Published in the Proceedings of the
International Symposium on Artificial Neural Networks
National Chiao-Tung University, Hsinchu, Taiwan, ROC, December 18-20, 1995

Evaluating Pruning Methods

Georg Thimm and Emile Fiesler
IDIAP, P.O. 592, CH-1920 Martigny, Switzerland.
Tel: ++41 26 22 76 64, fax: ++41 26 22 78 18,
email: Georg.Thimm@idiap.ch

Abstract: A notorious problem in the application of neural networks is to find a small suitable topology. High order perceptrons already solve a part of this problem as they require no hidden layers. However, the number of connections in a fully interlayer connected high order perceptron grows quickly with their order. Partially connected topologies are therefore highly desirable and can be obtained by applying em connection pruning methods

A framework is provided here that allows a practical comparison of pruning methods which is based on final network size and generalization capability, but also considers the total training time. This framework is applied to the comparison of an easy to implement, low complexity method with four other pruning methods of similar complexity.

Keywords: pruning, generalization, optimality criteria, high order perceptrons, backpropagation neural networks.

Introduction

A mayor problem in the application of neural networks is the choice of a *topology*: a considerable amount of architectures and methods for the construction of *optimal* neural networks have been proposed. Some of those approaches try to improve well-known architectures by training a network which is expected to be big enough to solve the given problem and subsequently remove (*prune*) units.

As no pruning method is guaranteed to find the best neural network, different pruning heuristics have to be compared. This comparison of connection pruning methods for neural networks is a problem in itself, as existing optimality criteria may not match the needs of applications. A *minimal network topology* can be defined, but is rarely obtainable for a specific application [Fiesler-93]. Another important criterion, *good generalization*, is, besides not properly defined, also application dependent. In general, tradeoffs between *training time*, network size, and generalization performance should be taken into account when comparing neural network training methods. Hence, a comparative study of pruning methods has to take into account also the necesary training time for which optimality criteria need to be established.

Eventhough this publication is focused on high order perceptrons[1], as they simplify the choice of the initial topology by having only one choice of interlayer connections, the brought up questions, problems, and proposed solutions are also valid for other neural networks.

Connection Removal

A pruning algorithm basically has to do two things: to decide *when* to prune and *which unit(s)* to prune. In the experiments performed in this research, a network is pruned whenever its training converged. The unit to remove is selected by a one of five heuristics, all designed to minimize the error induced by the removal of the unit.

The first method is the in this publication proposed *smallest contribution variance* method $min(\sigma)$. This method removes connections having the smallest contribution variance on the training set, where the

[1] See [Thimm-94] for details on high order perceptrons and the benchmark data sets used in the simulations.

contribution of a connection is the value available to the connection from the lower layer, multiplied by its weight. The contribution variance is calculated over all training patterns. This method is motivated by the observation that a connection is unimportant if it has nearly the same output for the whole training set and therefore mostly acts like an additional bias. As such an 'additional bias' (the mean contribution of a connection) differs from zero, this value is added to the bias of the neuron from which the connection is removed. This method is applicable to most feedforward neural networks, especially multilayer perceptrons[2].

For comparison, four other weight removal heuristics were chosen based on their low computational complexity and their applicability to high order perceptrons.

1) The simplest connection pruning method $min(\mathbf{w})$ removes the smallest weights. The increased error is reduced by adding the *mean contribution* of this connection to the corresponding bias.

2) E. D. Karnin estimates the sensitivity of a network to the removal of a weight by monitoring the sum of all weights changes during training [Karnin-90].

3) The weight removal method of M. C. Mozer and P. Smolensky estimates the error induced by the removal of a connection based on a manipulation of the *objective function* (the function to be minimized by backpropagation) [Mozer-89].

4) W. Finnoff *et al.* use a test statistic for the probability that a weight becomes zero, which is used in a pruning algorithm called *autoprune* [Finnoff-93].

Minimal Network Size and Generalization

The pruning of a neural network is usually motivated by two aims: to obtain networks of a small size and/or with a good generalization performance. Pruning methods are therefore usually compared by means of the average final network size and/or their average generalization performance.

However, in practical applications the total training time also plays a crucial role: the necessary training time may or may not allow the performance of several training sessions. If only one or a few training sessions are possible, the pruning method should nearly never produce *bad* networks. If many repetitions are possible, it does not harm if a mayor number of solutions is unacceptable, as far as a few networks are very good. This implies, that the average network size, respectively average generalization performance, may be misleading and the distribution of the network size, respectively generalization performance, has to be taken into account. The following performance measures are therefore proposed: the 10% (90%) limit, which is the maximal network size reached in at least 10% (90%) of the simulations per experiment. Method A is judged better than method B, if method A has a smaller 10% (90%) limit, or the methods have the same 10% (90%) limit and method B produces less networks than method A of this maximal size. Similar criteria can be easily formulated for the generalization.

An example justifying this consideration is displayed in figure 1 (the horizontal axis shows the number of connections in the pruned networks; the vertical axis the percentage of simulations with final networks of this size). Both, $min(\mathbf{w})$ method and the $min(\sigma)$ method produce networks of a mean size of 46 connections[3]. Nevertheless, a large difference the two pruning methods is observable: the smallest networks produced by the $min(\mathbf{w})$ method have 40 connections, those produced by the $min(\sigma)$ method have 30 connections (each experiment consists of 100 simulations).

Over-fitting of neural networks with a static topology is usually prevented by *early stopping* [Weigend-90]. Some pruning algorithms try to adopt this: if the generalization of the network decreases steadily (on average), it is assumed, that further pruning steps does not lead to better networks.

