China 1991 International Conference on Circuits and Systems, June 1991, Shenzhen, China

AN ALGORITHM FOR DYNAMICALLY ADAPTING NEURAL NETWORK TOPOLOGIES

F. Piazza, M. Marchesi, G. Orlandi (*), A. Uncini

Dip. Elettronica ed Automatica - Università di Ancona,
Via Brecce Bianche, 60131 Ancona, Italy

(*) Dip. INFOCOM, Università di Roma "La Sapienza",
Via Eudossiana 18, 00184 Roma, Italy

ABSTRACT

Recently, it has been proposed that the biological networks change not only the synaptic strengths of connection but also partially their internal topologies, according to either the received external stimuli and the pre-existent connection layouts. Following this idea, in the paper a method is presented to dynamically adapt the topology of neural networks with supervised learning, using only the information of the training set. The method eliminates connections from an initial fully connected network, concurrently with the learning algorithm. Several experimental results obtained with multilayered networks are also reported in the paper to demonstrate the capabilities of the proposed method.

1. Introduction

It is well known that Artificial Neural Networks can solve a large variety of problems. In order to reach this goal, usually a formal model of the neuron is chosen and used to build a network with a given topology. Then this network is trained, adapting the synaptic weights of every neuron, by using a suitable learning algorithm.

The network topologies which are extensively used, are chosen among a set of known fixed models that are characterized by the presence of dense connection matrices among all neurons (fully connected network) or among groups of neurons, as in the well known MultiLayer Perceptron (MLP) [1]. These topologies however are different from those found in biological neural systems, where the networks are often much less regular. Moreover, since a particular problem may be optimally solved by a network with a non-regular connection topology, the task of transforming an oversized regularly connected network into such optimal non-regular network is completely committed to the learning algorithm. Most of known learning algorithms, however, are not able to produce a reduced topology by zeroing some synaptic weights, instead they typically produce different structures spreading nonvanishing weights all over the network. The efficiency can result poor and several dimensions of a chosen configuration must be tried to yield an acceptable solution. In fact, if the network is too small, the input-output mapping cannot be learned with satisfactory accuracy. Conversely, if the network is too large, after a long training phase it will learn the given set correctly, but will badly generalize due to learning data overfitting [2]. Therefore, the goal when training a network is to find a topology large enough to learn the mapping and as small as possible to generalize correctly.

Two possible approach to produce networks with correct topologies have been proposed:

1) start with a small network and grow additional synapses and/or neurons until the desired behaviour is reached;
2) start with a large network and prune off synapses and/or neurons until the desired behaviour is retained.

The first approach, although it could be efficient since most of the job is performed on networks of small size, can be difficult to realize and control in order to reach network topologies which are "optimal" in some senses (see for example [3]).

The second approach is much more studied and consists of finding a subset of network synaptic weights that, when set to zero, lead to the smallest increase of an error measure at the output. Several methods has been proposed in the last years in particular for the MLP [4,5,6,7,8]. However they can be slow, requiring periodic retraining of the network, and often require a "supervisor", a non-local algorithm which performs the pruning task. Moreover, some of them can require a difficult fine-tuning of the pruning coefficients in order to work correctly.

Therefore it is highly desiderable to find new methods which are able to dynamically adapt the network topology to the problem. In this paper, using some observations on the biological network plasticity, an idea is introduced to develop a procedure for varying the network topology by simply pruning off connections in an initial overdimensioned fully connected network.

This work was supported in part by the C.N.R. of Italy under the project "Sistemi Elettronici Avanzati - Reti Neurali" and in part by the Ministero dell'Univ. e della Ricerca Scientifica e Tecnologica of Italy.

2. The Proposed Method

The plasticity of brain, i. e. the capability of changing neural connections over time, is an extensively studied phenomenon. In this field the work of Hubel and Wiesel on measuring changes in the ocular dominance statistics of a large neuron sample of the visual cortex in mammalians (particularly cat and monkey) is a fundamental milestone. They found that monocular deprivation drives most neurons of the cortex to prefer the open eye, provided that the animal is still sufficiently young [9]. In particular it is now clear that the plasticity varies over the course of ontogenetic development and reaches a sharp maximum during the so-called critical period, during which a strong change of the brain structure is driven by external environmental stimuli.

