2012 3rd International Conference on Information Security and Artificial Intelligence (ISAI 2012)
IPCSIT vol. 56 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V56.28

Mathematical Modeling of Hidden Layer Architecture in Artificial Neural Networks

N. M.Wagarachchi [1+] and A.S. Karunananda [2]

[1, 2] Faculty of Information Technology, University of Moratuwa, Sri Lanka

Abstract. The performance of a multilayer artificial neural network is very much depends on the architecture of the hidden layers. Therefore, modeling of hidden layer architecture has become a research challenge. At present most of the models of hidden layer architecture have been confined to neural networks with one hidden layer. However, this approach may not be the most appropriate solution for the given task. In this research we have come up with an approach to model hidden layer architecture with arbitrary number of layers and neurons. An approach has been presented to trim the hidden layer architecture during the training cycle while meets the pre-defined error rate. The experiments show that new theory can train artificial neural networks with lesser training time through a simpler architecture that maintains the same error rate as the Back propagation.

Keywords: Artificial neural networks, Backpropagation training, Hidden layer modelling.

1. Introduction

Multilayer artificial neural networks have been able to produce solutions for many real world problems with very high generalization power. Therefore they have been widely applied in several fields. The number of layers in a multi layer artificial neural network has been crucial for their generalization power. In general, artificial neural networks with large number of hidden layers would be produced more generalized solution. However, this approach may not be computationally optimal. As such, a large number of researchers carried out to model the hidden layer architecture.

For example, [3], [4] have been suggested pruning methods to minimize the network. [1] proposed a technique to eliminate nodes by introducing additional cost function on a set of auxiliary linear response units. Also, [5] and [6] have been optimized hidden layer architecture by using the mean square error. Further, in [8] - [10] some constructive methods have been proposed.

However, these approaches have various limitations especially, when the network comprises with a large number of layers and neurons. Hence, there is no exact solution for the problem of hidden layer architecture in multi layer artificial neural networks and still it remains as an unsolved problem.

We have researched into mathematical modeling of hidden layers and decided to model hidden layer architecture based on backpropagation training [12] for multilayer artificial neural networks. Accordingly, we selected to model the hidden layers thorough the delta (δ) values of hidden layer neurons. Our insight was based on the fact that δ values of the layer immediate before the output layer are computed using the error involved in output layer. Hence, delta values can be used in minimizing the said error over the training cycles. Generally, the summation of such delta values shows negative correlation with the error of the training cycle. Therefore, we postulated that identification of hidden layer neurons with large negative values of δ would be significant for modeling hidden layer architecture. Effectively, it may be able to remove such neurons to prune the hidden layers. Experiments have been conducted to identify the best way to decide on

+ Corresponding author.
E-mail address: mihirini@is.ruh.ac.lk

the most appropriate neuron to be removed. As a result, we were able to set a threshold value for δ of the hidden layers and remove neurons which are less than the defined threshold value.

The different approaches to hidden layer architecture presented by other researchers are given in next section. Section 3 gives the theory for new approach while the new algorithm presents in section 4. Research methodology and results are present in sections 5 and 6 respectively. Finally, the conclusion is given by sections 7.

2. Different Approaches to Hidden Layer Architecture

2.1. Pruning Algorithms

The objective of pruning algorithms is to obtain the optimum structure of network by incrementally removing low-saliency neurons from the architecture. Optimal Brain Damage (OBD, presented by Le Cun and Jhon Denker [13] is a commonly use pruning algorithm, where parameters with low-saliency are detected based on the second derivative and removed from the network. The intension of OBD is that, it is possible to obtain network which perform in same manner (or better), by eliminating about half of the weights. However, a major issue arises with the enormous size of the Hessian matrix. This causes computational barriers and also takes considerable time to train. Hence, it assumes the Hessian matrix is diagonal.

In contrast to Le Cun et al. Hassabi [14] introduced Optimal Brain Surgeon(OBS) and claimed that Hessian matrix is non-diagonal and for every problem and hence, weights elimination of OBD may not accurate. Even though, it argues that OBS is better than OBD and yields better generalization, much computational limitations arises especially working with the inverse of the Hessian matrix.

