International Journal of Advanced Technology in Engineering and Science
Vol. No.3, Special Issue No. 01, November 2015
www.ijates.com
ijates
ISSN 2348 - 7550

NEURAL NETWORK BASED REGRESSION TESTING

Esha Khanna

Assistant Professor, IT Department, D. A.V Institute of Management, (India)

ABSTRACT

Regression testing re-executes all the test cases in order to ensure that changes made in a new version of software have no adverse affect on the functionality inherited from previous version. Due to limited time and resources, all the test cases cannot be re-executed. A prioritization technique is therefore required, in order to execute only important test cases and thus save time and resources without compromising the quality of software. The work reviews several test case prioritization techniques for regression testing. The paper proposes a prioritization technique that uses neural networks to order the test cases. Neural network is trained to prioritize the test cases for regression test suite. The work paves the way of neural networks in regression testing.

Keywords: Neural Networks, Regression Testing, Test Case Prioritization

I. INTRODUCTION

In order to enhance the quality of the software, testing is required. Software testing is a process of executing software with intent of finding an error [1]. Software testing is an essential phase of software development life cycle (SDLC) which is carried out during all the phases. Software undergoes lot of changes during its life cycle. Developed software is modified according customer's requirement. New functionality is added to the software and old, obsolete functions are removed. Some previous features are also updated. In order to verify that changes made in software have not created trouble for old functions, testing is required. Such testing technique is known as regression testing. Regression testing is selective retesting of a system or components to verify that modifications have not caused unintended effects and that the system or component still complies with its specified requirements [2]. New test cases are added to test the new features and old test cases are re run to verify that modifications have not caused unintended effects. Re-executing all the test cases of regression test suite is not possible, due to limited time and resources. Moreover, all the test cases are not equally important. Test cases are therefore prioritized in order to improve the effectiveness of regression testing. In test case prioritization, test cases are ordered in such a way that high priority test cases are executed first. A lot of research has been carried out in test case prioritization. It has been observed that not much work has been done in prioritization using computational intelligence techniques. The work proposes a new test case prioritization technique that uses neural networks. Neural network is a computational intelligence technique that mimics the human brain and is used for pattern recognition. The work paves the way of neural networks in software testing. The paper has been organized as follows. Section two of the paper reviews test case prioritization, section three discusses neural networks, section four presents the proposed technique and section five concludes.

II. BACKGROUND

2.1 Test Cases

In order to carry out software testing, test cases are created. A test case is defined as a set of test inputs, execution conditions and expected results developed for a particular objective, such as to exercise a particular program path or to verify compliance with a specific requirement [3]. Test cases can be created manually or the process can be automated. Crafting of good test cases is one of the important tasks of testing team. A good test case is the one that has a high probability of revealing maximum number of undiscovered error. A test case includes test case ID, test case name, preconditions, inputs and expected outputs [4]. The test cases are then executed and the actual outputs are compared with the expected outputs. Test cases can be positive or negative [5]. A positive test case determines what the system is supposed to do whereas latter determines what the system is not supposed to do.

2.2 Test Case Prioritization

Test case prioritization is one of the effective methods for efficient regression testing. Any change in the software may affect the previous features and working of software. Therefore, there arises a need to retest the software. Re-executing all previous test cases is a lengthy and exhaustive process. In order to overcome this problem, three techniques are available i.e test suite reduction, test suite selection and test suite prioritization [6]. First two techniques reduce the number of test cases in test suite whereas test case prioritization does not affect the number of test cases in a particular test suite. Test case prioritization assigns a value to each test case which tells the importance of executing that particular test case. High priority test cases are executed before the test cases having low priority. Prioritizing test case in itself is a tedious task. Therefore, this task can be automated.

2.3 Literature Review

An extensive literature review has been carried out to find the gaps in the existing literature. The review is carried out in accordance to the guidelines proposed by Kitchenham [7]. Various databases having high impact factor and good research papers have been searched. Databases like IEEE Explore, ACM Digital, Springer, and Wiley Online have been explored. Several techniques for test case prioritization have been studied. These techniques include coverage based prioritization [8], cost based prioritization [9], History based prioritization [10], Dataflow and control based prioritization [11], requirement based prioritization [12], search based prioritization [13].