Recently [10], it has been proposed that, during the critical period, the network changes not only the synaptic strengths of connection but also partially its internal topology. These changes are driven by either the received external stimuli and the pre-existent connection layout.

We have found that this idea can be applied also to simple artificial neural networks with supervised learning. By using only the external stimuli (i. e. input and output patterns) and the knowledge of the synaptic weights it is possible to adapt the network to the problem, changing concurrently the network synapses and the connection topology.

A simple procedure to prune off connection from a densely connected network is presented here. No new connections are created, although a similar approach could be followed to derive a procedure to add synapses to the network. In order to verify if a particular connection is necessary or not to solve a particular problem, a measure of the importance of the connection is necessary. According to the biological behaviour, this measure must be proportional either to the external stimuli and to pre-existent synaptic strength. A very simple formula can be derived by defining the following quantity:

the synaptic activity of the connection of the i-th neuron coming from the j-th neuron and relative to the p-th training pattern

$$a_{ij}(p) = [\, w_{ij}\ Act_j(p) \,]^2 \qquad (1)$$

where w_{ij} is the synaptic weight of the connection between the i-th and j-th neurons, and $Act_j(p)$ is the activation of the j-th neuron corresponding to the p-th input pattern. This activity can be averaged on the whole training set

$$\hat{a}_{ij} = \frac{\sum_{p=1}^{N_p} a_{ij}(p)}{N_p} \qquad (2)$$

where N_p is the number of the input training patterns. Since each neuron is driven by several different synapses, an average total synaptic activity relative to the i-th neuron can be defined

$$A_i = \sum_{j=1}^{N_i} \hat{a}_{ij} \qquad (3)$$

where N_i is the total number of inputs to the i-th neuron. A measure of importance of the synapsis can be obtained by comparing its averaged activity with respect to the the total average activity relative to the neuron which the synapsis belongs to. Therefore the Percentage Average Synaptic Activity (PASA) can be defined as follows

$$PASA_{ij} = \frac{100\ \hat{a}_{ij}}{A_i} \qquad (4)$$

3. The Pruning Algorithm

During the learning phase of the network, the average synaptic activity and the PASA of each connection change with the learning epoches. These quantities vary until the network has learnt the problem, then they stabilize. The idea for pruning the network is to compare the PASA of each connection with a threshold which dinamically changes with the learning epoches.

The proposed algorithm is therefore concurrent with the training algorithm and can be summarized by the following steps:

a) during every input-output pair presentation, expression (1) is computed for each connection, and the result is accumulated according to expression (2);
b) at the end of each learning epoch (i.e. after the whole learning set presentation), expression (3) is computed for each neuron, and the PASA's of each connection are derived by expression (4);
c) for each connection a comparison between the corresponding PASA and a threshold function Th(ep) (ep is the epoch index) is performed and a decision is taken:

$$\begin{cases} PASA_{ij} < Th(ep) & \text{connection i,j pruned} \\ PASA_{ij} \geq Th(ep) & \text{connection i,j retained} \end{cases} \qquad (5)$$

It can be noted that this algorithm is local in the sense that each neuron requires only local quantities to compute the PASA's. Moreover, the computational cost is low, since:

- the calculation of the synaptic activity per training pattern requires one multiplication/accumulation operation per synapsis (the multiplication can be avoided by using the absolute value instead of the square value in (1));
- the calculation of the PASA's of each connection relative to the i-th neuron at the end of one training set requires (N_i-1) sums, one division and one multiplication per synapsis. The comparison between the PASA's and the threshold can be easily arranged in order to further reduce the

computational burden.
However, the comparison (5) has the drawback that the pruning effect is dependent of the number of synapses of the neuron. In fact more connections belongs to a neuron, smaller will be their PASA's. In an ideal network trained with a complete well-balanced learning set, the percentage average synaptic activities of the N_i connections of the i-th neuron should be comparable and very close to the value $(100/N_i)$. Therefore, an alternative approach for pruning is to compare the PASA's with the ideal value $(100/N_i)$ and prune those connections whose activities are smaller enough with respect to it. At the ep-th epoch:

$$\begin{cases} \dfrac{(100/N_i)-PASA_{ij}}{(100/N_i)} > Th(ep) & \text{connection pruned} \\ \dfrac{(100/N_i)-PASA_{ij}}{(100/N_i)} \leq Th(ep) & \text{connection retained} \end{cases} \quad (6)$$

