American International Journal of Research in Science, Technology, Engineering & Mathematics

Available online at http://www.iasir.net

ISSN (Print): 2328-3491, ISSN (Online): 2328-3580, ISSN (CD-ROM): 2328-3629

AIJRSTEM is a refereed, indexed, peer-reviewed, multidisciplinary and open access journal published by International Association of Scientific Innovation and Research (IASIR), USA

(An Association Unifying the Sciences, Engineering, and Applied Research)

A Novel Technique for Optimizing the Hidden Layer Architecture in Artificial Neural Networks

N. M. Wagarachchi[1], A. S. Karunananda[2]

[1,2] Department of Computational Mathematics

Faculty of Information Technology

University of Moratuwa

SRI LANKA

Abstract: The architecture of an artificial neural network has a great impact on the generalization power. More precisely, by changing the number of layers and neurons in each hidden layer generalization ability can be significantly changed. Therefore, the architecture is crucial in artificial neural network and hence, determining the hidden layer architecture has become a research challenge. In this paper a pruning technique has been presented to obtain an appropriate architecture based on the backpropagation training algorithm. Pruning is done by using the delta values of hidden layers. The proposed method has been tested with several benchmark problems in artificial neural networks and machine learning. The experimental results have been shown that the modified algorithm reduces the size of the network without degrading the performance. Also it tends to the desired error faster than the backpropagation algorithm.

Keywords: backpropagation, delta values, feed-forward artificial neural networks, generalization, hidden layers

I. Introduction

Artificial neural networks have been showed their effectiveness in many real world problems such as signal processing, pattern recognition, and classification problems. Although they provide highly generalized solutions, we find several unanswered problems in using artificial neural networks. Determining the most appropriate architecture of artificial neural network is identified as one of those major problems. Generally, the performance of a neural network strongly depends on the size of the network. By increasing the number of layers generalization ability can be improved. However, this solution may not be computationally optimized. On the other hand, too many hidden neurons may over-train the data and which cause the poor generalization. Also, too few neurons under-fit the data and hence, network may not train the data properly. Thus, both too many and too few neurons show bad generalization. Therefore, determining the most suitable architecture is very important in artificial neural networks. As such, a large number of researchers have been carried out to model the hidden layer architecture by using various techniques. These techniques can be categorized as pruning techniques and constructive techniques. Pruning algorithms start with an oversized network and remove nodes until the optimal architecture occurs [1],[2],[3],[4] and [12]. Constructive algorithms [5],[6],[7],[8] do the other way. They build the appropriate neural network during the training process by adding hidden layers, nodes and connection weights to a minimal architecture. However, most of these methods are confined to networks with small number of neurons or single hidden layer neural networks. Hence, they have not addressed the existing problem of hidden layer architecture properly.

In this paper, a new pruning algorithm based on backpropagation training [11] has been proposed to design the optimal neural network. The optimal solution is obtained by two steps. First, the number of hidden layers in the most efficient network is determined. Then the network tends to the optimal solution by removing all unimportant nodes from each layer. The removable nodes are identified through the delta values of hidden neurons [9],[10]. The choosing of delta values was based on the fact that the delta values of the hidden layers are used to compute the error term of the next training cycle. Hence, delta value is a significant factor in error term. Thus, the delta values are used to identify the less saliency neurons and remove them from hidden neurons so that the error term tends to the desired limit faster than the backpropagation training.

The approaches of the other researchers are discussed in the next section. Section III describes the new algorithm and how to use the delta values in optimization of hidden layer architecture. The experimental method and results are discussed in section IV. Finally, section V presents the conclusions.

II. Current Approaches on Hidden Layer Architecture

The architecture of artificial neural network is crucial and therefore, many researchers have been approached to model hidden layer architecture by using several techniques. These approaches are based on pruning techniques, constructive techniques, or both pruning and constructive techniques.

