Optimization of Neural Network Topology and Information Content Using Boltzmann Methods

C L. Wilson, National Institute of Standards and Technology
Gaithersburg, MD 20899
O. M. Omidvar, University of the District of Columbia
Washington, DC 20008

Abstract

Reduction in the size and complexity of neural network is essential to improve generalization. reduce training error. and improve network speed. Most of the known optimization methods heavily rely on weight sharing concepts for pattern separation and recognition. The method presented here focuses on network topology and information content for optimization. We have studied the change in the network topology and its effects on information content dynamically during the optimization of the network. The changes in the network topology were achieved by altering the number of nonzero weights. The primary optimization is scaled conjugate gradient and the secondary method of optimization a Boltzmann method. The conjugate gradient optimization serves as a connection creation operator and the Boltzmann method serves as a competitive connection annihilation operator. By combining these two methods its is possible to generate small networks which have similar testing and training accuracy. good generalization, from small training sets. Our findings demonstrate that for a difficult character recognition problem the number of weights in a fully connected network can be reduced by over 90%.

1 Introduction

The size and the complexity of neural network applications has grown rapidly. The search for small networks with large information content and generalization capability is ongoing. Most of the optimization strategies are a trade-off between error and network complexity. The known optimization schemes[1,2,3] have used this trade-off to minimize the cost function. Among various complexity measures, Vapnic-Chervonenkis (VC) dimensionality [4], concentrates on information content and distribution of information in the network. The error term associated with increasing VC dimension can be reduced by greatly expanding the size of training set or by reducing the VC dimension of the network.

Boltzmann methods have been used as a statistical method for combinatorial optimization and for the design of learning algorithms[5,6]. This method can be used in conjunction with a supervised learning method to dynamically reduce network size. The strategy used in this research is to remove the weights using Boltzmann criteria during the training process. Information content is used as a measure of network complexity for evaluation of the resulting network.

The competing mechanisms involved when the Boltzmann method is used in conjunction with SCG are shown in table 1. This table lists five points where these two methods can be compared. The Boltzmann method is self-organizing while the SCG method is a supervised learning method. The Boltzmann method seeks to minimize the the number of weights while maintaining the information content of the network. The SCG method seeks to minimize an error function on the training set. The important controlling parameter for the Boltzmann method is the information in the network is the iteration time, t, as $t \to \infty$. The controlling informational parameter for the SCG method is the information provided at $t = 0$ in the initial weights. The algorithmic control in the Boltzmann method is the temperature sequence applied during the iteration. The equivalent controlling parameter for the SCG method is the restart sequence.

Boltzmann Method	SCG Method
Self-Organization	Supervised Learning
information minimization	error optimization
generalization in testing	error in training
Info ($t \to \infty$)	Info ($t = 0$)
Temperature sequence	Restart sequence

Table 1: Competing mechanisms when Boltzmann and SCG methods are combined for concurrent network optimization

2 Pruning via Boltzmann Methods

In this paper a fully connected network is optimized using the Scaled Conjugate Gradient method (SCG) developed by Moller [7] and modified by Blue and Grother [8]. The SCG method is used as a starting network for the Boltzmann weight pruning algorithm. The network has an input layer with thirty-two input nodes, a variable size hidden layer with sixteen, thirty-two or sixty-four nodes and an output layer with ten nodes. The initial network is a fully connected network. The pruning was carried out by selecting a normalized temperature, T, and removing weights based on a probability of removal:

$$P_i = \exp(-|w_i|/T)$$

The values of P_i are compared to a set of uniformly distributed random numbers, R_i, on the interval $[0,1]$. If the probability P_i is greater than R_i then the weight is set to zero. The process is carried out for each iteration of the SCG optimization process and is dynamic. If a weight is removed it may subsequently be restored by the SCG algorithm; the restored weight may survive if it has sufficient magnitude in subsequent iterations.

