International Journal of Artificial Intelligence and Neural Networks– IJAINN
Volume 5 : Issue 1 [ISSN : 2250-3749]
Publication Date: 30 April, 2015

Towards a Theoretical Basis For Modelling of Hidden Layer Architecture In Artificial Neural Networks

[N. Mihirini Wagarachchi, Asoka S. Karunananda]

***Abstract*— Artificial neural networks (ANNs) are mathematical and computational models that are inspired by the biological neural systems. Just like biological neural networks become experts by learning from the surrounding, ANNs also have the ability to be experts in the particular area by training the network. Despite of their many advantages, there are some unsolved problems in applying artificial neural networks. Determine the most efficient architecture for the given task is identified as one of those major issues. This paper provides a pruning algorithm based on the backpropagation training algorithm to obtain the optimal solution of ANN. The pruning is done according to the synaptic pruning in biological neural system. Experiments were done with some well known problems in machine learning and artificial neural networks and results show that the new model performs better than the initial network in training data sets.**

***Keywords*— artificial neural netowrks; backpropagation; delta values; hidden layer architecture;**

I. Introduction

Artificial Neural networks, commonly referred to as neural networks are massively parallel and highly interconnected information-processing systems, design by mimicking the information-processing behavior of biological neural system in human brain [1]. Neural networks are able to provide good approximation solution because of their properties like, massively parallel and ability to learn. Moreover, as human brain does they have capability of adapting their synaptic weights to change in the surrounding environment. Therefore, neural networks have been used in several real world problems such as signal processing, pattern recognition, and classification problems. Although applying neural networks in real world problems show many advances, still some unanswered questions can be found. Determine the most appropriate architecture for the given task is identified as one of the central issues in ANNs. According to the researchers, number of hidden layers and number of neurons in each hiddenlayer are two significant factors and performance of the network strongly depends on these factors. So determining the most appropriate architecture for the specific task is very important and hence, many researchers have been attempted to solve this problem by using different techniques. Basically, pruning, constructive, pruning-constructive and some evolutionary methods have been used to determine the most appropriate hidden layer architectures in ANNs.

A pruning algorithm starts with a large-sized network and gradually eliminates neurons until optimal solution occurs [2], [3], [4], [5]. In contrast, a constructive algorithm adds new neurons and weights to a minimal network to obtain the desired solution [6], [7], [8], [9]. However, most of the existing solutions in ANNs are to solve networks with single hidden layer. But it is known that the generalization ability will be improved by increasing the number of hidden layers [10],[11]. However, this solution may not be computationally optimized. Hence, this paper presents a new pruning method to determine the hidden layer architecture based on the backpropagation training algorithm [18]. The new algorithm first decides the number of hidden layers in an appropriate network and then it eliminates unnecessary neurons until the optimal solution occurs. The pruning method of neurons is inspired by the concept of synaptic pruning in biological neural systems [12], [13]. The removable nodes are identified by using the delta values of hidden layers [14],[15]. If delta value of a particular neuron is zero for all the inputs in the training set, that node is identified as a weak neuron. Thus, the identified nodes can be removed and connection weights are merged with similar weight vectors. For more précised solution correlation between summation of delta values of the particular layer and the error of the training cycle is used [14],[15].

The method of determining the number of layers and recognizing the removal nodes will be discussed in the next section. Section III presents the new algorithm while the experiments and results present in the section IV. The section V contains the theoretical basis for pruning. Finally, conclusion is given in the section V1.

II. The Approach to the New Model

Designing hidden layer architecture is very important in artificial neural networks. Usually the numbers of neurons in input and output layers are defined as the sizes of input and output vectors of the problem. So that the determining the architecture of neural network implies deciding the number of hidden layers and hidden neurons in each layer and which still is an open research challenge. Changes of neural network architecture cause for different measures of generalization power. Generalization is the most important feature in ANN that gives the amount of correct mappings when inserted a set

