ENTROPY LEARNING IN NEURAL NETWORK

Geok See Ng*, D. Shi, A. Wahab and H. Singh

Intelligent System Laboratory, School of Computer Engineering,
Nanyang Technological University,
Blk N4 #2A-32, Nanyang Avenue, Singapore 639798

Received 19 February 2003, Accepted 21 August 2003

ABSTRACT

In this paper, entropy term is used in the learning phase of a neural network. As learning progresses, more hidden nodes get into saturation. The early creation of such hidden nodes may impair generalisation. Hence entropy approach is proposed to dampen the early creation of such nodes. The entropy learning also helps to increase the importance of relevant nodes while dampening the less important nodes. At the end of learning, the less important nodes can then be eliminated to reduce the memory requirements of the neural network.

1. INTRODUCTION

The mapping capability of Back-propagation neural network depends on its structure, that is, the number of layers and hidden nodes. In most cases, given a certain application and training data, the network structure should be pre-determined. A network that has a structure simpler than necessary cannot give good approximations to patterns in the training set. A network with more layers and hidden nodes can perform more complicated mappings. However, better performance on unseen data, that is, generalisation ability, implies lower order mapping. A network that is more complicated than necessary with too many hidden nodes will over-fit the data and thus perform poorly on untrained data. Bigger networks also need larger data samples for training. It was pointed out, based on an information theoretic measure, that the required number of patterns in the training set grows almost linearly with the number of hidden nodes¹. With a smaller network size, less memory is required to store the connection weights and the computational cost of each iteration decreases.

In this paper, a total cost function, which includes an additional entropy penalty term is proposed during the training process of Back-propagation neural networks. At the end of entropy learning, inactive nodes created can be eliminated without affecting the performance of the original network.

*Corresponding author - e-mail: gsng@ieee.org, asgsng@ntu.edu.sg

2. THE ENTROPY LEARNING METHOD

Mutual information has been applied to input pruning as well as hidden nodes pruning^{2, 3}. Deco et al.⁴ have used mutual information measures to eliminate over-training in supervised multi-layer network. Their entropy cost function is:

$$E = \underbrace{\sum_{p=1}^{N_p} \sum_{k=1}^{N_o} (T_k^p - O_k^p)^2}_A + \lambda \left(\underbrace{- \sum_{j=1}^{N_h} \bar{P}_j \log \bar{P}_j}_B + \underbrace{\frac{1}{N_p} \sum_{p=1}^{N_p} \sum_{j=1}^{N_h} P_j^p \log P_j^p}_C \right) \quad (1)$$

where N_p is the number of patterns, N_h is the number of hidden nodes N_o is the number of output nodes, T_k^p is the target output of k th output node for pattern p , O_k^p is the activation value of k th output node for pattern p , P_j^p is the normalised activation value of j th hidden node for pattern p , $\bar{P}_j$ is the average of the normalised activation value of j th hidden node and λ is the momentum.

In Equation 1, term B tries to move $\bar{P}_j$ through time towards the two extreme ends of the Sigmoid function and term C tries to move the P_j^p towards the 0.5 region of the Sigmoid function. However, term C acts temporally across the batch patterns N_p and thus will have a smoothing effect across the activation encoding of sequential set of pattern inputs. In dynamic time-series forecasting problems, this term is added to average the time-series points⁵ as most of the hidden node activations lie in the linear zones of the Sigmoid function. In static pattern classification problems such as those discussed in this paper, there is no reason to apply this constraint across the temporal domains as it will lead to poorer classification accuracies. Similarly, the entropy penalty term concept introduced in this paper helps the learning process to move away from the saturation zones. Furthermore, term A which is the sum-squared error in Equation 1 has been replaced by the cross-entropy error in our proposed approach.

Kamimura and Nakanishi⁶ added a simple entropy measure to the cost term as shown in Equation 2

$$E = \sum_{p=1}^{N_p} \left[\sum_{k=1}^{N_o} (T_k^p - O_k^p)^2 + \lambda \left(- \sum_{j=1}^{N_h} P_j^p \log P_j^p \right) \right] \quad (2)$$

The entropy measure moves the normalised hidden node representation towards their extremes and speeds up training time. Kamimura and Nakanishi⁶ argued that the entropy measure had the ability to concentrate the information on a few hidden nodes but no details were given on the network behaviour. Only terms A and B of Equation 1 are used in their work which is similar to the approach described in this paper. Our paper enhances the incremental weight update approach used by Kamimura and Nakanishi. Kamimura and Nakanishi⁷ tried to solve the saturation problem by introducing another constraint $\sum_{j=1}^{N_h} y_j^p = \theta$ into their equation. However the setting of θ value is problematic. In this paper, the entropy is proposed to solve the early

saturation problem. The effect of entropy learning is investigated using five relevance criteria, three of which are proposed in this paper. The proposed method is also different from Kamimura-Nakanishi in the aspect that no inhibitory connections are used.

