Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 57 (2004) 257–274
NEUROCOMPUTING
www.elsevier.com/locate/neucom

Neural network induction graph for pattern recognition

O. Lezoray[a,*], D. Fournier[b], H. Cardot[c]

[a] IUT Dept. SRC, LUSAC EA 2607, 120 Rue de l'exode, Saint-Lô, F-50000, France
[b] LIH EA 3219, 25 rue Philippe Lebon, B.P. 540, Le Havre, F-76058, France
[c] Dept. Informatique, Polytech'Tours, LI EA 2101, 64 Avenue Jean Portalis, Tours, F-37200, France

Received 31 August 2001; received in revised form 14 May 2003; accepted 20 October 2003

Abstract

This paper presents a novel architecture of neural networks designed for pattern recognition. The concept of induction graphs coupled with a divide-and-conquer strategy defines a neural network induction graph (NNIG). First, the NNIG concept is described and its properties detailed. It is based on a set of several little neural networks, each one discriminating only two classes. The specialization of each neural network simplifies their structure and improves the classification. The principle used to perform the decision of classification on an input pattern is explained. The latter enables to take into account dubious decisions identified by the NNIG. The last section presents experimental results. A significant gain in the global classification rate can be obtained by using an NNIG. The discussion is illustrated by tests on databases from the UCI machine learning database repository. The experimental results show that an NNIG can achieve a better learning, simpler neural networks and an improved performance in classification. A final illustration is presented on a real microscopical imaging problem for the classification of cells in serous cytology.

PACS: 07.05.Mh; 07.05.Kf; 87.57.Ra

Keywords: Induction graphs; Neural networks; Dubious decisions

* Corresponding author. Tel.: +33-(0)-233775517; fax: +33-(0)-233771167.
E-mail addresses: olivier.lezoray@info.unicaen.fr (O. Lezoray), dominique.fournier@univ-lehavre.fr (D. Fournier), hubert.cardot@univ-tours.fr (H. Cardot).

0925-2312/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2003.10.010

1. Introduction

Data classification is a central problem in the field of pattern recognition. A lot of methods have been proposed to this aim and they have become classical ones (decision trees [3], Bayesian approach [5], fuzzy clustering [6], cluster analysis [5]). Many of them have led to numerous industrial applications. In recent years, neural networks and more particularly multi-layer perceptrons (MLP) [9] have received a great deal of attention. The reasons for this success essentially come from their universal approximation capabilities. An important problem in the building of a suitable neural network architecture is the choice of its structure which is generally chosen a priori. Dealing with complex problems, to obtain a good generalization behavior is not a trivial task and has to be carried out by a neural network specialist. In this paper, a new strategy for building a neural classifier is introduced. The latter redefines the learning task of a classical large neural network in several simpler ones. Using simpler networks can lead to good generalization abilities without requiring human assistance. The redefinition of the learning task into several smaller ones follows a divide-and-conquer strategy since it splits the classification problem into several simpler ones.

The paper is organized as follows. Section 2 details the outline of our neural network architecture which is called neural network induction graph (NNIG). Section 3 details the principle and the construction of the NNIG. From the latter, class discrimination is performed even for uncertain decision cases as it will be further discussed. In the last section, we present experimentations on the University of California at Irvine (UCI) repository data sets [1] and from our own works on neural pattern recognition in microscopic imaging [14].