A. Pruning Algorithms

The objective of pruning algorithms is to obtain the optimum architecture of artificial neural network by incrementally removing low-saliency neurons from the architecture. Optimal Brain Damage (OBD), presented by Le Cun and John Denker [1] is known as the most well-known and commonly used pruning algorithm, where parameters with low-saliency are detected based on the second derivative of the objective function and removed from the network. The intension of OBD is that, it is possible to obtain the network which performs 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 that the Hessian matrix is diagonal. To avoid the limitations of OBD, Hassabi and Stork [2] introduced Optimal Brain Surgeon (OBS) and they claim that Hessian matrix is non-diagonal for every problem and hence, weights elimination of OBD may not accurate. Also they argue that OBS is better than OBD and yields better generalization. However, still much computational limitations arise, 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 reaches the optimal size by removing neurons successfully from a large-sized trained network and then adjusting the remaining weights to maintain the original input-output behavior. 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. Firstly, it prunes neurons with less saliency using the impact factor. At each remove it tries to replace with some other suitable neuron which has the similar effect. The similarity is defined by using the correlation coefficient and if two neurons are highly correlated they can be merged. Burkitt in [12] claims that, for any particular network configuration, there is a continuous set of weights and biases that have the same value of the cost function. This set of weights defines a contour of the cost function in the space of all possible weights. In order to determine the optimum architecture, network is trained by the back propagation algorithm and required network can be obtained by eliminating weights which are close to zero.

B. 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 [7]. This number is crucial for better performance of the network and it makes as small as possible to avoid the complex computations during the training. The cascade correlation algorithm (CCA), proposed by S. E. Falhman [5] is the most well known and widely used constructive algorithm. This algorithm has been proposed as a solution of problems such as local minima problem. The dynamic node creation (DNS) algorithm [6] is supposed to be the first constructive algorithm for designing single layer feed-forward networks dynamically. Sridhar [8] improved the adaptive learning algorithm for multi-category tiling constructive neural networks for pattern classification problems. Md. Moniral et al [13] have proposed an adaptive merging and growing algorithm to design artificial neural networks. This algorithm merges and adds hidden neurons repeatedly or alternatively during the training, based on the learning ability or the training progress of the artificial neural network.

Even though methods to optimize the hidden layer architecture have been developed, still there are many limitations on those approaches. They have not fully addressed the problem regarding the topology of multilayer artificial neural networks such that the designing hidden layer architecture for better performances of multilayer artificial neural networks. Thus, the determining the optimal neural network architecture for a given task still remains as an unsolved problem.

III. Technology Applied in the New Model

In this research, a new algorithm based on the backpropagation training has been proposed. Backpropagation algorithm is considered as the most well known and widely used among the optimization algorithms of artificial neural networks. The main objective of the proposed algorithm is, to reduce the size of the network that can be trained faster than the back propagation algorithm, while maintaining the same or better performance.

First it decides the number of hidden layers for the most efficient network. Then removes unimportant nodes from each layer and tends to the optimal solution. In order to decide the optimal solution, consider a network with *n* inputs and *m* outputs. The number of input/output training sets is *N*. Let the network consisting of *M* total number of hidden neurons. Then train the different architectures of network by backpropagation algorithm. The number of hidden layers of network is *h*.

Now compute the 'Accuracy Ratio (χ) defined by

$$\chi = \frac{Accuracy\ Rate}{Number\ of\ Training\ cycles}$$

Then higher values of χ indicates that the network is more efficient. Hence, the number of hidden layers which gives the highest χ is to be considered as the number of hidden layers in the most appropriate network.

After deciding the exact number of hidden layers (h), trim the network for better performance. Start with fully connected network with h hidden layers and M number of total hidden neurons. Hidden layers are denoted by $H_1, H_2, \ldots H_h$ and the layer H_j contains n_{H_j} number of neurons. i.e

$$\sum_i n_{H_i} = M$$

While training the network less saliency neurons are identified and remove them from the hidden layers. After removing all the unimportant nodes network trains by using backpropagation algorithm until error $E(n)$ defines in (1) tends to zero.