Giovanna et al. [3] proposed a conjugate gradient algorithm in least-square sense to prune neurons after trained the network. It reached the optimal size by removing neurons successfully from a large trained network and then adjusting the remaining weights to maintain original input-output behaviour. They claim that this algorithm yields better results relative to other existing methods. However, in this research it is not considered the multi-layered architecture and hence, it deviates from our aim. On the other hand, our goal is to identify and prune the neurons before train the network. Finally, we concentrate on removing a cluster of neurons rather than removing one neuron at a time.

Faisal et al [4] have been used hybrid sensitive analysis and adaptive particle swarm optimization (SA-APSO) algorithm to determine the minimal architecture of artificial neural networks. First, defined an impact factor and remove the nodes whose impact factor is less than some pre-defined threshold value. Then similar neurons can be merged. Anyhow, this research is restricted only for two layered network. But there is no such rule that two layer network can solve almost all the problems. Also there are some problems which require three layer networks [15]

2.2. Constructive Algorithms

In constructive neural networks the network structure is built during the training process by adding hidden layers, nodes and connections to a minimal neural network architecture. However, determining whether a weight should be added and it adds to the existing hidden layer of new layer is not straightforward. Therefore in most algorithms, pre-defined and fixed number of nodes are added in the first hidden layer and the same number of nodes are added to second layer and so on [16]. This number is crucial for the better performance of the network and it makes as small as possible to avoid the complex computations during the training.

The dynamic node creation (DNS) [17] algorithm is supposed to be the first constructive algorithm for designing single layer feed-forward networks dynamically. Also the cascade correlation algorithm (CCA), firstly proposed by S. E. Falhman [9] is a well known and mostly used constructive algorithm. This algorithm has proposed as a solution of problems such as local minima problem. Moreover, Sridar et al [10] improved the adaptive learning algorithm for multi-category tiling constructive neural networks for pattern classification problems

2.3. Evolutionary Methods

Other than the pruning and constructive methods some researchers have been suggested evolutionary methods to model the hidden layer architecture based on various mathematical concepts. For instance, an alternative method proposed by Stathakis [5] to obtain the near-optimal solution by searching with a genetic algorithm. This research was carried out to determine the optimum architecture and it claims that obtained network is much efficient in problem classification. Although it provides very small error in the training cycle, topology of network is comparatively large. Hence, it is difficult to apply especially when there is large number of inputs in the network.

Shuxiang et al . [6] have been approached to the best number of hidden units based on rooted mean square error. Although they have applied this to medium-sized data set and there is no guarantee of applying the theory for large number of hidden neurons.

Even though methods to optimize the hidden layer architecture have been developed, they have not fully addressed the problem regarding the topology of multilayer artificial neural networks such that the determining number of required layers for better performances and then minimize the error in the training cycle.

3. Theory for New Approach to Hidden Layer Modeling

An algorithm has been proposed to optimize the multilayered artificial neural network. It prunes neurons as much as possible from the hidden layers while maintaining the same error rate as the back propagation given by

$$E = \frac{1}{2N}\sum_{i=1}^{N}\sum_{j=1}^{m}\left(t_{ij} - o_{ij}\right)^2 \tag{1}$$

Where N is the number of training patterns and m is the number of neurons in the output layer. t_{ij} and o_{ij} represent the desired and actual outputs of the j^{th} neuron in the i^{th} input pattern respectively. Pruning is done by using the delta values of hidden layers. The delta value of the j^{th} neuron of the hidden layer immediate before the output layer is given by

$$\delta_j = f'(net_j)\sum_k \delta_k w_{kj}$$

where f is the pre-defined activation function of the hidden layer, w_{kj} is the weight of the connection from neuron j to neuron k. δ_k is delta value of k^{th} neuron in the output layer define by

$$\delta_k = f'(net_k)(t_k - o_k) \tag{3}$$

Where f is the activation function defined for output layer and t_k and o_k are the desired and actual outputs of the k^{th} neuron respectively. The intension of choosing delta value is that the delta values of hidden layers are calculated using the error of the training cycle. Hence, there is a strong relation with the delta values and the said error. Therefore these delta values can be used to minimize the error of the training cycle. However, empirical results show that the summation of δ values of hidden layer and the error of the training cycle are negatively correlated.