The dynamic effect of this is shown in figure 1 for five temperatures between 0.1 and 0.5 at 0.1 intervals, starting from a fully converged and fully connected network. As the size of the temperature change increases the number of weights removed initially increases, but the effect of later iterations of optimization and pruning is to decrease the rate at which weights are removed. The number of weights in the initial network was 1386, including bias weights. At all temperatures the initial iterations are very effective in reducing the weights. The decrease in the rate of pruning is the result of a critical phenomena characterized by a critical temperature, T_c, at which the new information added by the SCG training balances the information removed by pruning. At this critical point networks trained on small training sets will achieve identical testing and training accuracy even when tested on large test sets.

The effect of the number of hidden nodes can be seen in figures 2, 3 and 4. Figure 2 shows the effect on the network with 32 hidden nodes used in figure 1. As the temperature is increased the accuracy of the network for recognition decreases slowly for temperatures up to 0.4. As the temperature approaches 0.5 the rate of weight removal shown in figure 1 slows and the rate of accuracy decay accelerates. The two curves plotted are the training set and testing set accuracy of the network. The training set accuracy is initially greater than the testing accuracy. At a critical temperature, T_c, the testing accuracy and training accuracy are identical. In figure 2, at the critical temperature of 0.58, read from figure 2, chaotic behavior sets in the vicinity of T_c due to a critical effects of weight removal. The behavior of the 32-64-10 network in figure 4 is similar to the 32-32-10 network. The 32-16-10 network in figure 3 shows an increase in temperature, T_c, and a decrease in accuracy at T_c. This increase in T_c is caused by the reduced set of possible pruned configurations in the 32-16-10 network; the initial 32-16-10 network is too small.

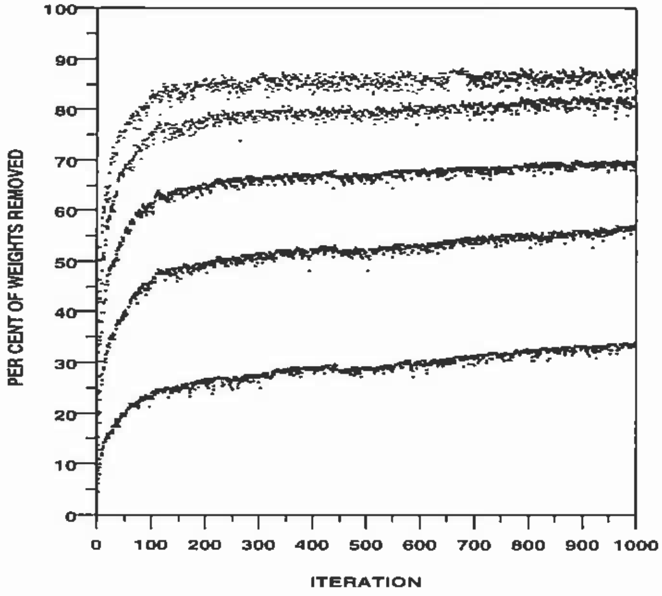

Figure 1: Weights removed as a function of iteration and temperature for $T = 0.1, 0.2, 0.3, 0.4, 0.5$. The lower curve is for $T = 0.1$; the upper curve is for $T = 0.5$.

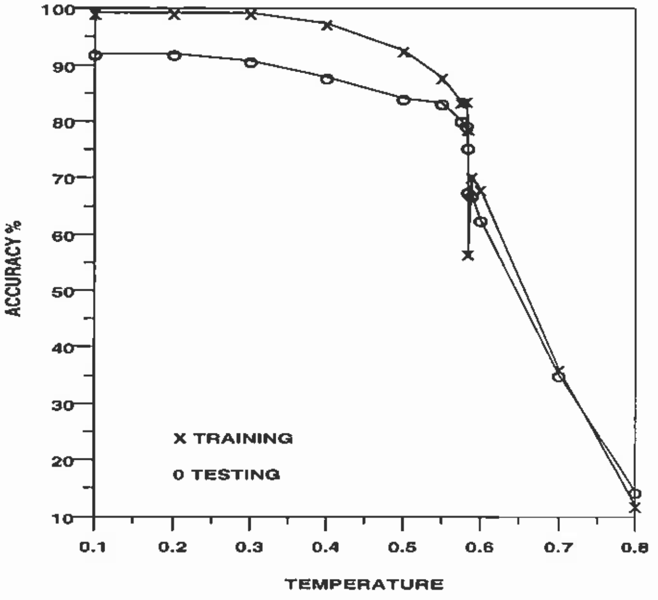

Figure 2: Change in testing and training accuracy as a function of temperature for a 32-32-10 network after 1000 iterations at each temperature.

