Optimization Methods in Multilayer Classifier Networks for Automatic Control of Lamellibranch Larva Growth

György G. VASS[*], *Mohamed DAOUDI*[**, ***], *Faouzi GHORBEL*[***]

[*] *Budapest University of Technology,*
Department of Microwave Telecommunications
H-1111, Budapest, Goldmann Gy. tér 3, Hongrie
[**] *Département Informatique et Réseaux*
[***] *Groupe de Recherche Images et Formes ENIC/INT*
Ecole Nouvelle d'Ingénieurs en Communication
Rue Guglielmo Marconi
Cité Scientifique 59658 Villeneuve d'Ascq Cedex
e-mail : daoudi@enic.fr

The problem considered here is the age discrimination of lamellibranch larvae. Patterns of larvae are presented to a multilayer feedforward neural network. Samples are represented by shape descriptors calculated on the basis of a normalized arc length parametrization of their boundary. After training, the network will classify samples on the basis of their characteristic shapes. In neural network applications one often faces the problem of optimal network size, which is an implicit function of problem complexity and available amount of data for training. This paper presents some possible solutions to cope with this problem. Results obtained are compared with previous experiments on feedforward networks.

1 Introduction

A large variety of pattern classification problems have been successfully solved in recent years by neural networks (NN). In many cases, however, how to find the optimal size of the network best suited to the given application (e.g. number of hidden layers, number of nodes in each, connections between them, etc.) still remains an open problem. These parameters are often determined in a trial-and-error manner. In this paper we apply multilayer perceptron neural network (MLP NN) for an age discrimination problem. An iterative node pruning algorithm will be introduced. By using this method we have the possibility to remove insalient nodes in the network. An other optimization method will be presented based on the principle that minimizing classification error in a NN is equivalent of maximizing an objective function, called network discrimination function, introduced by Webb and Lowe [(1)].

The real-life application considered here concerns the growth control and age discrimination of lamellibranch larvae. This study is limited to the case of scallop larvae (Pectinacea) which are representatives of the species. Rees [(2)] has shown that natural (i.e. manual) classification of these larvae is possible. In 1990, Ghorbel [(3)] demonstrated an automated method using statistics of calculated pattern descriptors. The benefit of invariant shape descriptors has been shown. Recently, the given age discrimination problem has been effectively solved by the use of multilayer neural networks [4]. This paper is organized as follows. In Section 2, feature extraction is described in detail. Calculated features have to be invariant with respect to certain elementary geometrical operations. Section 3 presents multilayer perceptron neural network (MLP NN) classifiers, in general. In Section 4, optimization methods will be discussed. Finally, Section 5 outlines structure of the database and presents experimental results.

2 Scallop Shell Contour Analysis

In the case of larval phases, the scallop can be described accurately knowing only its shape boundaries. In this work we describe the larva by its contour, which can be easily extracted by means of sophisticated methods like morphological filters. This leads to a considerable simplification of the feature extraction task.

2.1 Invariant Feature Extraction

For natural shapes which do not contain sharp edges, radial representation of curves can provide relatively simple shape descriptors. For applications manipulating 2D star-shaped contours, radial curve representation with n.a.l.p. is shown to be efficient by Bez [5], Ghorbel and Burdin [6]. From this particular parametrization, features invariant with respect to basic similarity transformations (e.g. translation, rotation, expansion, shrinking) are determined. This method uses the Fourier series expansion of log(ρ(l)) where ρ(l) measures the length of the line connecting the boundary of the closed curve to its centroid. (Fig. 1).

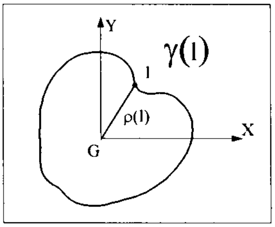

Fig. 1. Radial representation of a star-shaped contour

The radius function obtained this way is axis invariant. Fourier coefficients can be obtained from a clockwise curve description where *l* is the normalized arc length.

$$a_n(\gamma) = \int_0^1 log(\rho(l)) \cdot exp(-2\pi inl)\, dl \qquad (2.1)$$

In (2.1), log(ρ(l)) is used instead of ρ(l) to eliminate the unwanted effect of scale factors : log(Kρ'(l)) = log(K) + log(ρ'(l)). Note, that log(K) is constant.

