DIVIDE AND CONQUER ALGORITHMS FOR CONSTRUCTING NEURAL NETWORKS ARCHITECTURES

K. KOUTROUMBAS A. POULIAKIS * and N. KALOUPTSIDIS *

Abstract— **In this paper two algorithms for the construction of feedforward neural network architectures are discussed. Their close relation with a special class of decision tree classifiers is commented. Finally, the performance of the algorithms is assessed on a complex cytological pattern classification application.**

Keywords— **Neural networks, Back Propagation, Decision trees, Image Analysis, Image Measurements, Cytological Diagnosis, Stomach Lesions.**

INTRODUCTION

One of the major problems in the field of neural networks (NN) is the estimation of the size of the architecture as well as the determination of the corresponding parameters. Algorithms such as back propagation and its variants (e.g. [1], [2], [3]) compute the parameters for fixed size neural network architectures, while others determine dynamically the size as well as the corresponding parameters (eg. [4], [5], [6], [7], [8]).

In this paper two algorithms that construct three layer architectures are discussed. The number of the first layer nodes is of the same order with that of architectures trained using back propagation. Moreover, the generalization ability exhibited by the resulting architectures is very satisfactory. Finally, these algorithms are closely related to a special class of decision tree classifiers, known as Linear Tree Classifiers (LTC).

Both algorithms follow the divide and conquer principle. More specifically, the first one, which is called SNN (Separation of Nearest Neighbors from different categories), finds the closest pair of the available patterns x, y that belong to different categories and constructs the hyperplane H that bisects the line segment xy. This procedure is applied recursively on the available patterns lying on the positive (negative) side of H until all the defined regions contain patterns of the same category.

The second algorithm, called OSC (Optimal separation of categories), determines a hyperplane H that achieves the best possible partition of the available patterns and then is applied recursively on the available patterns lying on the positive (negative) side of H. Again, the algorithm stops when all the defined regions contain patterns from the same category.

In the sequel we denote by $SNN(S)$ and $OSC(S)$ the architectures provided by SNN and the OSC algorithm, respectively, when applied on the data set S.

The authors are with the Department of Informatics, Division of Communications and Signal Processing, University of Athens, Panepistimiopolis, T.Y.P.A. Buildings, 157 71 ATHENS, GREECE, Tel. 7217941, E:mail: koutroum@di.uoa.gr.

Both algorithms classify correctly all the patterns of S and they are insensitive to the order of presentation. Also, it is worth noting that the resulting architectures do not implement the Voronoi tessalation (eg. [6]) of the feature space.

THE ALGORITHMS

A single Σ node with weight vector $w' = [w_1, \ldots, w_d]$, threshold w_0 and the hard limiter as output function, implements the separation of R^d achieved by the hyperplane H, $w^T x = \sum_{j=0}^{d} w_j x_j = 0$, where $w = [w',\ w_0]^T$ and $x_0 = 1$. If for a point $x \in R^d$, $w^T x > 0 (< 0)$, the output of this node will be $+1$ (-1). The architectures produced by both SNN and OSC employ Σ nodes only.

We consider the two category classification problem. Let $I = \{-1, 1\}$, $A \subset R^d$ and

$$S = \{(x_i, t_i), x_i \in A, t_i \in I, i = 1, \ldots, p\}.$$

S is the training set and t_i denote the labels of the patterns.

The SNN algorithm

This algorithm first determines the first layer nodes of $SNN(S)$ and then the second layer nodes. The third layer consists of a single Σ node. The first layer nodes corresspond to hyperplanes that bisect line segments and are determined by the following recursive procedure

- procedure $SNN\_first\_layer\_nodes(S)$
  - If all patterns of S belong to the same category then
    - return
  - Else
    - (A) Find the closest pair of patterns $x, y \in S$ such that x (y) belongs to category 1 (-1).
    - (B) Construct the hyperplane H_S that bisects the line segment xy with x lying on its positive side. Let $S^+ = S \cap H_S^+$ and $S^- = S \cap H_S^-$, where H_S^+ (H_S^-) denotes the positive (negative) half space of H_S.
    - call $SNN\_first\_layer\_nodes(S^+)$
    - call $SNN\_first\_layer\_nodes(S^-)$
  - End if
- End procedure.

Let k be the number of hyperplanes. For each hyperplane H defined by the above procedure a Σ node, F_H, that implements the separation implied by H is employed. Each such node is attached to the first layer. Thus, the number of the first layer nodes is k.