2.1 Network entropy

A Back-propagation neural network structure is shown in Figure 1. Suppose that an entropy function can be defined with respect to the activity Y_i of the hidden node i . If the entropy is minimised, only a certain subset of the hidden nodes will be turned on. On the other hand, if the entropy is maximised, all the hidden nodes are nearly equally activated in their linear region.

Y_i , which is the activation value of a hidden node i , is obtained by

$$Y_i = A_i (net_i) \quad (3)$$

where A_i is the Sigmoid function and net_i is defined by

$$net_i = \sum_{k=1}^{N_k} V_{ik} X_k \quad (4)$$

where X_k is the activation value of input node k , V_{ik} is the weighted connection from input node k to hidden node i , and N_k is the number of input nodes. The activation value Y_i is normalised as follows:

$$P_i = \frac{Y_i}{\sum_{j=1}^{N_h} Y_j} \quad (5)$$

By using this normalised activity, an entropy function can be formulated by

$$H = -\sum_{i=1}^{N_h} P_i \log P_i \quad (6)$$

Figure 1 : A Back-propagation neural network structure.

In the learning phase, the entropy function is minimised or maximised with the cross-entropy cost function as shown in Figure 2. Through this cost function, the growth of the hidden node's representation can be controlled. By minimising the entropy term, the hidden nodes' activation can be forced to develop extreme representation near 0 and 1 that can be used for pruning the redundant nodes through saturation. By maximising the entropy term, the hidden activation value is moved towards 0.5 region. The values in this range require net_i to be near zero and causes many of the weights V_{ik} to move towards zero similar to the effect of weight decay.

Figure 2: Network entropy.

A state with minimum entropy means that most nodes are operating in the non-linear region near the extreme values. As the training proceeds, the hidden nodes' activation is pushed towards their extreme value while some strong and pertinent nodes remain active in the linear region. As training progresses, more nodes get into saturation around 0 or 1. It is important to note that creation of such nodes may impair generalisation performance. To prevent the network from being driven into saturation before it starts to learn, the training is conducted in a form analogous to annealing⁸ as detailed in Section 2.3 by introducing a relaxing cycle called entropy cycle. These inactive nodes can then be eliminated sequentially according to five criteria mentioned in Section 3 without affecting the performance of the original network. If necessary, the entropy reduction process should be halted and a recovery phase of training cycles should be carried out to move the relevant hidden nodes back to the linear region.

2.2 The cost function

Our proposed total entropy cost function with the additional penalty term αH can be formulated as follows:

$$J = \alpha H + \beta E \quad (7)$$

where α is the entropy rate, H is the entropy cost function, β is the learning rate, and E is the cross-entropy cost function defined as:

$$E = - \sum_{p=1}^{N_p} \sum_{k=1}^{N_o} \left[T_k^p \log O_k^p + (1 - T_k^p) \log (1 - O_k^p) \right]. \quad (8)$$

In fact, the cross-entropy error function has much less local minima compared to the mean squared error function⁹.

Differentiating Equation 7 with respect to hidden-output weight W_{ji} , the general learning rule can be obtained from:

$$\begin{aligned} \frac{\partial J}{\partial W_{ji}} &= \alpha \frac{\partial H}{\partial W_{ji}} + \beta \frac{\partial E}{\partial W_{ji}} \\ &= \beta \frac{\partial E}{\partial W_{ji}} \end{aligned} \quad (9)$$

where $\frac{\partial E}{\partial W_{ji}} = (O_j - T_j) Y_i$.