The error of the n^{th} training cycle $E(n)$ can be written as

$$E(n) = \frac{1}{2}\sum_{k=1}^{m}(t_k(n) - o_k(n))^2, \qquad (1)$$

where, t_k and o_k denote the desired and actual outputs of the neuron k respectively at the n^{th} training cycle. N is the number of input/output training set.

The delta value of the output layer is defined as

$$\delta_k = f_o'(net_k)(t_k - o_k) \qquad (2)$$

where f_o is the activation function defined for the output layer. The delta value of the i^{th} neuron of the j^{th} hidden layer can be written as

$$\delta_i^{H_j} = f_{H_j}'(net_i)\sum_{k=1}^{n_{H_{j+1}}} w_{ki}\,\delta_k^{H_{j+1}} \qquad (3)$$

where f_{H_j} is the activation function defined for the hidden layer H_j. w_{ki} is the connection weight from i^{th} neuron of the j^{th} layer to the k^{th} neuron of $(j+1)^{st}$ layer. $\delta_k^{H_{j+1}}$ is the delta value of k^{th} neuron in the hidden layer H_{j+1}. Thus, the delta value of each neuron is computed by using the delta values of next adjacent hidden layer. After computing all the delta values weights are updated according to

$$w_{kj}^{H_j}(n+1) = w_{kj}^{H_j}(n) + \eta\delta_k^{H_j}(n)f_{H_j}(n). \qquad (4)$$

where η is the learning rate.

As error $E(n+1)$ will be calculated by using the updated weights, it is clear that there is a relation between $\delta(n)$ and $E(n+1)$. Now consider a fully connected multilayered artificial neural network with h number of hidden layers. Each layer contains a large number of hidden neurons. There are various upper bounds for the number of hidden neurons in ANNs [14], [15]. So that we have assigned a large number ($> N/h$) of neurons for each hidden layer. This number is much more than the maximum of the above upper bounds.

The delta values of hidden layers of the n^{th} training cycles $\left(\delta_i^{H_j}(n)\right)$ are used to compute the error of the $(n+1)^{st}$ training cycle, $E(n+1)$. Therefore, the correlation coefficient between the summations of delta values of

hidden layer H_j at the n^{th} training cycle $\sum \delta_i^{H_j}(n)$ and $E(n+1)$ has been used to identify the less saliency neurons. Let the correlation is denoted by r_{E,H_j}.

i.e

$$r_{E,H_j} = corr\left(\sum_{i=1}^{n_{H_j}} \delta_i^{H_j}(n), E(n+1)\right) \qquad (5)$$

A positive r_{E,H_j} implies that eliminating positive nodes from the hidden layer H_j, summation of delta values decreases and hence, $E(n+1)$ can be reduced. If r_{E,H_j} is negative, by removing neurons with negative delta values summation of delta values increases, and hence, error can be reduced. Therefore, eliminating nodes with positive or negative delta values according to their correlation error $E(n+1)$ becomes smaller than that of earlier. Thus, it converges faster than the backpropagation training. In the other words, the number of training cycles required to reach the solution becomes lesser than that of the backpropagation training.

Moreover, equation (4) implies that when delta is zero

$$w_{ki}^{H_j}(n+1) = w_{ki}^{H_j}(n)$$

That is there is no update of weight $w_{ki}^{H_j}$. Therefore, neurons with zero delta values do not have significant effect in minimizing the error. So that by removing nodes with zero delta values, size of the network can be reduced without degrading the performance of the network. Therefore, by removing neurons with

Small positive values when $r_{E,H_j} > 0$

and

Large negative values when $r_{E,H_j} < 0$

size of the network can make smaller while tending to the desired solution faster than the backpropagation training. When $r_{E,H_j} > 0$, a threshold value τ can be decided, such that by removing all delta values lies in $[0,\tau]$ reduces the error E of the training cycle. Similarly when $r_{E,H_j} < 0$ a threshold value τ' can be decided, such that by removing all delta values lies in $[\tau', 0]$ reduces the error E.