N. Mihirini Wagarachchi
University of Moratuwa
Sri Lanka

Asoka S. Karunananda
University of Moratuwa
Sri Lanka

of unseen data. It is noticeable that, although the number of parameters is high in two or three hidden layer network, they show significantly better performance than the single hidden layer networks. The generalization power goes up as the number of hidden layers increases. In [10] and reference therein claim that efficiency of learning procedure is better than two layer network than one layer as more information can be extracted from each example with the previous. However, adding more hidden layers to network may cause high computational complexity and increases of training time. On the other hand, too large or too small number of hidden neurons shows both good and bad performances. Using too many neurons network can be trained easily and data will be fitted more accurately. Also it easily avoids the local minima problem. But due to many degrees of freedom it arises high computational complexity and bad generalization performance. Even though, too few neurons save computational cost, they learn very slow or sometimes may not learn the dataset at all. Hence, the determining the most appropriate architecture before train the network is very important. A novel approach is discussed in this paper and it differs from the most of the other approaches first it is not restricted to a single hidden layer network. Second, in eliminating nodes, rather than considering weight matrix it concerns on the set of delta values of each hidden layer. Initially, it decides the number of layers in the most efficient network and then eliminates all possible neurons from the hidden layers.

A. Determining the Number of Hidden Layers

The learning speed and the generalization ability are known as the two main measures of artificial neural networks. Generalization the most important property required by training algorithm which can be defined as the number of correct mapping on the set of data which have not used for training purpose. It is known as generalization ability also referred to as mapping rate can be improved by increasing the number of layers. By mapping rate we mean the ratio of outputs lie in the interval $[t_k - \epsilon, t_k + \epsilon]$ to the total number of outputs. t_k is the desired output at the k^{th} neuron and ϵ is a predefined positive constant. Most current approaches are restricted to single hidden layer networks or at most three hidden layer networks due to the computational complexity and training time of using more hidden layers. However, these quantities depend on the vectors of the training set. Although generalization power or mapping rate improves with the number of hidden layers, sometimes to train network with several hidden layers, it takes a long time. Hence, form some particular networks increment of mapping rate is negligible with comparing to the training time. Therefore, we say that the most efficient network is the network which can be trained fast with good generalization power. However, if we partially trained the network for some particular number of training cycles, at the end of training the most efficient network should provide smallest error or zero error.

i.e $\|t_k - \mathrm{o_k}\| \approx 0, \ \forall\, k$

Where t_k and o_k are the desired and observed outputs of k^{th} neuron in the output layer. If there are N input/output training patterns error (E) is defined as

$$E = \frac{1}{2N} \sum_{p=1}^{N} \sum_{k=1}^{m} \left(t_{pk} - o_{pk}\right)^2 \qquad (1)$$

So that neural networks provide good generalization power within small number of training cycles show large values of 'Accuracy Factor' (AF) defined by

$$AF = \frac{MR}{\sqrt{E}}\,. \qquad (2)$$

Therefore, the number of layers which provides the highest AF for any particular dataset is to be considered as the number of layers in the most efficient network. Thus, to decide the number of layers in the most appropriate network, we design an artificial neural network with arbitrary number of hidden neurons and partially train by backpropagation algorithm by changing the number of layers. Compute accuracy factor (AF) defined in (1) for each network. Then obtain the number of hidden layers which gives the maximum value for AF. For example in the datasets of Cancer problem considered in section IV, two layer network provides the highest AF after 500 iterations. Therefore, two layer architecture network of cancer problem is considered as the most efficient network.

Now we know how many hidden layers must be there in the solution space for the given task. But there are several architectures; all provide the same efficiency [16]. Moreover, by constraining an oversized network, both learning speed and generalization power can be improved [10]. Therefore, we reach to optimal solution by removing unimportant neurons from the hidden layers.

B. Eliminating Hidden Neurons

The efficiency of neural network measured not only by its performance of data in the training set but also by how it performs for unseen data. If a network has ability to perform well for unseen data then the network has good generalization power. It is known that if the number of parameter is the same or greater than input/output training patterns, network will be able to map data perfectly, by reducing the problem of local minima. But this approach may not fit unseen data correctly as it has not learn to generalize at all [11],[17]. Hence, using too many neurons may cause to poor generalization. On the other hand, too few neurons may save computational cost while showing some good generalization. But they may not learn data properly. Therefore too many and too few neurons have both advantages and disadvantages. Hence, determining the optimal architecture is very important, before train the network.

In this research we start from an oversized network and eliminate all possible unimportant neurons to obtain the optimal solution. The removable neurons are identified by using delta values of hidden layers. Authors have been discussed on using delta values in optimizing delta values in [14] and [15].

The backpropagation algorithm [18] is the most well known and widespread algorithm among many numerous algorithms that have been proposed to train artificial neural networks. The basic idea behind this is to obtain weight matrices in order to minimize the error define in (1). For minimum error

$$\frac{\partial E}{\partial w_{kj}} = 0 \tag{3}$$

Where w_{kj} be the weight to k^{th} neuron of the output layer from j^{th} neuron of the adjacent hidden layer.