The magnitude of these coefficients $|a_n(\gamma)|$; *n=1..N;* altogether form a useful set of descriptors, since it has the profitable property of invariance. In Table 1 calculated descriptors for three different ages of larvae are listed.

n	30 days	11 days	2 days
1	0.00789	0.00640	0.00969
2	0.03077	0.04606	0.06782
3	0.01568	0.02535	0.03261
4	0.00303	0.00762	0.01805
5	0.00547	0.00510	0.00804
6	0.00356	0.00266	0.00556
7	0.00243	0.00102	0.00228
8	0.00258	0.00109	0.00196
9	0.00240	0.00144	0.00134
10	0.00204	0.00133	0.00152

Table 1. Fourier descriptors of extracted contours for three different age samples

In general, the first few coefficients (e.g. N=10) characterize well the global shape of the sample, while the remaining part gives the fine details. Consequently, we can assume that with neglecting the remaining part we retain majority of the information for classification.

3 The Neural Network MLP Classifier

Neural networks are dynamic systems composed of highly interconnected layers of simple neuron-like processing elements. In multilayer perceptrons (MLPs), the only connections allowed go from layer to layer and from input to output. Loops and connections within layers are not allowed.

In a MLP network, each unit output is computed as the weighted sum of the activation levels of all the units connected to unit *i*. Let this sum be denoted by S_i. The output is then calculated as a function $f(S_i)$, where f represents a kind of nonlinearity, being very often the sigmoid function, defined by :

$$f(S_i) = \frac{1}{1+e^{-S_i}} \tag{3.1}$$

The input layer receives information carried by the invariant descriptors to be classified. In our experiments, element d ($d = 1..10$) of the qth observation ($q = 1..225$) is formed as

$$i_{q,d} = \left| a_d\left(\gamma_q\right) \right| \ ; \tag{3.2}$$

the respective magnitudes of invariant descriptors calculated for each pattern.

The generalized delta rule of Rumelhart, et al. is a widely used procedure for learning a set of input patterns. In this method, weight adjustment is performed iteratively in the network in order to reduce mean squered system error as rapidly as possible.

4 Methods for Optimization

The problem to determine the optimal size of the NN which conforms well the classification at hand has motivated numerous authors in research (1, 7, 8, 9, 10). Small networks tend to learn fast, but classification errors are often higher than a certain tolerance level. On the other hand, unnecessarily large networks demand long training periods. These NNs can classify learning samples accurately, but in the case of test samples they often have poor performance.

4.1 Iterative Node Pruning

Recently, Mao (7) proposed a node-pruning procedure which is capable of removing insalient nodes in the network to create a small sized network, which can not only approximate faithfully the training set but also generalize well on the test patterns.

According to Mao, the saliency of a node is defined as the amount of increase in the error if this node is removed. From Taylor series expansion of the error function with respect to the output of all nodes, y, and neglecting higher order terms, we have:

$$\Delta E_k = \frac{\partial E_k}{\partial y_i}\Delta y_i + \frac{1}{2}\frac{\partial^2 E_k}{\partial y_i^2}\cdot\left(\Delta y_i\right)^2 \qquad (4.1)$$

where E_k represents the squared error in the network when pattern *k* is presented at its input ($k = 1..N$). In the above formula, we suppose that only one node is removed at once (only the i^{th} component of the vector *y* is allowed to change). The saliency of this node is then defined as

$$S_i = \sum_{k=1}^{N} \Delta E_k \qquad (4.2)$$

Now, in order to evaluate (4.1) we have to compute the first- and second-order derivatives of the error function with respect to the output of individual nodes. These derivatives can be most easily computed in a back-propagation fashion.

The node-pruning strategy can briefly be described as follows:

1. Choose an initial network architecture, *larger than necessary*;
2. Train the network a number of iterations on the input data;
3. For each pattern in the training set evaluate the first- and second-order derivative in the back-propagation fashion;
4. Compute the saliency of each input node and hidden node;
5. *Remove the node* with the lowest saliency value;
6. Retrain the network a small number of iterations;
7. Compute the squared error on both the training and the test data;
8. Repeat steps 3 - 8 until the *stopping criterion* is satisfied.

The stopping criterion is the point where the system error starts to increase. Usually, this happens after a critical node has been removed.

4.2 Optimization Based On Discriminant Analysis

An other kind of optimization technique is based on discriminant analysis. Investigations into the nature of the transformations performed by the MLP NN trained by the least mean squares error procedure have revealed interesting connections with classical discriminant analysis known from statistical literature (9,10). Numerical simulations suggest that a network with non-linear hidden units perform more severe feature extraction than linear discriminant analysis, by finding a non-linear transformation which returns a larger value of a separability criterion involving the scatter matrices of the training vectors. Gallinari et al. (10) has shown that from one layer to the next, internal representations tend to be more and more separated, as clusters become more and more compact, due to the compression effect of the sigmoid function. In 1990, Webb and Lowe (1) demonstrated that minimizing the sum of squares error at the network output is in fact equivalent to maximizing a particular norm, called *network discriminant function*, under the condition that the output layer transfer function is linear.