III. NEURAL NETWORKS

Neural Network is a computational intelligence technique inspired by biological nervous system. They are information processingA paradigm and are use for pattern recognition. Neural networks are physical cellular systems which can acquire, store and process the experiential knowledge [14]. Like humans, neural networks learn by examples and their past experiences. They are expert in deriving meaning form imprecise data and extracting patterns. They are adaptive to the surroundings. They do not use any algorithmic approach to solve a problem.

Neural network is composed of a large number of massively parallel connected elements called neurons. Fig. 1 shows the basic architecture of neural network model.

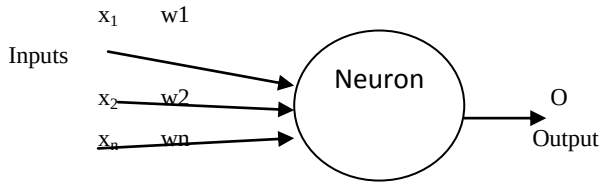

Figure 1. Architecture of a basic Neural Network

Inputs X=(x_1, x_2….x_n) and weights w= (w_1,w_2….w_n) are provided to the neuron and net is calculated. Output of the neuron is obtained by applying activation function to the summated input. The procedure is summarized in following equations.

$$net = \sum XiWi$$

$$output = f(net)$$

Where f() is the activation function applied to total input aggregated. The most commonly used activation functions are hard limiting function, linear transfer function and sigmoidal transfer function [15].

Neural network is classified as single layer or multilayered [16]. Single layer neural network consists on an input layer, a neuron layer and an output layer. Multilayered neural network is formed by interconnection of several neuron layers. Outputs from first neuron layer act as an input to other layer. This layer is known hidden layer. Neural networks is also classified as feedforward and feedback [14]. In feedforward neural networks, outputs of the networks are not looped back to the inputs. Therefore, there is no time delay between input and output mapping. On the other hand, in feedback networks, outputs of the network are looped back to the inputs. Therefore, a time delay between input and output mapping is introduced. Neural network can be implemented using MATLAB software. Procedure to train the neural network model is as follows [15].

- Collect data
- Create the network
- Configure the network
- Initialize the weights and biases
- Train the network
- Validate the network
- Use the network

Neural network has been used in the fields of robotics, medical science, defense, electronics, banking and aerospace [17, 18, 19, 20, 21].Many researchers have been carried out to apply NN in various different fields. The work uses neural networks for prioritizing the test cases of regression testing. The work extends the applicability of neural networks in software testing

IV. PROPOSED WORK

The work proposes a test case prioritizing technique based on neural networks. All the test cases of a regression test suite cannot be executed due to limited time and resources. Moreover not all the test cases are equally important. Therefore, test cases are prioritized in order to execute more important test cases first. Manual prioritization of test cases is itself a lengthy and tedious process. In order to solve this problem, prioritization task is automated. Neural networks can be trained to automate the prioritization of test cases. The work proposed use of neural networks in test case prioritization. The neural network is first fed with test cases and prioritizing rules [5]. Network is allowed to learn the prioritization techniques and rules. Input to the neural networks in learning phase is set of test cases and test case prioritization policies [5]. The trained neural network is then be verified on a set of test cases. During the testing and verification phase, input to trained neural networks is a set of test cases and the output of neural network is prioritized set of test cases. The work has been summarized in Fig. 2 and Fig. 3. 70% of the test cases are used in training phase in order to train the neural network and rest 30% of test cases are used to verify and test the trained neural network.

Step 1: create a regression test case suite for software under test. This includes modifying the test cases, deleting the obsolete test cases and adding new the test cases.

Step 2: Regression suite test cases and test case prioritization policies [5] are used to create training data.

Step 3: Training data obtained from first step is used to train the neural network.

Step 4: Trained neural network is then tested and verified with a set of test cases.

Step 5: Output of neural network is a prioritized set of test cases.