Let $\Delta w_{kj}(p)$ be the weight correction for the p^{th} training pattern. Then $\Delta w_{kj}(p)$ is proportional to the partial derivative in (3).

i.e

$$\Delta w_{kj}(p) = -\eta \frac{\partial E}{\partial w_{kj}}$$

Thus,

$$\Delta w_{kj}(p) = \eta \delta_k(p) f_h(net_j) \tag{4}$$

Where, net_j is the input vector of neuron j for training pattern n, f is the activation function, and η is the learning rate.

Equation (4) implies that zero delta values means there are no update of the particular weights. Therefore, hidden neurons with zero delta values are not contributed to decrease the error. This implies that neurons with zero delta values are less salience neuron and by removing them does not affect to error decay process.

The 'local gradient' $\delta_k(p)$ is defined by

$$\delta_k(p) = \left(t_{pk} - o_{pk}\right) f'_h(net_k)(p) \tag{5}$$

for neuron k in output layer.

or

$$\delta_k(p) = \sum_{k=1}^{n_h} \delta_k^{h+1}(p) w_{kj}^{h+1} \; f'_h(net_k)(p) \tag{6}$$

For neuron k in the hidden layer h

Empirical results show that very often, there is a correlation between summation of delta values of hidden layers and the output error which can be positive or negative. Thus, we use this correlation to identify the removable neurons. Let this correlation be denoted by $r_{\delta,E}$.

$$r_{\delta,E} = corr\left(\sum_{k=1}^{n_h} \delta_k , E\right) \tag{7}$$

Therefore, to obtain most precise network, the correlation defined by (7) can be used. If $r_{\delta,E}$ is positive, minimal architecture obtains by removing neurons with positive delta that are very close to zero. In same manner when correlation is negative, neurons with large negative delta values remove to obtain the desired architecture.

The pruning has the same meaning of synaptic pruning in neuroscience. It facilitates changes in neural configuration by removing weak neurons and synapses while strengthening the remaining [19]. In same manner network initiates with an arbitrary number of neurons and identify removable neurons and then removes those from the network while merging the synaptic weights with some similar neurons.

C. *Merge the Similar Neurons*

Just as in synaptic pruning, we strengthen weights of the network by merging possible neurons. So that more efficient network will be obtained. Initially start with random normalized weights which have unit lengths. Let

$$V = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n_l} \\ v_{21} & v_{22} & \dots & v_{2n_l} \\ \vdots & \vdots & \vdots & \vdots \\ v_{n_{l'}1} & v_{n_{l'}2} & \dots & v_{n_{l'}n_l} \end{bmatrix}$$

be the weight vector form layer H_l to H_{l+1}.where $n_{l'}$ is the number of hidden neurons in the layer H_{l+1}. Let V^{R_i} and W^{C_j} represent the i^{th} row and j^{th} column of V respectively. i.e

$$V^{R_i} = \begin{bmatrix} v_{i1} & v_{i2} & \dots & v_{in_h} \end{bmatrix}$$

and

$$W^{C_j} = \begin{bmatrix} v_{1j} & v_{2j} & \dots & v_{n_h j} \end{bmatrix}^t$$

Then the vectors with same orientation are considered as similar vectors. i.e if V^{R_i} and V^{R_k} are similar vectors,

$$\left\langle \frac{V^{R_i}}{\|V^{R_i}\|}, \frac{V^{R_k}}{\|V^{R_k}\|} \right\rangle = 1 \tag{8}$$

Thus, if neuron k in layer H_l is identified as the removable neuron, and V^{R_k} and W^{C_j} are the similar vectors to V^{R_i} and W^{C_i} in input and output weight matrices V and W respectively. Then V^{R_k} and W^{C_j} are replaced by $V_{new}^{R_K}$ and $W_{new}^{C_j}$ respectively as follows.

$$V_{new}^{R_K} = V^{R_i} + V^{R_k} \tag{9}$$

$$W_{new}^{C_j} = W^{C_i} + W^{C_j} \tag{10}$$

III. The New Algorithm

Initiate a network with h number of hidden layers and total number of M hidden neurons. l^{th} hidden layer is denoted by H_l , $l = 1,2, \dots, h$. Let the number of hidden neurons of the k^{th} layer is n_k. Then

$$n_1 + n_2 + \dots + n_h = M.$$

Divide the set of data into two classes for training and testing purposes. Let the number of different vectors in

training set be N. Initialize random normalized weights. Let V and W be the input and output weight matrices of hidden layer H_l from layer H_{l-1} and to layer H_{l+1} respectively,

Train the network for a fixed number of training cycles by backpropagation algorithm and observe the mapping rate (MR) on the testing set. Compute the accuracy factor (AF) according to (2). The number of layers in the network with the highest accuracy rate is to be considered as the number of layers in the most efficient network. Now let q be the number of such layers in this network.