4. Experimental Results

The proposed method has been tested with fully connected recurrent networks as well as with the Multilayer Perceptron (MLP) model [1]. For the sake of brevity only the results of several simple experiments using MLP networks are reported here. In all cases, the PASA's are averaged on 2 learning epoches, although this was seen to be not critical. The threshold function used in comparison (5) is the gaussian function:

$$Th(ep) = v\ e^{-\frac{1}{2}\left(\frac{ep - m}{\sigma}\right)^2} \quad (7)$$

where "v","m" and σ are constant values, and the factor 100 has been removed. Moreover, a simple procedure has been added to the pruning algorithm which allows in a layer to prune off an entire neuron if it is not connected to any neuron of the upper layer.
In the first experiment, networks with 2 inputs, 4 hidden neurons and 1 output have been used to classify 100 points equally distributed inside and outside a unit circle on the X-Y plane. The high output classifies points inside the circle, while the null output classifies points outside it. 50 networks with different, randomly selected initial weights have been generated and used in the experiment. Each network was trained over 300 epoches using the 100 points as learning set.
The various simulations have shown that the best results are obtained when the gaussian threshold function is centred on a value positioned after few tens of epoches but before the complete convergence has been reached. The values of "m" and σ parameters are therefore not critical providing that the network is not disturbed during the early and late phases of learning. Typical values in this example range from 10 to 60 for "m" and from 4 to 8 for σ. The "v" value depends also on the size of the network, i.e. on the number of synapses per neuron, and it is not particularly critical. Notice that selecting m and s small allows to greatly reduce the computational burden since most of the learning is performed on a reduced network.
In the second experiment, networks with 2 inputs, 8 hidden neurons and 1 output have been used to classify points inside a unit circle in the square plane region with corners [-3,-3; -3,3; 3,3; 3,-3]. A regular grid of 500 points was used as learning set, while 2000 points, randomly selected inside the region, were used as test set. Fig. 1 shows the original network (a) and a network (b) obtained by the algorithm in a typical case (v=0.1, m=30, σ=4, see Fig. 2). About 62.5% of connections have been pruned off (15/24) and the network still continue to converge. Fig. 3 reports the Mean-Square-Error (MSE) at the output, computed over the whole training set, versus the learning epoch, respectively for the complete network (a) and for the pruned network (b). Notice that the pruning period does not disturb the convergence process as shown by the smoothness of the MSE curve in the (b) case. After many epoches, however, both MSE and maximum error of the complete network are often slightly better than those of the pruned network, due to the larger number of parameters of the first network with respect to the second.
A simple generalization experiment has been also performed. Table 1 reports the MSE and maximum error values obtained by the 2 networks after 150 training epoches, using the test set as input. Although in this case the redundancy of the complete network is little, the pruned network shows similar or better performance on the test set with respect to the complete network, using a number of synapses reduced to the 37.5% of the original.

Table 1 MSE and maximum output error obtained by the networks of Fig. 1 with the training and test set. The number of wrong classifications is also reported.

	training set		test set (2000 points)		
	MSE	MaxErr	MSE	MaxErr	failed
Pruned 62.5 %	0.0602	0.4293	0.0662	0.99	174
Pruned 0%	0.0543	0.506	0.0655	0.99	169

References

[1] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, Parallel Distributed Proc. (PDP): Exploration in the Microstructure of Cognition (Vol. 1), MIT Press, 1986.
[2] E.B. Baum, D. Haussler, "What size net gives valid generalization?", Neural Computation, No.1, 1989, pp.151-160.
[3] T. Ash , "Dynamic node creation in backpropagation networks", ICS Report 8901, UCSD Feb. 1989.
[4] J. Sietsma and R.J.F. Dow, "Neural net pruning - Why and how?", in Proc. IEEE Int. Conf. on Neural Network, vol. 1, San Diego, CA, 1988, pp. 325-332.

