A Research on Error-Back Propagation Algorithm for Regression Software Testing

Samuel Vanramthara
Amity University, Rajasthan
Department of Computer Science & Engineering

Abhay Sharma
Amity University, Rajasthan
Department of Computer Science & Engineering

Abstract: This research is to provide a survey on using Artificial Intelligence method, the Error-Back Propagation algorithm for Regression Software Testing technique. The paper conveys the functionality of the Error-Back Propagation algorithm and Regression Software testing. The purpose of this paper is to research on making an optimized testing technique using AI.

Keywords: Artificial Intelligence, Software Testing, Error-Back Propagation Algorithm and Regression Software testing.

---

1 INTRODUCTION

Artificial Intelligence techniques such as knowledge based systems, neural networks, fuzzy logics and data mining have been advocated by many researchers as the way to improve many of the software development activities. As with many other disciplines, software development quality improves with the experience, knowledge of the developers, past project and expertise. Hence, there is significant potential for using AI in all phases of SDLC.

Software testing is used for improving software reliability by finding errors and failures in software products. Error detection was performed by various testing process models and techniques. These include many testing technique like unit testing, acceptance testing, load testing regression testing and methods such as Black Box and White Box testing. As one of the key methods for software quality assurance, software testing can be very labor intensive if not automated.

2 ARTIFICIAL INTELLIGENCE

AI is the intelligence of machines and branch of computer science that aims to create it. AI textbooks define the field as "the study and design of intelligent agents" where an intelligent agent is a system that perceives its environment and takes action that maximizes its chances of success. John McCarthy, who coined the term in 1995, defines it as "the science and engineering of making intelligent machines".

AI is accomplished by studying how human brain thinks and how human learns, decide and work while trying to solve a problem.

2.1 Artificial Neural Network

An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signal received is strong enough, the neuron is activated and emits a signal through the axon. In other words, it is comprised of a layered set of interconnected processors. Processor goes by another name called neurodes instead of neurons.

ANN can take on values only in the range of 0 and 1. If these are values outside the range, the data have to be normalized. If there are M possible outcomes, the number of binary nodes N required to design the ANN should satisfy the condition,

$$M <= 2^N$$

2.1.1 Back Propagation Algorithm

Back propagation is a systematic method for training multi-layer artificial neural networks. It has a mathematical foundation that is strong if not highly practical. It is a multi-layer forward network using extend gradient –descent based learning rule, commonly known as back propagation (of errors) rule. The input is introduced at the nodes in the input layer and then propagated through the network in one propagation without any loop to the output layer. As inputs are fed at the input nodes, they are modulated by the connection weights before reaching the next layer. For a fully connected BPN, a node in any layer is connected to all the nodes of the proceeding layer but not connected with any node in the same layer.

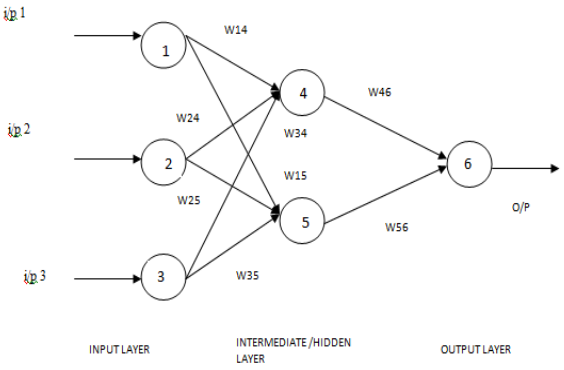

Fig.1 Back-Propagation Network

2.1.2 Training Algorithm of BPN

The network receives inputs by neurons in the input layer, and the output of the network is given by the neurons on an output layer. There may be one or more intermediate hidden layers. The BPN algorithm uses supervised learning, which means that we provide the algorithm with examples of the inputs and outputs we want the network to compute, and then the error (difference between actual and expected results) is calculated. The idea of the algorithm is to reduce the error until the ANN learns the training data. The training begins with random weights and the goal is to adjust them so that the error will be minimal.

The activation function of the artificial neurons in ANN implementing the back propagation algorithm is a weighted sum (the sum of the input x_i multiplied by their respective weights w_{ji}).

$$A_j(\overline{x},\overline{w}) = \sum_{i=0}^{n} x_i \; w_{ji}$$

Each node combines all the incoming values transmits that sum either unaltered to the next layer or alters that sum by an activation function before transmitting. The activation function can be a simple threshold values i.e., it transmit 0 if the combined values is less than the threshold and transmits 1 if the combined values is more than the threshold.

$$Sigmoid\ function\ f(x) = \frac{1}{1+e^{-kx}}$$

3 SOFTWARE TESTING

Software testing is the process of validation and verification of the software product.
According to Boehm:
Verification- "Are we building the product right?".
Validation-"Are we building the right product?".

Effective software testing will contribute to the delivery of reliable and quality oriented software product, more satisfied users, and lower maintenance cost and more accurate and reliable results. The importance of testing can be understood by the fact that around 35% of the elapsed time and over 50% of the total cost are expanding in testing programs.

To optimize the software testing quality and reliability, Error-Back Propagation algorithm for Regression Software Testing is introduced in this paper.

3.1 The Regression Software Testing

Regression Testing is a software testing technique to confirm that a recent program or code change has not adversely affected existing features. In other words, retesting the system's functionality after the code has changed from previous cycle of testing.