As such, the error in (1) can be minimized by maximizing the summation of δ in equation (2). Therefore, optimized architecture can be obtained by removing the delta values of hidden neurons which have large negative values. On the other words, error can be reduced by removing the neurons, which are negative and much less than the mean of δ values of the particular layer. On the other words, when $\overline{\delta^h}$ and σ_{δ_h} are the mean and the standard deviation of δ values of h^{th} hidden layer respectively, it may possible to choose a real positive α such that removing neurons which are less than $\overline{\delta^h} - \alpha.\sigma_{\delta_h}$ minimizes the error in training cycle.

After removing all possible neurons in the last hidden layer, remaining are used to calculate the delta values of the previous layer. Apply the same technique to trim all the layers as much as possible. Finally, update the weights according to $w_{new} = w_{old} + \eta.\delta f(net)$, where η is the learning rate and f is the activation function. By repeating the procedure until the network is stable, optimum architecture can be obtained.

4. New Algorithm for Hidden Layer Modeling

As we have discussed in the previous sections main aim of this paper is to propose a method to trim the network by maintaining error rate as the backpropagation training given in the equation (1). In order to find the solution a new algorithm is used. According to this algorithm train the network once for the first input and observed the error between desired and actual errors. This error is used to determine the delta values of the output layer as follows. $\delta_j = E_j f'(net_j)$,where E_j is the error of the j^{th} node in the output layer, defined by the equation (1). Calculate the delta of second hidden layer by

$$\delta_i^2 = f_2'(net_i) \sum_{j=1}^{m} w_{ji}\, \delta_j$$

and remove all the neurons with $\delta_i^2 < \overline{\delta^2} - \alpha.\sigma_2$ from the second hidden layer. Let the number of remaining nodes in second hidden layer be n_2', δ values of first hidden layers are computed according to

$$\delta_i^1 = f_1'(net_i) \sum_{j=1}^{n_2'} w_{ji}\, \delta_j.$$

After removing all possible nodes weights are updated. Apply another input and continue the procedure until the network becomes stable. The complete procedure is given by the following algorithm.

4.1. The New Algorithm

Step 1: Initialize the network by inserting input,/output vectors, maximum error, learning rate and threshold value of α.

Step 2: Initialize the weights

Step 3: Calculate error E for the first input

Step 4: Stop the procedure if $E > E_{max}$ and repeat the steps 2 and 3.

Step 5: Remove all possible nodes from the 2nd hidden layer

Step 6: Remove all possible nodes from the 1st hidden layer

Step 7: Update the weights as follows

$w_{kj}(new) = w_{kj}(old) + \eta.\delta_k f_j(net)$

Step 8:Insert another input chosen randomly and Repeat steps 5-7.

Step 9: Repeat steps 5-8 until the network remains unchanged.

Step 10: Repeat the procedure for different α to obtain most efficient architecture .

5. Experimental Methodology and Results

Many problems were tested to confirm the theory discussed in section 3. This paper presents one example which has two hidden layers two inputs and one output. Number of training patterns is 4. An arbitrary large number of nodes are assigned for each layer. Two activation functions log sigmoid and linear are used for hidden layers and output layers respectively. It is known that normalized weights are accelerated the back propagation [18]. Hence, we normalized the randomly selected weights. Learning rate fixed at 0.01. Comparison was made with backpropagation.

In this problem number of training cycles required for backpropagation training was 1836. The Table 1.shows the results obtained by new algorithm for same set of connection weights. Epochs refers the number of training cycles required to obtain error E by the new model respectively. Training initiated with 10 neurons in each layer. H1 and H2 refer the remaining number of neurons after removing all possible neurons using the proposed algorithm. Alpha changes from 1.5 to 2.0.

Table I: Results for the New Approach

H1	H2	Epochs	alpha
4	4	2301	1.5
5	6	1909	1.6
6	7	1786	1.7
9	9	1831	1.8
9	10	1977	1.9
10	10	1836	2.0

It implies that the new algorithm is able to minimize the network with lesser number of training cycles. For example, in this problem network with the configuration 2 – 10 – 10 – 1 can be reduced to 2 – 6 – 7 – 1 at $\alpha = 1.7$, which maintains the same error rate. It is clear that this result obtained by lesser training cycles relative to backpropagation training.

6. Conclusion

We have reviewed some major approaches to determine the hidden layer architecture and proposed a new approach determine the hidden layer architecture of multilayer artificial neural networks. The delta values of hidden neurons are used to identify the removable neurons. The proposed algorithm reduces the size of the network while maintaining the same error rate as backpropagation training. However, the performance of the network depends on several parameters such as weights of the connections, learning rate η, number of hidden neurons etc. Hence, some moderate changes will be done with these parameters as future improvements.