Let R_i, $i = 1, \ldots, l$, be the regions defined by the above hyperplanes that contain patterns from category 1. The

second layer consists of l nodes, N_{R_i}, $i = 1, \ldots, l$, each one corresponding to one of the above regions. What remains to be determined is the connections between the first layer nodes and each one of the second layer nodes. For each node N_{R_i}. The following recursive procedure determines the connections ending to N_{R_i}.

- procedure $connections(R_i, H_S)$
  - If $R_i = S$ then
    - return
  - Else if R_i lies on the positive side of H_S then
    - Connect F_{H_S} with N_{R_i} with weight connection equal to 1.
    - call $connections(R_i, H_{S^+})$
  - Else if R_i lies on the negative side of H_S then
    - Connect F_{H_S} with N_{R_i} with weight connection equal to -1.
    - call $connections(R_i, H_{S^-})$
  - End if
- End procedure.

The threshold for N_{R_i} is set to $m_i - 0.5$, where m_i is the number of corresponding weights. Finally, the outputs of the second layer nodes are fed to the third layer node with threshold $-l + 1$. The connection weights are all set to 1.

One can easily verify that $l = k + 1/2$ when k is odd and $l = k/2$ or $k/2 + 1$ when k is even. The behavior of the constructed architecture is very simple to explain. What $SNN(S)$ essentially does is to determine the region where the input pattern x lies and to assign the input pattern to the category where the points of that region lie.

The OSC algorithm

As in the SNN algorithm, the OSC algorithm first determines the first layer nodes of $OSC(S)$, then the second layer nodes and then the connections between the first and the second layer.

The recursive procedure, $OSC\_first\_layer\_nodes(S)$, that determines the first layer nodes of the network is the same with $OSC\_first\_layer\_nodes(S)$, with the exception that the steps (A) and (B) are substituted by the following step.

- Find a hyperplane H_S that minimizes the number of the misclassified patterns of S. Let $S^+ = S \cap H_S^+$ and $S^- = S \cap H_S^-$, where H_S^+ (H_S^-) denotes the positive (negative) half space of H_S.

The second and the third layer nodes as well as the connections between the first and the second layer are determined exactly as in the SNN algorithm.

OSC1 AND OSC2 SCHEMES

As we saw earlier, the $OSC\_first\_layer\_nodes$ procedure, determines hyperplanes that minimize the number of misclassified patterns of S. However, this is not easy even for moderate sizes of S. In order to overcome this problem approximate OSC schemes can be used. The first scheme, called OSC1, employs the pocket algorithm ([9]) for the determination of the best separating hyperplane H at each step.

Another variant of OSC, also called OSC2 scheme, employs the well known LMS algorithm (eg. [3]) for the determination of the separating hyperplane H at each step. In contrast to the pocket algorithm, the LMS algorithm takes into account the distance of each pattern from the hyperplane.

CONNECTION WITH DECISION TREES

Decision tree classifiers (eg. [10]) have been successfully used in pattern recognition applications. A special class of the above classifiers contains the so called Linear Tree Classifiers (LTC) (eg. [11]). An LTC consists of a root node, a set of terminal nodes and a set of non terminal nodes. Each of the nonterminal nodes is associated with a linear discriminant function (or equivalently a hyperplane in the feature space). An LTC performs a hierarchical partition of the feature space into nonoverlapping polyhedral sets (i.e. sets defined as intersections of half-spaces). Each terminal node corresponds to one of the above regions and, each such region is assigned to a class. In order to decide to which category a given pattern x belongs, we have to compute the value of the discriminant function of the root node and, according to the resulting value, we move to the next node. This procedure continues until a terminal node is reached.

LTC's are closely related to the 3-layer feedforward neural networks with Σ nodes. Specifically, in [12], [11], it is discussed how an LTC can be mapped into a 3-layer feedforward neural network, of the structure discussed above.

The major problem associated with the design of an LTC is the choice of the optimal number of nodes as well as the determination of the discriminant functions associated with each one of them. It is not difficult to realize that the above algorithms can also be used for the construction of LTC's. Specifically, only the procedures that construct the first layer nodes are needed.

PERFORMANCE OF THE ALGORITHMS ON A REAL CYTOLOGICAL PROBLEM.

Presentation of the Specific Cytological Diagnostic Problems

During the last decade several efforts to test the capability of Artificial Neural Networks (ANNs), to perform medicine diagnostic tasks have been reported [13], [14]. Classification, pattern recognition and decision support are some of the most important and emerging applications of neural nets in the field of cytology.