[5] S.J. Hanson and L.Y. Pratt, "Comparing biases for minimal network construction will back-propagation", in Advances in Neural Inf. Proc. 1, D.S. Tourezky, Ed. Kaufman, 1989, pp. 177-185.
[6] M.C. Mozer and P. Smolensky, "Skeletonization: A technique for trimming the fat from a network via relevance assessment", in Advances in Neural Information Proc. 1, D.S. Tourezky, Ed. Morgan Kaufman, 1989, pp. 107-115.
[7] E.D. Karin, "A Simple Procedure for Pruning Back-Propagation Trained Neural Networks", IEEE Trans. on Neural Networks,Vol. 1 No. 2, June 1990.
[8] Y. Le Cunn, J.S. Denker, S.A. Solla,"Optimal Brain Damage", Computer and System Science.
[9] D.H. Hubel, T.N. Wiesel, "The period of susceptibility to the physiological effects of unilateral eye closure in kittens",Journal of Physiology, No. 206, pp. 419-436.
[10] C. Aoki, P. Siekevitz, "Ontogenetic changes in the Cyclic Adenosine 3',5'-Monophosphate-stimulatable phosphorylation of cat visual cortex proteins, particularly of microtubule-associated protein 2 (MAP2): effects of normal and dark rearing and of the exposure to light", Journal of Neuroscience, No. 9-5, Set. 1985.

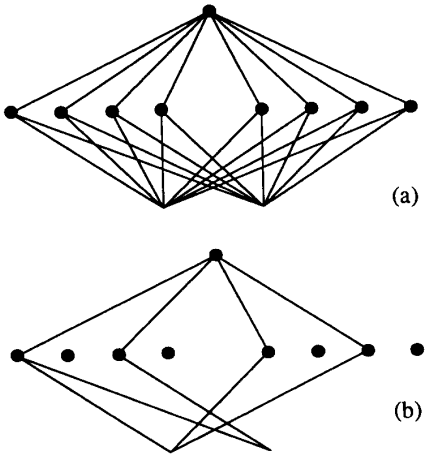


Fig.1 (a) Complete network; (b) network obtained by the proposed method with v=0.1, m=30, σ=4.

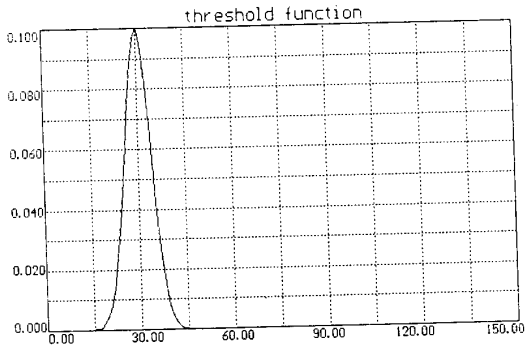
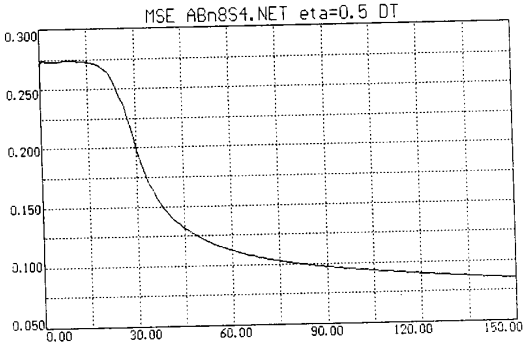
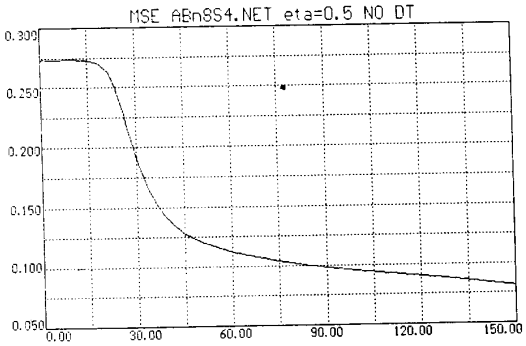


Fig.2 Plot of the threshold function used in fig. 1b vs the number of epoches (ep).



(a)



(b)

Fig.3 Plot of the MSE vs the number of learning epoches obtained (a) by the network of fig. 1a and (b) by the network of fig. 1b.