7. References

[1] A. N. Burkitt, "Optimization of the Architecture of Feed-forward Neural networks with hidden layers by Unit Elimination," *Complex Systems 5,* pp. 371-380, 1991.

[2] J. Amini, "Optimum Learning Rate in Back-Propagation Neural networks for Classification of Satellite Images (IRS-1D)," *Scientia Iranica,* vol. 15, pp. 558-567.

[3] Giovanna Castellano, "An Iterative Pruning Algorithm for Feedforward Neural Networks," *IEEE Transactions on Neural Networks,* vol. 8, pp. 519-531, May 1997.

[4] Faisal Muhammad Shah, Md. Khairul Hasan, Mohammad Moinul Hoque, Suman Ahmmed, "Architecture and Weight Optimization of ANN Using Sensitive Analysis and Adaptive Paticle Swam Optimization," *IJCSNS International Journal of Computer Science and Network Security,* vol. 10, no. 8, August 2010.

[5] D. Stathakis, "How many Hidden Layers and Nodes," *International Journal of Remote Sensing,* vol. 30, pp. 2133-2147, April, 2009.

[6] Shuxiang Xu, Ling Chen, " Novel Approach for Determining the Optimal Number of Hidden Layer Neurons for FNN's and its Application in Data Mining," *5th International Conference on Information Technology and Applications,* pp. 683 - 686, ICITA 2008.

[7] B. Yoganarayana, rtificial Neural Networks, PHI Learning Private Limited.

[8] Ah Chung Tsoi, Markus Hagenbuchner and Alessio Micheli, "Building MLP Networks by Construction," *IEEE,* pp. 549-554, 2000.

[9] S. Fahlman, "The Cascade-Correlation Architecture," May 1991.

[10] Sridhar S.S, "Improved Adaptive Learning Algorithm for Constructive Neural Networks," *International Journal of Comuter and Electrical Engineering,* vol. 3, no. 1, 2011.

[11] Antony N. Burkitt,Peer Ueberholz, "Refined Pruning Techniques for Feed-forward Neural networks," *Complex Systems 6 ,* pp. 479-494, 1992.

[12] G. E. H. R. J. W. D. E Rumelhart, "Learning Internal Representations by Error Propagations".

[13] J. S. D. A. S. Yann Le Cunn, "Optimal Brain Damage," *Advances in Neural Information Processing Systems.*

[14] Hassaibi. B., Stork D. G., "Second Order Derivatives for Network Pruning: Optimal Brain Surgeon," *Neural Information Processing Systems-92.*

[15] B. G. H. Don R. Hush, "Progress in Supervised neural networks," *IEEE Signal Processing,* vol. 10, pp. 8-39, 1993.

[16] P. C. Sudir Kumar Sharma, "Constructive Neural Networks: A Review," *International Journal of Engineering Science and Technology,* vol. 2, no. 12, pp. 7847-7855, 2010.

[17] M. R. A. S, "Recursive Dynamic Node Creation in Multi Layer Neural Network," *IEEE Transactions on Neural Networks ,* vol. 4, no. 2, 1993.

[18] S. Haykin, Neural Networks: A Comprehensive Foundation (2nd Edition), Prentice Hall .

N. Mihirini Wagarachchi received the B.Sc degree in Mathematics (special) from University of Colombo, Sri Lanka in 1993. She completed her M.Sc degree in Mathematics from University of Pune, India in 1998. She is a Lecturer at Faculty of Engineering of University of Ruhuna, Sri Lanka. Currently she is reading for her MPhil at Faculty of Information Technology of University of Moratuwa, Sri Lanka. Her research interests are artificial neural networks, optimization, graph theory and their applications.

Asoka S. Karunananda is the Professor and the Dean of the Faculty of Information Technology of University of Moratuwa, Sri Lanka. His research interests include the general area of Artificial Intelligence, Cognitive Systems, Machine Learning, Ontological Modeling, Multi Agent Systems and Theory of computing. He has presented and published more than 100 referred research papers locally and internationally. He is a founder member of Sri Lanka Association for Artificial Intelligence and served as the President of the same for several times. He has also contributed to promote computing education and research in computing by writing many books for general public and the student community. He has won several awards for his contribution to promotion of computing education, teaching excellence and research excellence.