Differentiating Equation 7 with respect to input-hidden weight V_{ik} , the general learning rule can be obtained from:

$$\frac{\partial J}{\partial V_{ik}} = \alpha \frac{\partial H}{\partial V_{ik}} + \beta \frac{\partial E}{\partial V_{ik}} \quad (10)$$

where $\frac{\partial E}{\partial V_{ik}} = \sum_{j=1}^{N_o} (O_j - T_j) W_{ji} Y_i (1 - Y_i) X_k$ and $\frac{\partial H}{\partial V_{ik}}$ is derived as follows:

$$\begin{aligned} \frac{\partial H}{\partial V_{ik}} &= \frac{\partial H}{\partial P_i} \frac{\partial P_i}{\partial Y_i} \frac{\partial Y_i}{\partial V_{ik}} \\ &= -(1 + \log P_i) \frac{\sum_{j=1}^{N_h} Y_j - Y_i}{\left(\sum_{j=1}^{N_h} Y_j \right)^2} \frac{\partial Y_i}{\partial A_i} \frac{\partial A_i}{\partial V_{ik}} \\ &= -(1 + \log P_i) \frac{\sum_{j=1}^{N_h} Y_j - Y_i}{\left(\sum_{j=1}^{N_h} Y_j \right)^2} Y_i (1 - Y_i) X_k \end{aligned} \quad (11)$$

So the weight update equations for the entropy minimisation are

$$\Delta W_{ji} = - \frac{\partial J}{\partial W_{ji}} = -\beta (O_j - T_j) Y_i \quad (12)$$

and

$$\begin{aligned}\Delta V_{ik} &= -\frac{\partial J}{\partial V_{ik}} \\ &= \alpha (1 + \log P_i) \frac{\sum_{j=1}^{N_h} Y_j - Y_i}{\left(\sum_{j=1}^{N_h} Y_j\right)^2} Y_i (1 - Y_i) X_k - \beta \sum_{j=1}^{N_o} (O_j - T_j) W_{ji} Y_i (1 - Y_i) X_k\end{aligned}\quad (13)$$

2.3 Entropy cycle

The entropy penalty term is added to the total cost function for some number N of iterations and then removed by setting the entropy rate α to 0 for the next N number of iterations. N is termed as the entropy cycle. The reason for including such an entropy cycle is to allow learning to take place alongside with the entropy reduction process as demonstrated in the experiments of Section 4.2.

3. CRITERIA FOR HIDDEN NODE PRUNING

Many researchers^{10, 11, 12, 13, 14} have worked on criterion that correctly reflects the relevance of hidden nodes in a neural network. Methods like sensitivity analysis¹², optimal brain damage¹⁵ and many others are expensive to compute. In this paper, two existing criteria proposed by Mozer and Smolensky¹⁰; and Kamimura and Nakanishi⁶ will be discussed and three new, relatively inexpensive criteria are proposed.

3.1 Mozer-smolensky relevance criterion¹⁰

The relevance (ρ_i) for the i th hidden node is computed by the equation:

$$\rho_i = \sum_{p=1}^{N_p} \sum_{k=1}^{N_o} (T_k^p - {}^iO_k^p)^2 - \sum_{p=1}^{N_p} \sum_{k=1}^{N_o} (T_k^p - O_k^p)^2 \quad (14)$$

where ${}^iO_k^p$ is the network output when the i th hidden node is eliminated. A high relevance value implies that the hidden node is important during the network training. Mozer-Smolensky relevance criterion requires two passes through a particular pattern and thus it is computationally intensive.

3.2 Kamimura-nakanishi variance criterion

Kamimura and Nakanishi⁶ proposed a measure that is easier to compute and strongly resembles Mozer-Smolensky relevance criterion. Their measure is to calculate the variance of input-hidden connections. They based the measure on their observation that the hidden nodes playing important roles tend to have a large variance on the incoming weights, compared with other less important hidden nodes. The variance (R_i) for the i th hidden node is defined by

$$R_i = \frac{1}{N_i - 1} \sum_{j=1}^{N_i} (V_{ij} - \bar{V}_i)^2 \quad (15)$$

where $\bar{V}_i$ is defined as:

$$\bar{V}_i = \frac{1}{N_i} \sum_{j=1}^{N_i} V_{ij} \quad (16)$$

This measure is approximately equivalent to the Mozer-Smolensky relevance criterion but its magnitude is usually low.