In order to find the number of hidden neurons consider the network with q number of hidden layers. Let $n_1, n_2, \dots, n_q$ be the hidden neurons in each layer respectively. Compute the correlation coefficient $r_{\delta,E}$ for each layer. If $r_{\delta,E}$ is negative, remove all nodes whose delta value is negative and close to zero for every input in the training set. That is for each input, we consider the intersection of the largest α_k number of negative delta valued neurons from the hidden layer k and remove them from the network. Where $0 < \alpha_k < n_k, \forall\, k$

Similarly, if $r_{\delta,E}$ is positive eliminate those nodes whose delta value is positive and close to zero. In this case we choose intersection of sets with minimum delta valued neurons. After removing all possible nodes, network will be trained by backpropagation algorithm until it reaches to the desired error level. By assigning different values for α_k, most appropriate network can be obtained. While removing the neurons, connection weights are merges with similar neurons according to the equations (9) and (10).

IV. Experiments and Results

In order to test the performance of the proposed method, seven data sets were chosen from different domains. The details of data sets are shown in Table I. These are widely used data in machine learning and artificial intelligence experiments. More details of these data are available on [20]. Each data set is divided into two classes namely, training and testing sets. First, for each problem decided total number of hidden neurons (M). It is known that if the number of parameter is the same or greater than input/output training patterns (N), network can be trained without local minima problem. Thus, initially chose M such that $M \approx N$.

TABLE I. DETAILS OF DATA SETS

Problem	Size of training set (N)	Size of testing set	Number of inputs (n)	Number of outputs (m)
Cancer	525	174	9	2
Card	518	172	51	2
Diabetes	576	192	8	2
Flare	800	266	24	3
Glass	161	53	9	6
Heart	690	230	35	2
Horse	273	91	58	3

TABLE II. ACCURACY FACTOR OF CANCER PROBLEM

Number of Hidden Layers	*MR*	*error*	*AR*
1	0.873	0.5062	0.05
2	0.965	0.0051	0.14
3	0.965	0.0068	0.12
4	.0965	0.0068	0.12
5	0.971	0.0068	0.12
6	0.971	0.0060	0.13

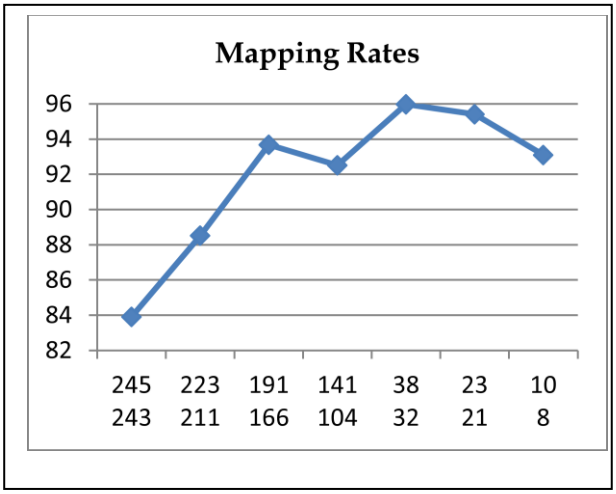

Fig 1. Mapping rates against the hidden layer neurons in pruning process

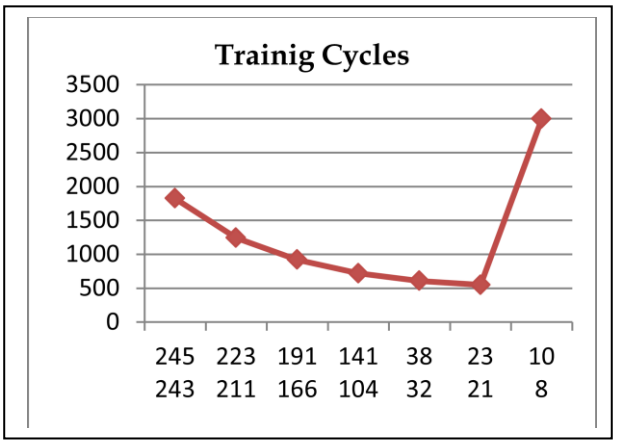

Fig 2. Training cycles against the hidden layer neurons in pruning process

TABLE III. IDDEN LAYERS FOR MOST EFFICIENT NETWORKS

Problem	*Cancer*	*Card*	*diabetes*	*flare*	*glass*	*heart*	*horse*
Number of hidden layers	2	3	6	2	1	3	2