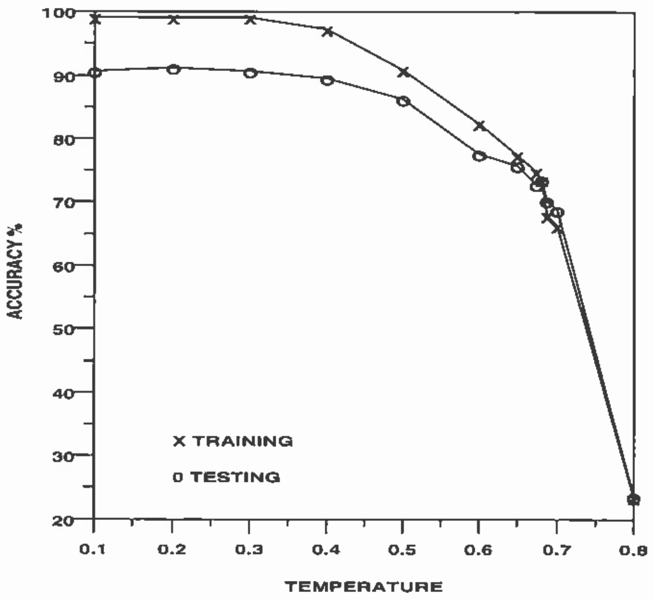

Figure 3: Change in testing and training accuracy as a function of temperature for a 32-16-10 network after 1000 iterations at each temperature.

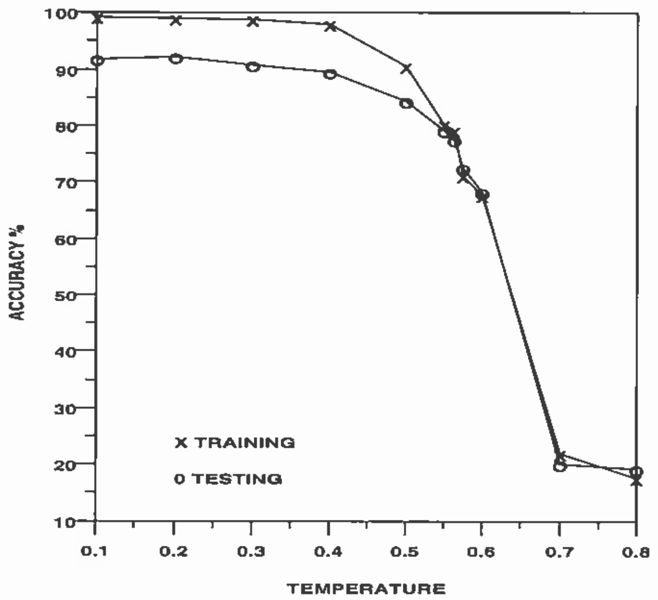

Figure 4: Change in testing and training accuracy as a function of temperature for a 32-64-10 network after 1000 iterations at each temperature.

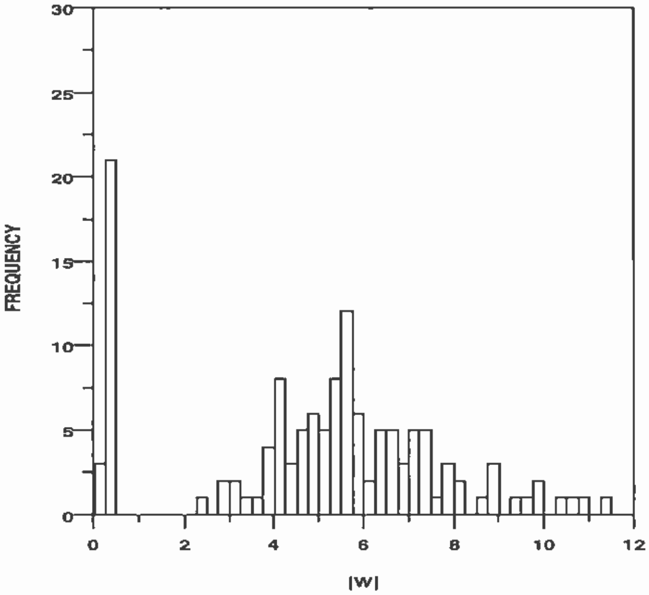

Figure 5: Weight distribution below T_c at $T = 0.55$.