Gastric cytology has not reached wide acceptance in the investigation of gastric lesions because of the difficulties in the discrimination of benign lesions with severe regenerative alterations from well differentiated cancer cells [15]. Several efforts to solve this problem by statistical evaluation of the morphometric data has not yielded to the individual patient level. Recently attempts to solve the problem of classification both at cellular level and at the patient level have appeared [13].

Geometric features	Densitometric features
Area	Mean value of histogram
Perimeter	Standard deviation of histogram
Major axis	Variance of histogram
Minor axis	Short run of run length matrix
Diameter	Long run of run length matrix
Circularity	Grey level of run length matrix
Roundness factor	Distribution of run length matrix
Contour ratio	Maximum of co-occurrence matrix
Contour index	Inertia of co-occurrence matrix
Form area	Entropy of co-occurrence matrix
Form perimeter	Contrast of differences histogram
	Mean value of differences histogram
	Standard deviation of differences histogram
	Entropy of differences histogram

TABLE I: The measured features.

The Cell Classification System

The cell measurement process is accomplished by the image processing system. An image acquisition module acquires images and transforms them to electrical signals. It is followed by the next module that performs the image sampling and quantization. Next the image segmentation subsystem isolates objects on the images. Each object is represented by a vector via the feature extraction subsystem which implements a number of measurements. Finally each vectorial representation of the cell is processed by one or a group of classifiers that assigns the cell nuclei to categories.

In the specific context of the cytologic application the above image processing system is specified as follows:

The image acquisition subsystem consists of a color CCD camera attached to the top of a microscope. The remaining subsystems are implemented by software. The image segmentation subsystem aims at the isolation of the cell nuclei. The variety of image types make segmentation hard. A mix of manual and completely automated segmentation techniques have been developed, depending on image type and the quality of the resulting outcome. Feature extraction is also performed by software. Three types of measures are considered: size, shape and texture, based on standard cytological practice. Table I summarizes the feature information generated by the extraction system for each nucleus. Features are grouped according to the physical characteristics of cells into two categories: *geometric* and *densitometric*; densitometric features are associated with texture.

The geometric features are related to the coordinates of the lines that represent the boundary of a nucleus. They describe properties relevant to the size (for example area, perimeter, diameter) and the shape (eg. form area, form perimeter, circularity). A detailed description of the computational methods employed to determine the geometric characteristics is supplied in [16], [17]. Densitometric features are relevant to the values of the pixels inside the region that is created by the lines surounding the nucleus.

Algorithm	Benign	Malignant	OA
NSS	100%	100%	100%
OSC1	100%	100%	100%
OSC2	100%	100%	100%
BP	90.3%	87.1%	89.5%

TABLE II: Performance of the classification schemes when applied on the training set. The numerals show the mean number of correctlly classified patterns in each class. BP=back propagation, OA=overall accuracy.

Densitometric characteristics are relevant to texture caused by the nucleus chromatin. From the various methods proposed in the literature for textural descriptors (see [16], [18], [19]), four models have been implemented, based on: a) histogram b) differences histogram c) run length matrix properties and d) co-occurrence matrix properties. The first two models are computationally simple but texture discrimination is poor. The other two models are more complex but give better information about the texture structure [19]. A detailed description of the computational methods of textural features is provided in [19].

The classification subsystem relies on neural network architectures and targets to bipartite classification into benign and malignant cells.

Data set

The aforementioned feature extraction mechanism leads to nucleus representations by vectors of 25 components (11 related to geometric features and 14 related to densitometric features).

The data set consists of 3538 vectors. 851 of these are associated with cancer or neoplastic nuclei (malignant) and the remaining 2687 vectors are related to non neoplastic (benign) nuclei that are caused by gastritis or ulcer. 10% of the data of each class are randomly selected for training. Thus the total training set contains 354 vectors: 85 derived from cancer of neoplastic nuclei and 269 derived from benign nuclei. The remaining data in the test set consists of 766 vectors from malignant cases and 2418 vectors from benign lesions.

Note that nucleus classification was carried out by three expert cytologists, to ensure that all cells are correctlly classified.

For comparison purposes a two layer back propagation network has been employed. The number of nodes for the hidden layer is incremented one and increased by one neuron for each trial until it reached 50 neurons. The architecture that gave the better generalization results was finally used.

All algorithms were tested ten times by selecting different training sets of the same number of vectors and by using the remaining data as test sets.