IV. Experiments and Results

Experiments on deciding the appropriate number of hidden layers and obtaining optimal solution are discussed in this section. For these experiments, datasets chosen from three well-known benchmarking problems namely car evaluation, breast cancer and Iris problem. More details on these data are available on [16]. These sets have been used in many projects in artificial neural networks and machine learning. All the sets were divided in to two classes for training and testing purposes. Different architectures of the above problems were considered. The log-sigmoid and linear functions were chosen as activation functions for hidden layers and output layer respectively. The learning rate in each case was fixed to 0.1. Network was trained for N input/output training sets until

$$E_N = \sum_{p=1}^{N}\sum_{k=1}^{m}\left(t_{pk} - o_{pk}\right)^2 < 10^{-4}$$

In order to determine the suitable number of hidden layers trained a fully connected networks with layers 1, 2, 3 4 and 5 by backpropagation algorithm. The accuracy of the results was tested by using the testing sets. Accuracy ratio (χ) was computed in each case. Table I shows the results. h and M refer the number of hidden layers and total number of hidden neurons respectively. TC denotes the number of training cycles. It is clear that generalization ability is improving when the number of layers is increasing. For example, in car evaluation problem, the accuracy rate of the output was 74.81% when there were two layers. But same data set agrees with 78.12%, if we used 5 hidden layers. The similar difference can be observed in breast cancer and Iris problems.

However, to train higher number of hidden layers it requires more time. When there are only 2 and 3 hidden layers in car problem network can be trained only by 17 and 23 training cycles respectively. But the numbers of training cycles require to train network with 4 or 5 hidden layer are much greater than that. i.e the improvement of the performance is negligible, comparing with the effort and time we put to train the networks. It can be described by the accuracy ratio. In this problem when there 2 or 3 hidden layers χ is much higher than that of 4 or 5 hidden layers. Hence, we can consider the network with 2 hidden layers, which provides the highest χ value, can be considered as the most appropriate network for this problem.

Problem	***N***	***M***	***h***	**TC**	**Accuracy Rate**	**Accuracy Ratio (χ)**
Car Evaluation	100	120	2	17	74.81	4.40
			3	23	75.28	3.27
			4	521	76.87	0.17
			5	308	78.12	0.25
Breast Cancer	120	160	2	6	85.58	9.51
			3	7	96.98	9.70
			4	17	97.24	5.72
			5	58	97.24	1.68
Iris	50	60	2	7	74.46	10.64
			3	19	78.70	4.14
			4	306	79.33	0.26
			5	387	81.93	0.21

Table I shows Accuracy Ratio χ for different benchmarking problems

Similarly, in cancer and Iris problems, networks with 3 and 2 hidden layers respectively provide the best highest values for χ. The summary of the results are shown in Table II. To test the modified architecture the networks with hidden layers given in Table II were considered. The total number of hidden neurons (*M*) appears in Table I divided among the hidden layers as shown in Table IV. While training the data by backpropagation algorithm, tested the correlation coefficient. Table III shows the correlation r_{E,H_j} for each layer. Then the network was trained by using the new algorithm. A benchmark comparison was done with backpropagation algorithm. The required numbers of training cycles for backpropation and the modified algorithm respectively are shown in Table IV. Also it shows the number of hidden neurons in initial configuration and the modified architecture. Accuracy rate of each case was calculated. The results show that for every case, it was able to find an architecture which has lesser number of neurons and shows better performance. First example in car evaluation problem shows positive correlation for the 1st layer and negative correlation for the 2nd layer. Hence, by removing neurons with small positive delta values from 1st layer and large negative delta values from the 2nd layer, modified architecture was obtained. It needs 17 training cycles to reach error $E = 10^{-3}$ by backpropagation algorithm. But by using new algorithm within 13 training

Table II Shows the number of hidden layer

Problem	***N***	***h***
Car Evaluation	100	2
Breast Cancer	120	3
Iris	50	2

Table III Shows Correlation coefficients

Problem	***N***	***h***	r_{E,H_1}	r_{E,H_2}	$r_{E,,H_3}$
Car Evaluation	100	2	0.85	-0.23	
Breast Cancer	120	3	-0.29	0.52	0.79
Iris	50	2	0.07	0.42	