Noting n_i the a priori probability of class w_i, the total and between-class scatter matrices can be defined respectively as:

$$T = E\left\{\left(Y - E\{Y\}\right)\left(Y - E\{Y\}\right)^t\right\} \tag{4.3}$$

$$B = \sum_{I=1}^{N_c} n_i \left(E\{Y|w_i\} - E\{Y\}\right)\left(E\{Y|w_i\} - E\{Y\}\right)^t \tag{4.4}$$

where E{Y} represents the expected value of random vector Y, and E{Y|w_i} the class conditional expected value of Y. As usual, *t* denotes transposition.

In order to formulate a criterion, we need to convert the scatter matrices into a single number. This number must be larger when the between-class scatter is larger or the within-class scatter is smaller. Several criteria have been proposed by Fukunaga (17). Among all those, the most frequently used one is

$$J_1 = tr\left(T^{-1}B\right) \tag{4.5}$$

In this paper, criterion J_1 will be used.

5 Experimental Results

Three set of larva samples of ages 2, 11, and 30 days, respectively, have been chosen for classification. Contour of the shell have been extracted in the way described in Section 2. For each of these contour, Fourier Descriptors are calculated. Because of the simple shape of the larvae the decomposition is reduced to 10 significant components, which altogether form input database for the classification.

In this work, two-layer feedforward neural networks have been trained by the well known backpropagation (bp) learning rule. Size of the input layer was fixed by the dimension of the feature space (i.e. 10), while output of the network consisted of three nodes. Fig. 2 illustrates such an architecture.

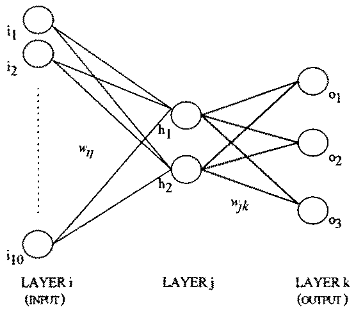

Fig. 2. Architecture of the network

In a classical MLP layout, choice of neurons for hidden layer is up to the network designer. In this sense, an optimal architecture can be found as the smallest sized network still performing well on training and test data, respectively. Different approaches are possible towards this end. First, one can proceed manually by doing experiments on a collection of networks having different number of hidden layer neurons and keeping track of classification results for each. Among all these architectures the optimal one can be easily selected.

In a second phase of experiments we can accept a more automated algorithm in that a relatively large network is set up at the beginning and an iterative node pruning process is allowed to eliminate less significant nodes (see description in Section 4.1). During this procedure both input and hidden layer neurons were eliminated until an optimal architecture is reached. In this state of the NN, removal of a node results in no additional gain in overall system error. Results of these experiments are summarized in Table 2. It is interesting to remark that these results are in good accordance with those found by the simple step by step approach. For instance, a simple eigenvalue - eigenvector analysis can show that majority of discriminatory information tends to be concentrated in the first few descriptors. Thus, neglecting, say, three of them, and keeping only the first seven descriptors, *94.5 %* of the overall information content can be retained. As a result, three corresponding input nodes can be eliminated from the network, which justifies the results found by the node pruning procedure (see Table 2).

Original network structures	Optimal structure detected (after pruning)	Number of iterations needed	Learning file recognition rates	Test file recognition rates
10 - 15 - 3	7 - 2 - 3	360	1.00	0,847
10 - 11 - 3	7 - 2 - 3	365	1.00	0,847
10 - 8 - 3	7 - 2 - 3	374	1.00	0.847
10 - 5 - 3	7 - 2 - 3	357	1.00	0,847

Table 2. Results obtained by node pruning. Different initial network sizes have been considered; training has been stopped after system error decreased below 0.0001 .

The third approach is based on the revealed relationship between statistical discriminant analysis and error minimization performed by the NN (see Section 4.2 for details). According to this approach, internal separability of data during successive iterations increases in parallel with the decreasing tendency of overall system error. This phenomenon could be clearly observed during our experiments. In Fig. 3a, separability criterion (equ. 4.5) together with the squared system error have been represented for a network with two hidden neurons. Fig. 3b. depicts the same quantities for a network with three hidden neurons. In fact, observing the separability criterion in case of different architectures allows to inspect NN performance in a global scale and makes it easier to select among all the explored architectures the best possible one in terms of optimal data separation. The training process can be then stopped after a suitable amount of separability in the data is achieved. Indeed, possibly high data separability is of primal importance in order to have good final classification rates. This method can, thus, facilitate the optimal choice of network.