Figure 2: Algorithm

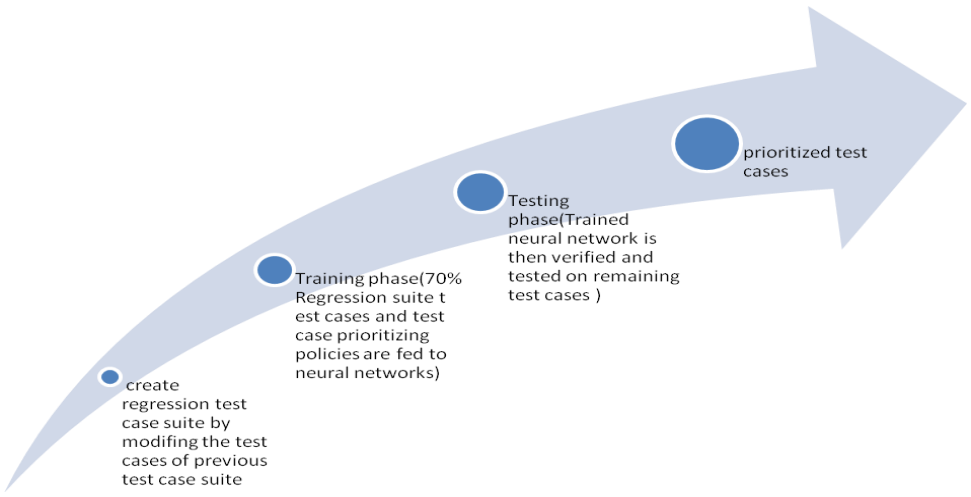

Figure 3: Neural Networks Based Regression Testing

V. CONCLUSION

The work proposes a neural network based test case prioritization technique for regression testing. Neural network is a powerful computational intelligence technique, which is primarily used for pattern recognition. The work proposes a framework which prioritizes the regression test cases on the basis of this technique. The work paves the way of neural networks in regression testing.

REFERENCES

[1] J. Myers, The Art of Software Testing, second edition, John Wiley and Sons, 2004.

[2] IEEE Standard definition of regression testing [online].

[3] IEEE Standard 610 (1990) definition of test cases [online].

[4] ERP Testing "Test Case Structure" [online] available at http://erp-testing.blogspot.in/2012/04/test-case structure.html.

[5] H. Bhasin, E. Khanna, Neural Network based Black Box Testing, ACM Sigsoft Software Engineering Notes, 40, 2014 .

[6] Yoo S., Harman M., Regression testing minimization, selection and prioritization: a survey, Software Testing, Verification & Reliability, John Wiley and Sons Ltd ,22(2), 2012, 67-120.

[7] Kitchenham, B.A et. al. Systematic literature reviews in software engineering .A tertiary study, Information & Software Technology .INFSOF , 52(8), 2010,792-805

[8] Leon D., Podgurski A., A Comparison of Coverage-Based and Distribution-Based Techniques for Filtering and Prioritizing Test Cases, Proceedings of the 14th International Symposium on Software Reliability Engineering, IEEE, 2003, 442.

[9] Hema Srikanth, Myra B. Cohen, and Xiao Qu., Reducing Field Failures in System Configurable Software: Cost-based Prioritization, In Proceedings of the 20th IEEE International Conference on Software Reliability Engineering , IEEE , Piscataway, NJ, USA, 2009, 61–70.

[10] Jung-Min Kim, Adam Porter, A history-based test prioritization technique for regression testing in resource constrained environments, Proceedings of the 24th International Conference on Software Engineering, 2002,119-129

[11] Rummel J. M. et. al., Towards the prioritization of regression test suites with data flow information, Proceedings of the ACM symposium on Applied computing, 2005,. 1499-1504.

[12] R. Kavitha, V.R. Kavitha, N. Suresh Kumar, Requirement Based Test Case Prioritization. In Proceedings of International Conference on Communication Control and Computing Technologies, 2010, 826–829.

[13] Li. et. al., Search Algorithms for Regression Test Case Prioritization, Software Engineering, IEEE Transactions, 33(4), 2007, 225 – 237.

[14] J.M Zurada, Introduction to Artificial Neural Systems, Jaico Publishing House.

[15] Demuth, H., Beale M., Neural Network Toolguide, user guide, version 4, The Mathworks

[16] S.N Shivanandam, S.N Deepa, Principles of Soft Computing, Wiley-India.

[17] S. Y King, J. N. Hwang, Neural network architectures for robotic applications, Robotics and Automations, IEEE, 1989.

[18] M. B Gorzalczany, An idea of the application of fuzzy neural networks to medical decision support systems, IEEE, 1996.

[19] J. C Lupo, Defense applications og neural networks, IEEE, 27(11), 1989.

[20] S. F Zhao, L. C Chen, The BP neural networks applications in bank credit risk management system, IEEE, 8th international conference, 2009.

[21] S. Salahova, Remote Sensing and GIS Application for Earth Observation on the base of Neural Networks in Aerospace Image Classification, Recent Advances in Space Technologies (RAST), 3rd international conference, 2007.