Table IV Shows a comparison between Backpropagation and New algorithms

Problem	***N***	**Backpropagation Algorithm**			**Modified Algorithm**		
		Initial Configuration	**TC**	**Accuracy Rate**	**Modified Architecture**	**TC**	**Accuracy Rate**
Car Evaluation	100	70 – 60	17	74.81	66 – 56	13	75.73
Breast Cancer	120	60 – 50 – 50	10	96.98	42 – 34 – 38	7	97.06
Iris	50	30 – 30	11	78.56	28 – 24	9	81.20

cycles it can be trained. In this case, network tends to the desired limit faster and new model decreases the number of hidden neurons by about 9%. Moreover, it has upgraded the accuracy rate from 74.81% to 75.73%. In the car problem it was able to remove about 20% of hidden neurons without degrading the performance. Also in cancer problem with 3 layers hidden neurons reduce by 28% while improving the generalization ability.

Hence, it is clear new algorithm declines the error rapidly and reaches to the desired error faster than the backpropagation training.

V. Conclusion

In this paper, a new algorithm for multilayer hidden architecture was proposed. The algorithm is based on a pruning technique. Hidden neurons were pruned by using the delta values of hidden neurons. The correlation between the summation of delta values of each layer at the n^{th} training cycle and the error of the $(n+1)^{st}$ training cycle was considered to identify the less saliency neurons. Moreover, neurons with zero delta values were recognized as unimportant neurons as they do not have much effect on updating the weights. Therefore hidden neurons with small positive or large negative values (depends on the correlation) can be successfully used to reduce the size of the multilayer artificial network. In this paper a benchmark comparison done with the back propagation and it demonstrates that new approach can be used to minimize the network by maintaining the same error rate as back propagation training with lesser number of training cycles. Further, the modified architecture can be obtained with very limited computations. Generally, 5% - 30% of neurons can be removed from hidden layers without degrading the performance of the output. In future approach may be extended to more real-valued problems. Also comparison with existing approaches on hidden layer architecture will be discussed.

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

References

[1] Le Cunn Y.,Denkar, Solla S. A. "Optimal Brain Damage," *Advances in Neural Information Processing Systems* D.S. Touaretzky, Ed, San Mateo CA:Vol 2 pp 598-605, 1990.

[2] Hassabi . B., Stork. D. G., "Second Order Derivatives for Network Pruning: Optimal Brain Surgeon," *Neural Information Processing Systems*-vol 5,1993.

[3] Giovanna C., Anna M. F.,Marcello P., "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] Fahlman S. E, "The Cascade-Correlation Architecture," May 1991.

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

[7] 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.

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

[9] N. M. Wagarachchi, A. S. Karunananda, "Mathematical Modeling of Hidden Layer Architecture in Artificial Neural Networks," ,*Intenational Proceedings of Computer Science and Information Technology, Power and Energy Systems II*, Vol.56, pp154-159, November 2012

[10] N. M. Wagarachchi, A. S. Karunananda, "Optimization of Multi-layer Artificial Neural Networks Using Delta Values of Hidden Layers," *IEEE Symposium on Computation Intelligence, Cognitive Mind and Brain* , pp. 80-86, April 2013

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

[12] A. N. Burkitt, "Optimization of the Architecture of Feed-forward Neural networks with hidden layers by Unit Elimination," *Complex Systems 5*, pp. 371-380, 1991J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[13] Md. Monirul Islam, Md. Abdus S, Md. Faujul A, Xin Y., "A New Adaptive Merging and Growing Algorithm for Designing Neural Networksf," *IEEE Transactions on Systems Man and cybernetics*, vol. June 2009.

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

[15] Baum E. B, Haussler D., "What Size Network Gives Valid Generalization" *Neural Computations*-January,1989.

[16] A. Frank, UCI Machine Learning Repository ," Irvine, CA: University of California, School of Information and Computer Science., 2010. [Online]. Available: http://archive.ics.uci.edu