of simulation. Section 6 summarizes the results and concludes with the future scope of work in improving the accuracy of software development effort estimation.

2. RELATED WORK

Various approaches have been adopted in literature to make an accurate estimation of software effort. In recent years, application of machine learning approaches has been attempted. This has been possible due to the availability of data sets of a large number of completed projects. Among various machine learning techniques, neural networks based models are newly emerging models. Though several researchers (Venkatachalam, 1993; Finnie et al., 1997; Samson et al, 1997; Lee et al.,1998; Heiat, 2002; Ideri et al, 2002; Idri et al, 2004; Tadayon, 2005; Idri et al, 2006; Kanmani et al, 2007; Tronto et al., 2007; Park and Beak, 2008; Tronto et al, 2008; Iwata et al, 2009; Reddy and Raju, 2009; Ajitha et. al, 2010; Kaur et al,

27 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/analysis-and-comparison-of-neural-network-models-for-software-development-effort-estimation/294464

Related Content

Cyber Physical Systems Management

(2015). *Challenges, Opportunities, and Dimensions of Cyber-Physical Systems* (pp. 286-305).

www.irma-international.org/chapter/cyber-physical-systems-management/121261

Steps Towards Fuzz Testing in Agile Test Automation

Pekka Pietikäinen, Atte Kettunen and Juha Röning (2016). *International Journal of Secure Software Engineering* (pp. 38-52).

www.irma-international.org/article/steps-towards-fuzz-testing-in-agile-test-automation/144789

The Distribution of a Management Control System in an Organization

Alfonso A. Reyes (2010). *Emerging Systems Approaches in Information Technologies: Concepts, Theories, and Applications* (pp. 310-328).

www.irma-international.org/chapter/distribution-management-control-system-organization/38187

Multithreaded Programming of Reconfigurable Embedded Systems

Jason Agron, David Andrews, Markus Happe, Enno Lübbers and Marco Platzner (2011). *Reconfigurable Embedded Control Systems: Applications for Flexibility and Agility* (pp. 31-54).

www.irma-international.org/chapter/multithreaded-programming-reconfigurable-embedded-systems/50424

Ubiquitous IoT in the Automotive Domain: Decentralized Adaptation

Laszlo Z. Varga (2018). *Solutions for Cyber-Physical Systems Ubiquity* (pp. 27-51).

www.irma-international.org/chapter/ubiquitous-iot-in-the-automotive-domain/186900