TABLE IV. COMPARISON OF NEW MODEL WITH THE INITIAL CONFIGURATION

Problem	Initial Network			New Model		
	Hidden layer Architecture	Training Cycles	Accuracy	Hidden layer architecture	Training Cycles	Accuracy(%)
Cancer	250 – 250	1752	94.83%	38 – 32	608	95.36%
Card	200 – 200 – 100	388	86.05%	176 – 170 – 58	347	86.63%
Diabetes	100 – 80 – 80 – 80 – 80 – 80	493	71.35%	75 – 45 – 33 – 37 – 40 – 43	432	71.35%
Flare	400 – 400	8714	99.98%	183 – 215	4012	99.98%
Glass	80 – 80	4091	77.36%	56 – 57	4896	83.02%
Heart	200 – 200 – 200	851	81.30%	90 – 102 – 112	320	87.00%
Horse	150 – 150	1221	64.84%	72 – 80	790	68.31%

Divided *M* number of hidden neurons among the *h* hidden layers. For example in cancer problem there are 525 training sets. So at the beginning, networks were created with 500 hidden neurons. When there are two layers each layer contains 250 hidden neurons

The log-sigmoid function was used as the activation function for hidden layers and linear function was used at the output layers. The learning rate fixed as 0.1 for all the networks. Each data set trained for 500 iterations to compute the accuracy ratio (*AF*). Table II gives the mapping rate (*MR*), mean square error (*E*) of training set after 500 iterations of cancer problem. It is clear that mapping rate is increasing with the number of hidden layers. But to train a large number of hidden layers it takes more time. For example after 500 iterations two layer network reaches to error 5.1×10^{-3}. But for 5 layer network this value becomes 6.8×10^{-3}. According to table II network with two hidden layers is identified as the most efficient network. Table III shows the number of hidden layers in the most efficient network for each problem. However, it is agree that most of the problems can be solved by using networks with at most three hidden layers. The above networks given in the table III trained for new model. Table IV represents the performance of new model. It is clear that new algorithm reduce the size of network by improving the generalization power and limiting the number of training cycles.

Moreover, it shows poor measures in generalization and training cycles when over-pruning and under-pruning. For example, initially we have chosen two layer neural network with 500 hidden neurons for the cancer problem. The network can be trained by 1752 Iterations to reach 10^{-3} error level. When we start to prune neurons, initially it shows mapping rate less than that of initial network. But when it continues mapping rate improves while number of training cycles decreases. However, for further continuation although training cycles decreases mapping rate goes up. Fig.1 and Fig.2 explain how mapping rate and training cycles change with the number of hidden neurons. According to these graphs, for cancer problem 38 – 32 hidden neurons give better performance. However, 23 – 21 configuration also acceptable as it shows similar mapping rate and training cycles as 38 – 32 configuration.

v. The Theoretical Basis

Artificial neural networks are mathematical and computational models that mimic the information processing behavior of the human brain. The human brains are well structured networks which are capable of learn to adapt to the surrounding environment by experience from the child's birth. This is process is known as the brain plasticity which challenges the idea that brain functions are fixed and cannot be changed. So the brain is not hard-wired. New dendrites and synaptic are able to formed through the entire life [3], when it affected a part of the brain, non-damaged or intact cortex adjacent to the affected part can function to regain the damaged neurons. If a person becomes paralyzed or in the situation such that he cannot move his limb, he can recover by compensating with other limb. However, this process takes certain time, hours, months or sometimes several years. Although, majority of the developments of human brain take place at the infant age, it continues beyond that level into maturity and even older ages. Nodes of the artificial neural networks are working in the same manner in the training process as the neurons in brain work at input-output processing. Training neurons refer the modification of synaptic weights of neural network in order to tend to the desired output. In this research we used pruning technique based on the backpropagation algorithm. Just as synaptic pruning occur in neural system in human brain, artificial neural networks remove less salience neurons and reach to the desired output.

