An Adaptive Neural Network Based Technique to Enhance Software System Assurance For Regression Testing

Zhiwei Xu
Motorola Labs, Motorola Inc.
Schaumburg, IL USA
azx095@email.mot.com

Noel D. Samaan
Motorola Labs, Motorola Inc.
Schaumburg, IL USA
cnc030@email.mot.com

Abstract

In this study, we propose a new learning algorithm, Adaptive Propagation (APROP*), for multilayer feedforward neural networks, which is more stable, converges faster and achieves a better prediction accuracy than the conventional backpropagation learning algorithms.* APROP *accelerates and stabilizes the training process in a way that not only dynamically change learning rate according to the differences between the computed and the desired outputs, but also consider the behavior of the partial derivative of weights. We used* APROP *in the test cases selection for regression.*

The APROP *neural network will resemble a human judgement via correlating knowledge from customer profile, analysis of past test results, system failure rate analysis, and dynamic behavior of executable software object to identify potentially critical test cases for regression testing. We tested* APROP *neural network model with a sample dataset for a large telecommunications systems and compared the result with pure gradient-descent, and adaptive learning rate. The comparison results indicate that* APROP *neural network, after enough training, can better resemble a human judgement in identifying the critical test cases than pure gradient-descent, and adaptive learning rate.*
Keywords*: Neural network, learning algorithm, backpropagation, feedforward networks*

1. Introduction

In this study, we propose a new learning algorithm, Adaptive Propagation (APROP), for multilayer feedforward neural networks, which is more stable, converges faster and achieves a better prediction accuracy than the conventional backpropagation learning algorithms. There are various conventional backpropagation learning algorithms for neural network [2, 4, 7, 9, 10]. However, the training speed, memory usage, potential diverge training problems, and local minimum, are inherent disadvantages of these algorithms. To overcome the inherent disadvantages, We designed APROP based on the resilient propagation [10], adaptive learning rate [9], and momentum [9]. We used APROP in the test cases selection for regression.

The existing regression test selection techniques are based on information about the code of the program and the modified version [1, 3, 6, 8]. However, in some institutions, the software developers and system testers are scattered, and due to the security reasons, the information on the source codes can't be provided to the system testers. The unavailability of source code at system (black-box) level testing is true for any organization that implements known test practice recommended by independent verification and validation processes. Under this circumstance, Samaan [11] reported on the need to correlate the user profile together with the change in architecture and analysis of past test results using simple logic in an attempt to optimize test cases selection for regression. Samaan also states the limitation of using simple logic particularly when the number of nominated test cases for regression is far more than what can be accepted for industry standard as cost effective in order to markedly reduce cycle time. Samaan suggested the use of fuzzy logic or neural network (or combination of) to address the said limitations.

The APROP neural network will resemble a human judgement via correlating knowledge that represents one or more of the following: customer profile, analysis of past test results, system failure rate [13], and dynamic behavior of executable software object [12], to identify potentially critical test cases for regression testing. We tested APROP neural network model with a sample dataset for a large telecommunications systems. We also applied other well-known learning algorithms, such as pure gradient-descent, and adaptive

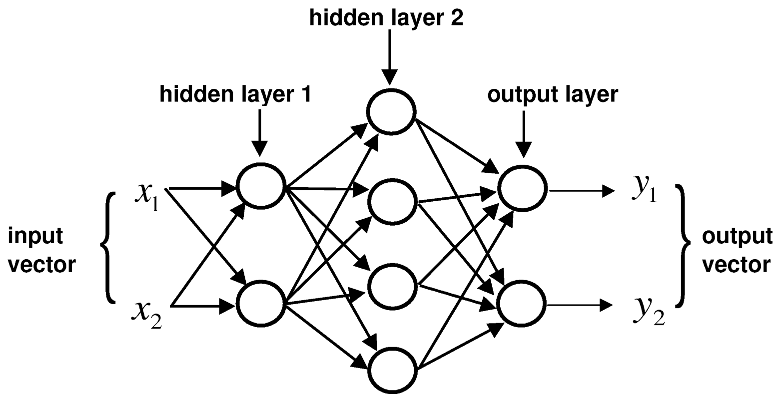

Figure 1. A feedforward neural network

learning rate, to the same application. The comparison results indicate that APROP neural network, after enough training, can better resemble a human judgement in identifying the critical test cases than pure gradient-descent, and adaptive learning rate.

2. Adaptive Resilient Propagation Network

The multilayer neural networks have been very popular for various applicattion since the error-backpropagation algorithm was publicized in 1986 [5].