3.3 Hidden-output weight variance criterion

A measure similar to R_i is proposed in this paper to compute the variance of the hidden-output connections. This measure is usually higher in magnitude compared to R_i measure. The variance (HOV_i) for hidden node i is defined by:

$$HOV_i = \frac{1}{N_o - 1} \sum_{k=1}^{N_o} (W_{ki} - \bar{W}_i)^2 \quad (17)$$

where $\bar{W}_i$ is defined as:

$$\bar{W}_i = \frac{1}{N_o} \sum_{k=1}^{N_o} W_{ki} . \quad (18)$$

This measure correlates well with the Mozer-Smolensky relevance ρ_i criterion but requires at least two output nodes.

3.4 Hidden Node Activation Variance Criterion

A fourth measure is proposed to examine the variance of the i th hidden node (HV_i), which is defined as:

$$HV_i = \frac{1}{N_p - 1} \sum_{p=1}^{N_p} (Y_i^p - \bar{Y}_i)^2 \quad (19)$$

where $\bar{Y}_i$ is the mean activation value of i th hidden node across N_p pattern presentation:

$$\bar{Y}_i = \frac{1}{N_p} \sum_{p=1}^{N_p} Y_i^p \quad (20)$$

3.5 Hidden node activation differential criterion

Another criterion suitable for networks trained with the entropy learning cost function is also proposed. The temporal behaviour of pattern encoding in the hidden nodes varies across pattern presentation. This variation of the hidden nodes' activation from one pattern to the next could

serve as a good measure of relevance if the encoding has extreme representation. Basically, the hidden node's activation differential (HA_i) for the i th hidden node is defined as:

$$HA_i = \sum_{p=2}^{N_p} (Y_i^p - Y_i^{p-1}). \quad (21)$$

This measure is based on the assumption that if a hidden node varies very little in its encoding process, then it is minimal in its information content as it becomes a bias to the output node.

4. EXPERIMENTAL RESULTS

Three sets of experiments are conducted to highlight the use of entropy in pruning. These are the XOR, 3-bits parity, and Iris¹⁶ classification problems. The XOR problem is formulated as a 2-bit classification task. Output 1 will be activated for (0, 1) or (1, 0) inputs and Output 2 will be activated for (1, 1) or (0, 0) inputs. Like XOR problem, 3-bits parity problem is a 2-bit classification problem. There are 3 inputs (i.e. 3 bits) to the neural network. Output 1 will be activated when the number of bits of the input is odd number and the Output 2 will be activated when the number of bits of the input is even number. The Iris problem is to classify an input pattern into three Iris flower species. The input pattern has four features, which are the sepal length, sepal width, petal length and petal width.

In these three problems, first the behaviour of the entropy term is examined in the Back-propagation learning process using only the cross-entropy cost function, i.e. βE . Then, the proposed entropy reduction algorithm with the entropy penalty term αH is used to steer the activation levels of the hidden nodes in the network. Finally, a sequential pruning process is conducted to illustrate the number of relevant nodes remaining after entropy learning using the five criteria as detailed in Section 3.

4.1 Entropy of hidden nodes

The following experiments are conducted to show that learning is an entropy reduction process. In all the classification tasks, the entropy term H eventually decreases with training although the direction of entropy reduction may be jittering at times in the error minimisation phase. It can be seen from Figure 3 that the entropy measure decreases constantly in the Iris classification problems. This corresponds to the convergence of hidden node representations towards extreme representation direction near 0 or 1. Such a smooth decrease in entropy is due to the fact that there are not many local minima in the training process. In the other two non-linear problems, which are the XOR and 3-bits parity problems, the hidden nodes are seen to develop extreme behaviour at the beginning of learning. They are forced to change direction after encountering the saddle region in the error surface and converge towards the linear operating mode around 0.5. Upon getting out of the saddle region, the hidden nodes enter into the non-linear operating mode and converge towards an optimal solution.

(a)

(b)

(c)

Figure 3 : Hidden entropy in three classification problems. (a): XOR. (b): 3-bit parity. (c): Iris.

The entropy reflects the status of learning process. It can be seen through the entropy whether the networks are stuck in a basin of attraction. The entropy curve flattens or moves upwards when the network is searching a solution in the saddle region. The entropy decreases rapidly when the networks find the correct direction and converge. The hidden entropy thus indicates whether or not the majority of the nodes are operating in the linear region. If the majority of the nodes are operating in the linear region, then the entropy is high. If the nodes develop extreme behaviour near the flat non-linear end of the curve, then the entropy is low.