A. *Synaptic Pruning*

Synaptic pruning refers the eliminating process of extra synapses and neurons in order to increase the efficiency of neural system. Until about 2 years from the child birth new neurons and synapses are increasing very rapidly. At the end of this process brain contains far more neurons and synapses than needed to function it. At this age process of synaptic pruning begins to eliminate malfunction neurons and synapses and strengthen the reaming to increase the efficiency of then system of neural network. This process continues until 10 – 15 years. By this it eliminate half of the synapses contained at the age of 2 years.

However, over pruning and under pruning of synapses show symptoms of mental disorders. For example mental disorders such as autism and Rett's syndrome are assumed to be happened due to the lack of synaptic pruning and alternation of neuronal number during the development [21],[22]. In contrast schizophrenia is suggested to due over-pruning of synapses during the developments of late adolescence.

B. Behavior of the New Model

Behavior of artificial neural networks is very similar to the biological neural system. Synaptic weights are correlated with the synapses of human brain cells. In addition, artificial neural networks can be trained for the given task and just as human brain works, we start artificial neural network with a randomly selected large number of hidden neurons. There are various limitations for this number which we have been discussed in [15]. However, some of these neurons do not contribute in error decay process and hence, as synaptic pruning occurs in human brain; those neurons can be eliminated from the network while strengthening the remaining neurons. Nevertheless, pruned network won't end with a better result if inappropriate pruning is done. This behavior can be explained by synaptic over-pruning and under-pruning; neural networks also show bad performance in use of incorrect number of neurons elimination. Thus, this behavior agrees with the concept of synaptic pruning and neuroplasticity in the human brain.

VI. Conclusion

In this research a pruning method was presented to obtain an efficient network. Firstly, it identifies the number of layers in the most appropriate network by using the accuracy ratio and then prunes unimportant neurons from the network. The pruning is done based on the back propagation training algorithm and the removable neurons are identified by using the delta values of hidden neurons. The modified architecture is lesser than half of the neurons of its initial configuration. The behavior of network has been compared with the process of pruning system of biological neural system. As in synaptic pruning, it can be observed that both over pruning and under pruning show bad performance. Although hidden layer architecture is crucial in artificial neural networks, the performance of network depends on several other variables. Such as types of input, connection weights, activation function etc. Hence, it needs to modify the algorithm by considering these parameters as future works.

Acknowledgment

This research is supported by a grant from HETC project. The authors express their sincere thanks to the staff of HETC for their helpful corporation.

References

[1] S. Heykin, "Neural Networks and Learning Machines", *PHI Learning Private Limited,*

[2] 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

[3] 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.

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

[5] 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.

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

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

[8] 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.

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

[10] Y. L. Cun,"Generalization and Network Design Stratergies", *In connectionism in Perspective,* R. Pfeifer, Z. Schreter, F.Fogelman-Soulie, and L. Steels eds. Amestradam Elsevier,1989,pp 143-155.

[11] S. Karsoliya, "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture", *International Journal of Engineering Trends and Technology*- Vol.3,Issue6- 2012,pp 714-717

[12] R. Viviani ,M. Spitzer, "Developmental pruning of synapses and category learning" , *ESANN'2003 proceedings - European Symposium on Artificial Neural Networks,* Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 287-294

[13] M. H. Johnson,"Functional Brain Development in Humans" *Macmillan Magazines Ltd,* vol. 2,July 2001,pp 475-483

[14] 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

[15] 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.

[16] A.N. Burkitt, "Refined Pruning Tecniques for Feed-forward Neural Networks ", Complex Systems 6 1992, pp 479-494

[17] http://en.wikipedia.org/wiki/Overfitting

[18] Rumelhart, D. E. Hinton,Williams,"Learning Internal Representations by Error Propagation", *In Parrellel Distributive Processing, Exploration in the microstructure in Cognition,* Vol.1, Bradford Books, Cambridge MA, 1986.

[19] http://en.wikipedia.org/wiki/Synaptic_pruning

[20] A. Frank, UCI Machine Learning Repository ," Irvine, CA: University of California, School of Information and Computer Science., 2010.

[21] D. G. Glaze," Rett Syndrome: of Girls and Mice—Lessons for Regression In Autism", *Mental Retradation and Development Disabilities Research Review,* pp 154-158, 2004.

[22] J. L. Neul, H. Y. Zoghbi, "Rett Syndrome: A Prototypical Neurodevelopmental Disorder", *The NeuroScientist,* Vol 10, pp 118-128, 2004