A typical neural network has the input layer, a numbers of hidden layers, and the output layer. Figure 1 shows a feedforward multilayer neural network. The traditional backpropagation algorithm adjusts the weights and basis by application of gradient to compute the influence of each weight in the network with respect to a cost function E.

$$\frac{\partial E}{\partial w_{ij}} = -\sum_{k=1}^{p}(d_i^{(k)} - o_i^{(k)})f'(net_i^{(k)})x_j^{(k)} \qquad (1)$$

where $f'(\cdot)$ means the derivative of function $f(\cdot)$. Recall that $net_i^{(k)} = f(\sum_{j=1}^{m} w_{ij}x_{ij}^{(k)})$ is the net input to the activation function of the ith neuron when the kth input pattern is presented. Since minimizing the output error requires that the weight changes be in the negative gradient direction, we have:

$$\nabla w_{ij} = -r\frac{\partial E}{\partial w_{ij}} = r\sum_{k=1}^{p}(d_i^{(k)} - o_i^{(k)})f'(net_i^{(k)})x_j^{(k)} \quad (2)$$

However, it becomes clear that training multilayer networks is usually a time-consuming process, and is not easy to achieve the global optimal solution. Many algorithms have been proposed to speed up the learning process and achieve better solutions. APROP differs from traditional steepest gradient backpropagation in a way that not only adapt learning rate but the partial derivative. The advantage of adaptive is to make the learning process faster and more stable.

In summary, the APROP learning algorithm can be expressed as following.

$$\nabla w_{ij} = -f_1(r, E)f_2(\frac{\partial E}{\partial w_{ij}}) \qquad (3)$$

where f_1 and f_2 are two adaptive functions.

3. Case Study

We studied a sample data from a GSM system test data base. The sample data contains 69 test cases. We extract seven matrix from the database as the input to the neural network. The desired output is the score of priority of the test case. The seven inputs comes from past test case analysis, customer profiles, and reliability analysis.

The seven input matrix was normalized to a scale of [0 10], and the desired output was a numerical number of [0 1]. Our neural network model is based on this normalized data. We randomly partitioned this data into two subset, two thirds of the observations for a fit data set, the remainng third for a test data set.

We built a APROP neural network and other three neural networks using other learning techniques. They are traditional steepest descent training algorithm(ASDA), adaptive learning rate algorithm(ALRA). Table 1 specifies the structure of the neural networks used in our case study.

We evaluated the accuracy of the model using the test data set. The simulation result showed that APROP

Table 1. Summary of Neural Networks for CCCS

Architecture	
Class	Perceptron
Connections	Feedforward
Layers	3
Input nodes	7
Hidden nodes	10
Output nodes	1
Node details	
Activation function	Unipolar Sigmoid
Temperature (λ)	1
Training	
Mode	Supervised
Algorithm	Backpropagation
Weight updates	Each epoch
Learning rate	0.01
Momentum rate	0.95

increased the accuracy 65% than other two algorithms, and the training process is more stable and converges faster.

4. Conclusions

In this paper, we report a new learning algorithm, APROP. We also demonstrate, through analysis and simulation, how APROP can help better identify critical test case in regression test.

References

[1] H. Agrawal, J. R. Horgan, E. W. Krauser, and S. A. London. Incremental regression testing. In D. Card, editor, *Proceedings of the Conference on Software Maintenance*, pages 348–357, Washington, Sept. 1993. IEEE Computer Society Press.

[2] R. Battiti. First and second order methods for learning: Between steepest descent and Newton's method. *Neural Computation*, 4(2):141–166, 1992.

[3] Y.-F. Chen, D. S. Rosenblum, and K.-P. Vo. TestTube: A system for selective regression testing. In B. Fadini, editor, *Proceedings of the 16th International Conference on Software Engineering*, pages 211–222, Sorrento, Italy, May 1994. IEEE Computer Society Press.

[4] M. H. Fun and M. T. Hagan. Levenberg-Marquardt training for modular networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pages 468–473, 1996.

[5] R. Hecht-Nielson. Applications of counterpropagation network. *Neural Networks*, 1:131–139, 1988.

[6] B. Korel and A. M. Al-Yami. Automated regression test generation. In *Proceedings of the ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA-98)*, volume 23,2 of *ACM Software Engineering Notes*, pages 143–152, New York, Mar. 2–5 1998. ACM Press.

[7] Y. LeCun. A learning procedure for asymmetric network. *Cognitiva*, 85:599–604, 1985.

[8] N. Mansour, R. Bahsoon, and G. Baradhi. Empirical comparison of regression test selection algorithms. *The Journal of Systems and Software*, 57(1):79–90, Apr. 2001.

[9] J. C. Principe and N. R. Euliano. *Neural and Adaptive Systems: Fundamentals through Simulations.* John Wiley & Sons, INC., New York, 1999.

[10] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591, 1993.

[11] N. Samaan. Optimization of test case selection for regression. In *Proceedings of International Symposium on Software Reliability Engineering*, Hong Kong, Dec. 2001.

[12] N. Samaan and A. Ibrahim. Characterization of dynamic behavior of executable software objects to optimize test case selection for regression. *Sixth IEEE International Symposium on High Assurance Systems Engineering*, Oct. 2001.

[13] Z. Tan and N. Samaan. Software reliability analysis for regression testing. *Sixth IEEE International Symposium on High Assurance Systems Engineering*, Oct. 2001.