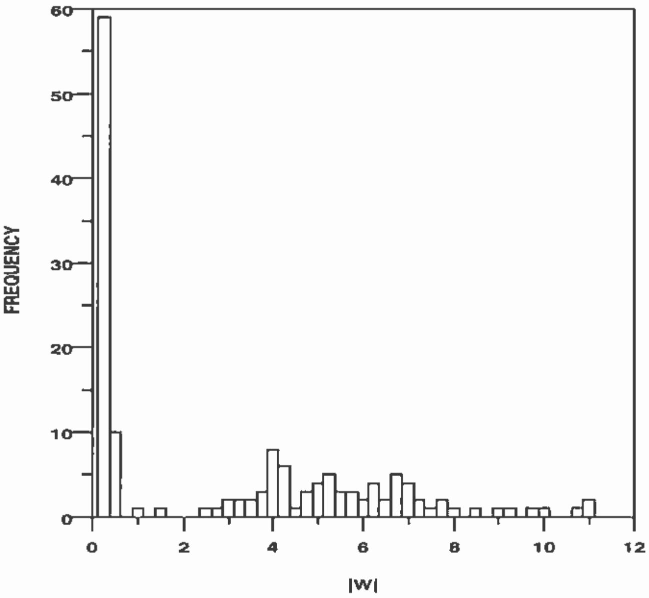

Figure 6: Weight distribution above T_c at $T = 0.6$

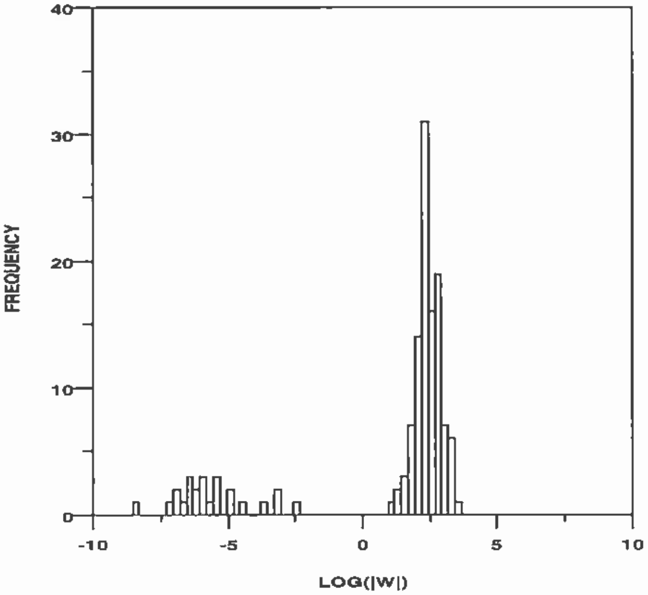

Figure 7: Information in weights. $\sum \log_2(|w_i|)$. below T_c at $T = 0.55$.

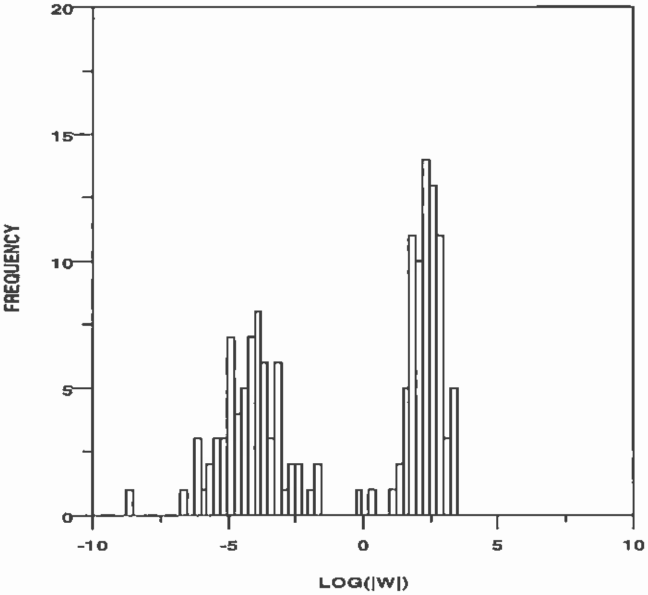

Figure 8: Information in weights. $\sum \log_2(|w_i|)$. above T_c at $T = 0.6$.