4.2 Entropy cycle

Figure 4, 5 and 6 show an instance of the contribution of the entropy cycles to the performance of the network when it encounters a local minima in the XOR problem. The XOR problem is formulated as a 2-bit classification task. Output 1 will be activated for $(0, 1)$ or $(1, 0)$ inputs and Output 2 will be activated for $(1, 1)$ or $(0, 0)$ inputs. The entropy cycles dampen the extreme minimisation of the network entropy and improve generalisation. Figure 4 shows better accuracy (generalisation) with the use of entropy cycles. Figure 5 shows that without entropy cycles, the hidden nodes go into saturation very fast without capturing the essence of the training data. In

Figure 4 : Accuracy of using with and without entropy cycle.

Figure 6, the Mean Square Error converges to 0 only when the entropy cycles are used indicating 100% classification.

Figure 5 : Number of saturated nodes of using with and without entropy cycle.

Figure 6 : Mean square error of using with and without entropy cycle.

4.3 Relevance Criteria

4.3.1 XOR problem

Fifty trials are conducted using a single layer Back-propagation neural network and starting with ten hidden nodes. The five criteria introduced in Section 3 are used to prune the ten hidden node network with and without entropy penalty term. The result of the experiment in Figure 7 shows the magnitude of the criteria and network classification accuracy with and without entropy penalty term. To interpret, Figure 7(a) shows that Node A is the least relevant node and Node J is the most relevant node. Hence the sequence of pruning the hidden nodes should be Node A, Node B,..., Node J. In Figure 7(b), it shows that if Node A is pruned, the network still

Figure 7 : Five criteria and their accuracy obtained before and after entropy penalty term is used for XOR problem. (a) & (b): Mozer-Smolensky relevance and its accuracy. (c) & (d): Kamimura-Nakanishi variance and its accuracy. (e) & (f): Hidden-Output weight variance and its accuracy. (g) & (h): Hidden node activation variance and its accuracy. (i) and (j): Hidden node activation differential and its accuracy.

achieves 100% accuracy with and without entropy penalty term. The accuracy of the network starts to drop when Node I is pruned.

Mozer-Smolensky relevance is high for relevant nodes, i.e. Node I and J, when entropy penalty term is used (Figure 7 a, b). Without entropy penalty term, only one node, i.e. Node J, is high (Figure 7 a). Kamimura-Nakanishi variance is small and it fails to capture the relevance of the nodes (Figure 7 c, d). However, better accuracy is observed when entropy penalty term is used. Hidden-output weight variance is high for more important nodes (Figure 7 e, f). Hidden node activation variance is not reflective of the importance of the hidden nodes and thus it is not a good measure for pruning (Figure 7 g, h). Relevant nodes have higher activation differential with entropy penalty term (Figure 7 i, j).

4.3.2 Three-bits parity problem

Fifty trials are conducted using a single layer Back-propagation neural network and starting with ten hidden nodes. Figure 8 shows the experimental results of using five criteria on 3-bits parity problem. The results are comparable to that of XOR problem except that Kamimura-Nakanishi variance is higher (Figure 8 c) with entropy penalty term but fails to capture the relevance of the nodes (Figure 8 d). In most cases the network is effectively pruned down to an optimal size of three hidden nodes. In particular, hidden node activation differential (Figure 8 i, j) performs well with entropy learning. The network is pruned down to three nodes as compared to four nodes without entropy learning. The magnitude of each criterion is also higher after entropy learning.

4.3.3 Iris classification problem

One hundred and fifty samples of three Iris flower species are used. Twenty-five specimens from each of the three species are used to train the network and the remaining half for the testing.

Mozer-Smolensky relevance is higher for important nodes with entropy penalty term (Figure 9 a, b). In terms of accuracy, two nodes can be retained without entropy penalty term. For the Kamimura-Nakanishi variance (Figure 9 c, d), there is an accuracy drop from pruning node F to node G for network without entropy penalty term. When entropy penalty term is used, three nodes can be retained. The accuracy remains constant while pruning from node A to F and it increases from node F to G. Kamimura-Nakanishi variance is also seen to be higher for important nodes with entropy penalty term. The ordering of the important nodes seems to line well for sequential pruning. Hidden-output weight variance (Figure 9 e, f) is higher for important nodes with entropy penalty term. The network is pruned down to three nodes as compared to five nodes without entropy penalty term. Hidden node activation variance (Figure 9 g, h) fails to capture the importance of the hidden nodes. Hidden node activation differential (Figure 9 i, j) isolates clearly the important nodes with entropy penalty term. This criterion is more effective with entropy penalty term as more nodes are forced to encode at the non-linear end of the Sigmoid function while some still persist in the linear region. A network accuracy of 96% is achieved with 3 nodes as compared to 5 nodes without entropy penalty term.

