Optimal Hidden Structure for Feedforward Neural Networks

P. Bachiller, R.M. Pérez, P. Martínez, P.L. Aguilar, P. Díaz

Department of Computer Science. University of Extremadura. Escuela Politécnica. 10071 Cáceres. Spain.
{pilarb, rosapere, pablomar, paguilar, pdiaz}@unex.es

Abstract. The selection of an adequate hidden structure of a feedforward neural network is a very important issue of its design. When the hidden structure of the network is too large and complex for the model being developed, the network may tend to memorize input and output sets rather than learning relationships between them. In addition, training time will significantly increase when the network is unnecessarily large. We propose two methods to optimize the size of feedforward neural networks using orthogonal transformations. These two approaches avoid the retraining process of the reduced-size network, which is necessary in any pruning technique.

1 Selection of the Optimal Hidden Structure

The basic principle of network size reduction is to detect and to eliminate collinearity between the input data sets at the hidden layers. Consider a feedforward neural network with N nodes in its single hidden layer and M output nodes. Let h_i be a vector formed by the outputs of the i-th hidden node for all the training patterns. If h_i can be computed as a linear combination of h_j's (expression 1), the i-th hidden unit can be eliminated and equation (2) provides a direct adjustment for the weights of the hidden links connected to the nodes to be retained (w_{lj}) that preserves the initial behavior.

$$h_i = \sum_{j=1\, j\neq i}^{N} c_j h_j \tag{1}$$

$$w_{lj} = w_{lj} + c_j w_{li} \qquad \forall j \neq i \text{ and } 1 \leq l \leq M \tag{2}$$

Orthogonal transformations can lead to a relative decorrelation of the network information providing a good solution to the selection of the optimal set of hidden nodes. In particular, we apply the properties of Householder reflections [3] because of its simple structure and the low computational cost of the operations related to this kind of matrices. Next, these properties are described:

Property 1: *Given m vectors $\in \Re^n$ $[x_1,x_2,\dots,x_m]$, Householder reflections are used to determine which of them are linearly independent.*

Property 2: *If a vector $x \in \Re^n$ is a linear combination of m linearly independent vectors (x_1, x_2, ..., x_m), Householder reflections provide the coefficients of such combination by solving a square triangular equation system.*

The first method we propose uses the above properties to compute the optimal hidden structure of a given network. It starts with a large trained network and selects the optimum hidden units applying the property 1 on the vectors formed by the hidden outputs for the whole set of training patterns. Then, the coefficients c_j of expression 1 are determined by the second property and the weights of the optimal links are adjusted using equation 2.

2 The Optimizing and Training Algorithm

When a pruning technique is used to compute the optimal set of hidden nodes of a given network, it is always necessary to train an overparameterized network [2,4]. The training time increases significantly when the hidden structure of the network is too large and complex for the model being developed. In order to both, accelerate the training process and improve the network generalization, we proposed an optimizing and training algorithm (OTA). This method is based on the idea of determining, in each iteration of the training process, the set of linearly independent outputs of the hidden layer(s) and then updating only the weights of the links connected to those hidden units.

The method starts with a NxMxO neural network and P training patterns. In each iteration of the algorithm, the set of optimum hidden nodes is computed applying the first property of Householder matrices to the M vectors obtained by the outputs of the hidden nodes for the P training patterns. For all the hidden nodes belonging to the obtained set, the weights of their links are adjusted using the *Backpropagation* algorithm. The algorithm iterates until the network reaches the convergence.

To test the effectiveness of our algorithms, experimental results on different test problems have been obtained. These results show that the optimizing and training algorithm reduces significantly the network training time with regards to the original Backpropagation algorithm [1]. Moreover, the generalization capability of the network is improved when both optimizing algorithms are used.

References

1. P. Bachiller, R.M. Pérez, P. Martínez and P.L. Aguilar: *A method based on orthogonal transformation for the design of optimal feedforward network architecture.* Proceedings of the 3rd International Meeting on Vector and Parallel Processing (VECPAR'98), 1998, pp. 541-552.
2. G. Castellano, A. M. Fanelli and M. Pelillo: *An iterative pruning algorithm for feedforward neural networks.* IEEE Transactions on Neural Networks, vol. 8, no. 3, 1997, pp. 519-531.
3. G. H. Golub and C. F. Van Loan: *Matrix computations.* Baltimore, MD: John Hopkins Univ. Press, 1989.
4. P. Kanjilal and N. Banerjee: *On the application of orthogonal transformation for the design and analysis of feedforward networks.* IEEE Transactions on Neural Networks, vol. 5, no. 5, 1995, pp. 1061-1070.