The results are summarized in tables II and III. The percentages and the numbers of nodes that appear in the tables are averaged over the ten trials.

Algorithm	B	M	OA	# of nodes
SNSS	94.79%	89.01%	93.40%	37.6
OSC1	96.93%	85.48%	94.17%	5
OSC2	97.25%	86.58%	94.68%	4.2
BP	88.96%	74.67%	85.52%	25-7.5-1

TABLE III: Performance of the classifiers when applied on the test set. The numerals show the mean number of correcly classified vectors in each class.The last column shows the mean number of nodes for each architecture. BP=back propagation, OA=overall accuracy, M=malignant, B=benign.

CONCLUSIONS

In this paper two algorithms for the construction of neural network architectures are discussed. The performance of the above algorithms has been tested on a complex cytological case, the obtained results are more satisfactory than those given by other classifiers used in such applications. Table III indicates that the OSC algorithms are capable of encoding a greater amount of information than the SNN algorithm. This is due to the fact that the two versions of the OSC algorithm create the separating hyperplanes taking into account all the available patterns, while SNN creates its separating hyperplanes taking into account only the two closest of the available patterns that belong to different categories.

REFERENCES

[1] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Welsey, first edition, 1991.

[2] Simon Haykin, *Neural networks a comprehensive foundation*, Macmillan College Publishing Company Inc. ISBN 0-02-352761-7, 1994.

[3] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1998.

[4] J. P. Nada M. Mezard, "Learning in feedforward layered neural networks: The tilling algorithm", *J. Phys. A: Math. Gen.*, vol. 22, pp. 2191–2203, June 1989.

[5] N. N. Bose and P. Liang, *Neural Network Fundamentals with Graphs, Algorithms and Applications*, Mc-Graw Hill, Inc., 1995.

[6] O. J. Murphy, "Nearest neighbor pattern classification perceptrons", *Proc. of the IEEE*, vol. 78, no. 10, pp. 1595–1598, Oct. 1990.

[7] K. Koutroumbas and N. Kalouptsidis, "Nearest neighbor pattern classification neural networks", *Proc. of IEEE World Congress on Computational Intelligence*, June 1994.

[8] S. Vithlani A. N. Refenes, "Constructive learning by specialization", *Proc. ICANN '91*, June 1991.

[9] S.I. Gallant, "Perceptron based learning algorithms", *IEEE Trans. Neural Networks*, vol. 1, pp. 179–191, 1990.

[10] R. Olsen L. Breiman, J. Friedman and C. Stone, *Classification and Regression Trees*, Wadsworth International, Belmont, 1984.

[11] Y. Park, "A comparison of neural net classifiers and linear tree classifiers: their similarities and differences", *Pattern Recognition*, vol. 27, pp. 1493–1503, 1994.

[12] I.K. Sethi, "Entropy nets: From decision trees to neural networks", *Proceedings of the IEEE*, vol. 78, pp. 1605–1613, 1990.

[13] P. Karakitsos, E. Botsoli Stergiou, A. Pouliakis, M. Tzivras, A. Archimandritis, A. Ioakim Liossi, and K. Kyrkou, "The potential of the back propagation neural network in the discrimination of benign from malignant gastric cells.", *Analytical and Quantitative Cytology and Histology*, vol. 18, no. 3, pp. 245–250, June 1996.

[14] P. Karakitsos, B. Cochand-Priollet, P.J. Guillausseau, and A. Pouliakis, "Potential of the back propagation neural network in the morphologic examination of thyroid lesions", *Analytical and Quantitative Cytology and Histology*, vol. 18, no. 6, pp. 494–500, Dec. 1996.

[15] T. Kasugai and S. Kobayashi, "Evaluation of biopsy and cytology in the diagnosis of gastric cancer", *Am. J. Gastrenterol*, vol. 62, pp. 199–203, 1974.

[16] M. Sonka, V. Hlavac, and R. Boyle, *Image processing analysis and machine vision*, Chapman&Hall, first edition, 1994.

[17] J. C. Russ, *The Image Processing Handbook*, CRC Press, IEEE Press, Boka Raton, second edition, 1995.

[18] R. M. Haralick et. al., "Textural features for image classification", *IEEE Trans. Sys. Man and Cybern.*, vol. 3, pp. 610–621, Nov. 1973.

[19] I. Pitas, *Digital Image Processing Algorithms*, Prentice-Hall, first edition, 1993.