Figure 8 : Five criteria and their accuracy obtained before and after entropy penalty term is used for 3-bit parity problem. (a) & (b): Mozer-Smolensky relevance and its accuracy. (c) & (d): Kamimura-Nakanishi variance and its accuracy. (e) & (f): Hidden-Output weight variance and its accuracy. (g) & (h): Hidden node activation variance and its accuracy. (i) & (j): Hidden node activation differential and its accuracy.

Figure 9 : Five criteria and their accuracy obtained before and after entropy penalty term is used for Iris problem. (a) & (b): Mozer-Smolensky relevance and its accuracy. (c) & (d) : Kamimura-Nakanishi variance and its accuracy. (e) & (f): Hidden-Output weight variance and its accuracy. (g) & (h): Hidden node activation variance and its accuracy. (i) & (j): Hidden node activation differential and its accuracy.

5. CONCLUSIONS

The behaviour of entropy reduction learning is examined and a new training method is proposed using the additional entropy penalty term with the entropy cycle. The entropy reduction learning helps to accentuate and increase the importance of relevant nodes while dampening the less important nodes resulting in a compact encoding in the hidden layer for pruning. The selective use of the additional entropy penalty term during the entropy cycle allows hidden nodes representation to be steered without much impairment in generalisation. Various properties of the entropy learning algorithm are discussed with reference to two existing and three new proposed relevance criteria. The experimental results show that better accuracy levels can be obtained and the magnitude of each criterion is also boosted after entropy learning with entropy cycle. Hidden activation differential criterion is shown to be the most effective in pruning after entropy learning with entropy cycle, and it captures the relevant node more correctly than other relevance criteria. However, the number of remained nodes after pruning may not be the optimal in some cases, due to the fact that the activities of some neurons are correlated or equivalent.

REFERENCES

1. Muller, B. and Reinhardt, J. (1990), Springer Verlag.
2. Battiti, R.(1994), *IEEE Transactions on Neural Network*, vol. 5, no. 4, pp. 537-550.
3. Prat, L.Y.(1993), Computer Science Department, Rutgers University, ML-TR-37.
4. Deco, G., W.F. and Zimmermann, H.G. (1995), *Neural Computation*, vol. 7, pp. 86-107.
5. Burrows, T.L. and Niranjan, M. (1993), Technical report CUED/F-INFENG/TR158, Cambridge University Engineering Department.
6. Kamimura, R. and Nakanishi, S. (1995), *IEICE Trans. Inf. and Syst.*, vol. E78, pp. 484-489.
7. Kamimura, R. and Nakanishi, S. (1996), *International Conference on Neural Information Processing*, Hong Kong, pp. 145-150.
8. Kirpatrick, S., Gelatt Jr., C.D. and Vecchi, M.P. (1983), *Science*, vol. 220, pp. 671-680.
9. Ng, G.S., Erdogan, S.S. and Chan, K.H. (1996), *Fourth International Conference On Automation, Robotics and Computer Vision*, pp. 2472-2477.
10. Mozer, M.C. and Smolensky, P. (1989), *Connection Science*, vol. 1, no. 1, pp. 3-16.
11. Karnin, E.D. (1990), *IEEE Transactions on Neural Networks*, vol. 1, pp. 239-242.
12. Ruck, D.W., Rogers, S.K., Kabrisky, M., Priddy, K. L., and Tarr, G. L. (1992), Technical Report No 92-92, Department of Electrical and Computer Engineering, Air Force Institute of Technology.
13. Chung, F. L. and Lee, T. (1992), *International Journal of Neural Systems*, vol. 3, pp. 301-314.
14. Shamir, N., Saad, D., and Marom, E. (1993), *International Journal of Neural Systems*, vol. 4, pp. 143-158.
15. Le, C. Y., Denker, J.S. and Solla, S.A. (1990), Edited by Touretzky, D.S., Morgan Kaufmann, San Mateo, pp. 598-605.
16. Fisher, R.A. (1936), *Annals of Eugenics*, vol. 7, pp. 179-188.