Experiments show that this assumption is not true. The generalization performance in 10% of the simulations decreases monotonically for more than 20 and up to 200 steps at a time before the network with the best generalization is reached.

The unpredictable behavior in generalization performance implies for practical applications, which require good generalization capabilities, that the networks have to be pruned as far as possible and that non-minimal sized networks have to be considered.

[2] The efficiency of this method for multilayer perceptrons is currently evaluated.

[3] The initial network is a second order perceptron, trained on the monks 2 data set.

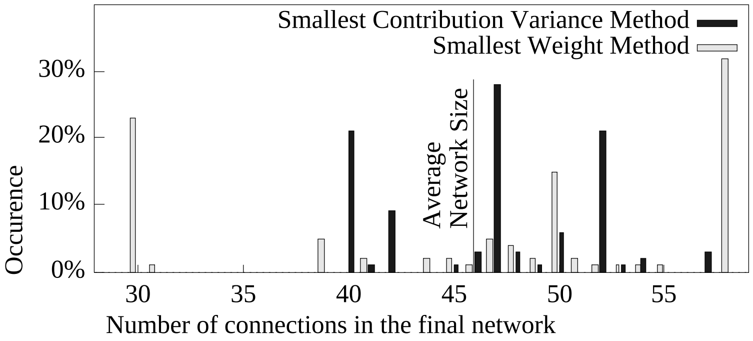

Figure 1: final network sizes of pruned networks

Experimental Results for High Order Perceptrons

The results reported below are based on experiments using 9 data sets. High order perceptrons of different order are applied to some of these data sets, which gives 19 series of experiments of together $\approx$ 20.000 simulations.

The experiments show that the $min(\mathbf{w})$ method and the heuristic used in the *autoprune* algorithm of W. Finnoff perform nearly identically for both network size and generalization. The latter can therefore be neglected as it is computationally more expensive. This might be the case if the networks are pruned before the network converged, as for example in the *autoprune* algorithm.

The best pruning method can differ for the 10% and the 90% criteria for the limit network size and generalization criteria (observed in about 20% of the experiments), justifying the application of these criteria for applications with high demands.

The method of E. D. Karnin and the method of M. C. Mozer *et al.* are less efficient in regard to the network size than the $min(\mathbf{w})$ or the $min(\sigma)$ pruning method, independent of the measure used. Only in a few cases these methods produce networks of a size comparable to the $min(\mathbf{w})$ and the $min(\sigma)$ method, but often the final network size is two or more times larger. The difference between the $min(\mathbf{w})$ and the $min(\sigma)$ method is less significant with a maximal difference of 50%. The number of experiments where one of these methods is superior to the other are equal.

The average generalization performance over all experiments of the networks produced by all five pruning are comparable, but differences in performance on specific data sets are rather big. This is true independently whether the 10%, the 90% criteria is applied, or the generalization of the smallest network or the network with the best generalization per simulation was chosen.

The common belief that a minimal network size necessarily implies a better generalization performance can not be confirmed independent from the pruning method and data set. For some experiments the best generalization per simulation was never observed for the smallest network. A similar observation was made by L. Prechelt for other pruning methods applied on multilayer perceptrons [Prechelt-95].

Conclusions

The commonly used average performance criteria applied to pruning methods can be misleading and are not always appropriate for applications, as they do not take into account whether a training session can be repeated or not. The proposed criteria do, and the difference for some applications is shown.

The outcome of the experiments justifies the usage of different pruning methods, even if the average performances of these methods are equal. For high order perceptrons the set of pruning methods to be considered consist of the $min(\mathbf{w})$ and the $min(\sigma)$ pruning method, as these methods produce often smaller networks than the methods of M. C. Mozer *et al.* and E. D. Karnin. These first two methods also have the advantage of a simpler implementation and a smaller demand of CPU and memory.

The pruning of neural networks does not always have the usually assumed positive influence on the generalization performance. If a network with a very good generalization performance is required, networks of a non-optimal size have to be considered.

A preference for one of the five pruning methods can not be established, but for specific data sets, where the differences are sometimes remarkable. Consequently, if neural networks (high order perceptrons) with a minimal size and a good generalization are required, several training sessions using different pruning methods are inevitable.

References

[Finnoff-93] W. Finnoff, F. Hergert, and H. G. Zimmermann. Improving model selection by nonconvergent methods. *Neural Networks*, num. 6, pp. 771–783, 1993.

[Fiesler-93] E. Fiesler. Minimal and high order neural network topologies. *Proceedings of the Fifth Workshop on Neural Networks*, pp. 173–178, San Diego, California, 1993. Simulation Councils, Inc. / The Society for Computer Simulation.

[Karnin-90] E. D. Karnin. A simple procedure for pruning back-propagation trained neural networks. *IEEE Transactions on Neural Networks*, vol. 1, num. 2, pp. 239–242, 1990.

[Mozer-89] M. C. Mozer and P. Smolensky. Using relevance to reduce network size automatically. *Connection Science*, vol. 1, num. 1, pp. 3–16, 1989.

[Prechelt-95] Lutz Prechelt. *Adaptive parameter pruning in neural networks.* Technical Report 95-009, International Computer Science Institute, Berkeley, CA, 1995.

[Thimm-94] G. Thimm and E. Fiesler. High order and multilayer perceptron initialization. Submitted to *IEEE Transactions on Neural networks.* Se also: Technical Report 94-07, IDIAP, Martigny, Switzerland, 1994.

[Weigend-90] A. S. Weigend, B. A. Huberman, and D. E. Rumelhart. Predicting the future: A connectionist approach. *International Journal of Neural Systems*, vol. 1, num. 3, pp. 193–209, 1990.