3 Weight Reduction and Information Content

The effect on the information content of the network can be evaluated by examining the the distribution of weights in the network as a function of temperature. Figure 5 shows the distribution of the absolute value of the weights at a temperature near, but below, T_c. Figure 6 shows the distribution of the absolute value of the weights at a temperature near, but above, T_c.

These distributions illustrate the mechanism involved in the collapse of testing and training accuracy near T_c. The accuracy collapse is caused by the large increase in weights near zero created by the most recent SCG iteration. In a given training cycle some weights are removed. If these weights are redundant they will be compensated for by other weights in the network. If these weights are critical they will be restored by the SCG optimization. The peak in the distribution near zero in both figures 5 and 6 is caused by this process. At T_c the SCG creation process is just balanced by the Boltzmann pruning.

The effect of the near-zero weights is more important when viewed as information content. The VC dimension and the information content are both approximately $\sum(\log_2(|w_i|) + 1)$. A weight distributions of this kind are shown in figures 7 and 8 for T above and below T_c. When large numbers of near-zero weights exist, their contribution to the sum dominates the network information. Under these conditions the network is dominated by recently created weights which have not been optimized by SCG iterations. This lowers network accuracy without reducing VC dimension.

To evaluate the generalization capability of the pruned network the network associated with a temperature $T = 0.55$ was tested on a sample of 221,000 digits [9]. The predicted accuracy from T_c data was 75.5%; the accuracy achieved in the test was 72.6%. In this region the change in accuracy of the network is about 5% for each ΔT of 0.001 so that this agreement is consistent with an accuracy of T_c of ±.0005 with a value of $T_c = 0.582$.

4 Conclusions

A method of network optimization has been developed which reduces by 80% to 90% the number of weights required for moderately accurate character recognition. The method is based on achieving equilibrium between the information in the training set and the number of network weights by concurrent weight creation by SCG optimization and Boltzmann weight removal. These reductions allow both smaller training sets and smaller classification networks to be used.

Acknowledgement

The authors would like to acknowledge Jon Geist for suggesting the analysis of the distribution of weights near T_c and Jim Blue for modifications to the SCG program.

References

[1] E. B. Baum and D. Haussler, "What size net gives valid generalization?", *Neural Computation*, **1**, pp. 151-160, 1989.

[2] M. C. Mozer and P. Smolensky, "Using relevance to reduce network size automatically", *Connection Science*, **1**, pp. 3-16, 1989.

[3] Y. Le Cun, J. S. Denker and S. A. Solla, "Optimal Brain Damage", in D. S. Touretzky, editor, Advances in Neural Information Processing System **2**, pp. 396-404, Morgan Kauffman, 1990.

[4] I. Guyon, V. N. Vipnick, B. E. Boser, L. Y. Botton, and S. A. Solla, "Structural Risk Minimization for Character Recognition", in R. Lippmann, editor, Advances in Neural Information Processing System **4**, Morgan Kauffman, 1992.

[5] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for Boltzmann machines", *Cognitive Science*, **9**, pp. 147-169.1985.

[6] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vacchi, "Optimization by simulated annealing", *Science*, **220**, pp. 671- 680, 1983.

[7] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning", *Neural Networks*, to be published.

[8] J. L. Blue and P. J. Grother, "Training feed-forward neural networks using conjugate gradients," *SPIE*, Character Recognition and Digitizer Technologies, San Jose, Feb, 9-14, 1992.

[9] M. D. Garris, C. L. Wilson, J. L. Blue, G. T. Candela, P. Grother, S. Janet, and R. A. Wilkinson, "Massively parallel implementations of character recognition systems," *SPIE*, Character Recognition and Digitizer Technologies, San Jose, Feb, 9-14, 1992.