It is needed to ensure that the changes have not altered the functionality of untouched part of the system.

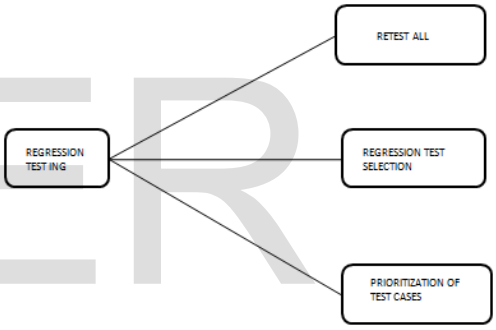

Fig.2 Regression Software Testing

3.1.1 Retest All

This is one of the methods for regression testing in which all the tests in the existing test bucket or suite should be re-executed.

3.1.2 Regression Test Selection

Instead of re-executing the entire test suite, it is better to select part of test suite to be run.

3.1.3 Prioritization of Test Cases

Selection of test cases based on priority will greatly reduce the regression test suite.

4 BUILDING PROPAGATION MODEL THROUGH REGRESSION

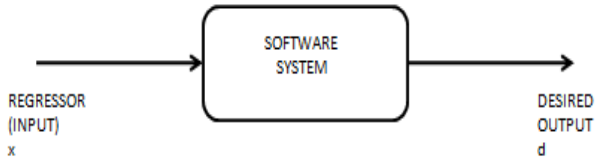

Fig. 3(a)

In Fig. 3(a), the software system is the focus of attention. The system is probed by applying a set of inputs, constituting the regressor.

$$x = [x_1, x_2, \ldots, x_m]^T \quad \text{..... Eq(1)}$$

where T denotes matrix transposition. The resulting output of the system, denoted by d whose functional dependence is unknown, so we propose a linear regression model, parameterized as:

$$d = \sum_{j=1}^{m} w_j \, x_j + € \quad \text{..... Eq(2)}$$

where $w_1, w_2, \ldots w_m$ denotes a set of fixed but unknown parameters. The additive term €, representing the mutation (the error) of the software. We may re-write Eq(2) as;

$$d = w^T x + € \quad \text{..... Eq(3)}$$

correspondingly, parameter vector w is defined by,

$$w = [w_1, w_2, \ldots, w_m]^T \quad \text{..... Eq(4)}$$

The matrix term $w^T x$ is the inner product of the vectors w and x

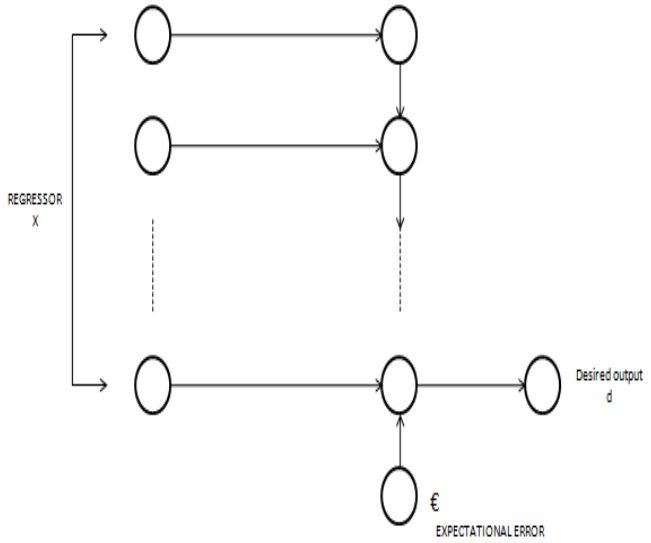

Fig. 3(b)

The proposed model can be used not only for testing the correctness of changed portion of a software program, but also for tracking the quality of its output without re-testing the entire software.

5 POINTS TO BE NOTED

If your software undergoes frequent changes, regression testing costs will escalate. In such cases, manual execution of test cases increases test execution time as well as costs. Automation of regression test cases is the smart choice in such cases.

6 CONCLUSIONS

The research of this paper started with a formal definition of AI and explains the idea behind Error-Back Propagation artificial neural network. Then, we move on to Software testing, defining it and then explain the idea behind Regression Software Testing technique. A model using Back-Propagation Algorithm was proposed and we were able to conclude mathematically that Error-Back Propagation artificial neural network algorithm could be use to optimize the regression software testing.

7 REFERENCES

[1] Simon Haykin, "Neural Networks and Learning Machines", Third Edition, 2011.

[2] S N Sivanandam, "Introduction to Neural Networks using MATLAB 6.0", 2006.

[3] Stuart Russell, "Artificial Intelligence A Modern Approach", Third Edition, 2010.

[4] S Dick, "Computational Intelligence in Software Quality Assurance".

[5] Redge Bartholomew, "Using Combinatorial Testing to Reduce Software Rework", Legacy System Software Sustainment, CrossTalk-Jan/Feb 2014.

[6] Milad Hanna, Nahla El-Haggar and Mostafa Sami, "A review of Scripting Techniques Used in Automated Software Testing", IJACSA, Vol.5,No.1,2014

[7] Rene Just, Darious Jalali, "Are mutants a valid substitute for real faults in software testing".

[8] Farid Meziane and Sunil Vadera, "Artificial Intelligence for improved software engineering development", Information Science References, 2010.