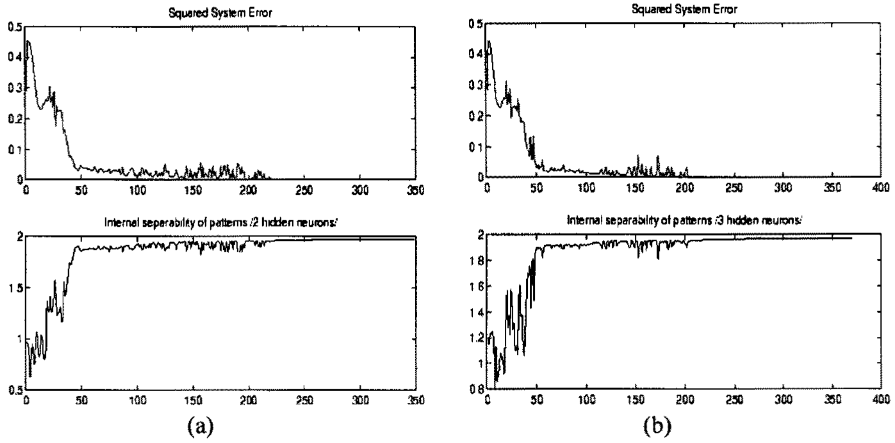

Fig 3. Internal separability of data and normalized system error vs. number of iterations.
(a) network with 2 hidden neurons (b) network with 3 hidden neurons

5 Conclusions

Recent research has shown that neural networks can be remarkably well applied to various pattern classification problems. In a good deal of applications, however, one can hardly find direct relationships between network dimensions and problem complexity. Particularly, the number of hidden neurons chosen for the network can have direct influences to classification performance.

This experimental study concerns the age discrimination of lamellibranch larvae using a two-layer feedforward neural network trained by the backpropagation learning rule. In the preprocessing phase, boundary of the patterns are described using a radial curve parametrization technique.

In this study, we present different techniques to achieve optimal performance for the classifier. First, an iterative node pruning algorithm is used to reduce the network to a reasonable size. Results are compared to conventional methods. An interesting phenomenon is also displayed, in that input layer neurons are eliminated in order to increase NN performance. In the second phase, by taking advantage of discriminant analysis techniques, we test different network architectures and show how to select, among all these architectures, the one which assures the highest data separability and results in the best classification rates.

6 Acknowledgement

The authors would like to express their thanks to IFREMER (Direction des Ressources Vivantes) for providing recordings of lamellibranch larvae for this research as well as to F. Ghorbel for useful discussions.

7 References

[1] Webb, A.R., Lowe, D.: "The Optimised Internal Representation of Multilayer Classifier Networks Performs Nonlinear Discriminant Analysis", *Neural Networks*, Vol. 3, pp. 367-375 (1990)

[2] Rees, C.B.: "The identification and classification of lamellibranch larvae", *Hull. Bull. Mar. Ecol.* Vol. 3, No.19, 73-104 (1950)

[3] Ghorbel, F.,de Bougrenet de la Tocnaye, J.L.: "Automatic Control of Lamellibranch Larva Growth Using Contour Invariant Feature Extraction", *Pattern Recognition*, Vol.23, No. 3 (1990)

[4] Vass Gy., Daoudi, M.: "Automatic Control of Lamellibranch Larva Growth using Neural Network Classifier" *Proc. of CESA'96*, Lille, France, pp. 531-536, July (1996)

[5] Bez, H.E.: "On analysis of symmetry for plane curves and plane curve algorithms", *Comp. Aided Geom. Design*, Vol.9, pp. 125-142 (1992)

[6] Ghorbel, F., Burdin, V.: "Invariant Approximation of Star-Shaped Form for Medical Applications", *Curves and Surfaces II*, P.J. Laurent, A. Le Méhauté and L.L. Schumaker (eds.), AKPeters, Boston, pp. 1-9 (1991)

[7] Mao, J.: "Design and Analysis of Neural Networks for Pattern Recognition", Dissertation submitted to Michigan State University (1994)

[8] Pao, Y.H.: "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley, Reading, Massachusetts (1989)

[9] Lengellé, R., Denoeux, T.: "Training MLPs Layer by Layer Using an Objective Function for Internal Representations", *Neural Networks*, Vol. 9, No. 1, pp. 83-97 (1996)

[10] Gallinari, P., et al.: "On the Relations Between Discriminant Analysis and Multilayer Perceptrons", *Neural Networks*, Vol. 4, pp. 349-360 (1991)