2. Induction graphs

Decision tree is a non-parametric classification method widely used in pattern recognition. Such classifiers use decision functions which partitions the feature space into two regions to determine the identity of an unknown input pattern. These decision functions are organized in such a way that the outcome of successive decision functions refines the decision of classification. The result of the learning process is represented by a tree whose nodes specify decision functions on attributes values and whose leaves correspond to sets of input examples with the same class or to elements in which no more attributes are available. This data classification method is brought into widespread use in induction graph theory [16,21]. Induction graphs are a generalization of decision trees. In a decision tree, the classification decision is made from root towards leaves without possible backward return from a node to a lower or higher level node in the tree. Induction graphs enable to introduce links between different level nodes and thus compose a graph structure. This method is now much used in browsing data methods such as knowledge retrieval from the data (also called data-mining [7]). Many works use a tree structure to build either a neural tree [19] (the nodes of the tree being neurons which are used as non-linear binary decision functions), or neural networks trees [17,18] (nodes of the tree being neural networks which are used as non-linear n-ary

decision functions). We propose to define a new structure based on a graph of neural networks which is called an NNIG. Unlike the usual methods, we do not build a neural network tree but a structure whose nodes are neural networks and are completely connected, namely, a neural network induction graph.

3. Classification by a NNIG

3.1. NNIG principle

The construction of the NNIG is supervised. It builds a neural network graph. When there is a large number of classes labelling the data, the classification by only one large network can be difficult: this neural network encounters difficulties with generalization. What we suggest consist in using only small neural networks and to simplify the problem we reduce the number of classes to be recognized: each network has to classify only two classes. Therefore, to discriminate more than two classes, several networks are needed. Our architecture arises in the following way. The neural networks used in this paper are MLP networks with back-propagation of the gradient error MLP. The NNIG construction is done in three steps:

- The construction of the neural networks, knowing the number of classes of objects to be separated.
- The training of each neural network.
- The construction of the NNIG.

3.2. Construction of a NNIG

For a classification problem with n classes, a set of unconnected networks is built, each one being in charge of separating elements from two distinct classes. The set of different classes is denoted by $C = \{C_1, C_2, \ldots, C_n\}$ and $|C| = n$. For n classes, that leads to have $(n(n-1))/2$ neural networks being used for classification. The set of neural networks is given by $\Re = \{\Re_{c_1,c_2}; \Re_{c_1,c_3}; \ldots; \Re_{c_{n-1},c_n}\}$. The training of each neural network is processed in a sequential and unordered way. The networks learn one by one according to the order in which they were created. That does not have any influence on the training of each network since there is at this time no connection between them. The difficulty in separating n classes is simplified by the specialization of each network, because a network is interested only in the separation of two classes. When one of these neural networks learns how to differentiate two classes, only the objects belonging to these two classes are presented to the neural network. This implies, on the one hand to simplify the training (since the set of data to be learned is restricted) and on the other hand, to make easier the discrimination between these two classes since the network learnt how to recognize only those. The global training data set containing patterns of all the different classes is denoted by S_T. The latter is divided in several subsets for each neural network. $S_T(c_i, c_j)$ is the data set which corresponds to the neural network which differentiates the classes C_i and C_j and contains patterns of only those two classes. The initial training data $(S_T(c_i, c_j))$

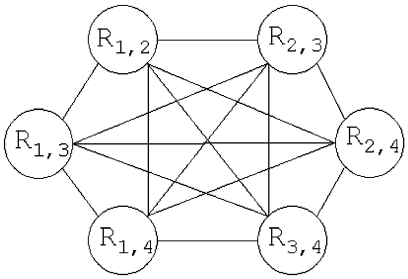


Fig. 1. The NNIG created for a four class problem. $\Re_{c_i,c_j}$ denotes the network discriminating the classes C_i and C_j.

associated to each neural network is split into two subsets: a learning set ($S_L(c_i, c_j)$) and a validation set ($S_V(c_i, c_j)$). The latter consists in 20% of $S_T(c_i, c_j)$ and the learning set in 80% of $S_T(c_i, c_j)$. The learning of a neural network is performed on $S_L(c_i, c_j)$ and the $S_V(c_i, c_j)$ validation set is used to evaluate the classification rate of the network during the training. Therefore, the validation set is not learnt by the neural networks. The structure of the neural networks used is the following one: a layer of inputs containing as many neurons as the number of attributes associated with the object to be classified, a hidden layer containing a variable number of neurons and one output neuron. The value of the output neuron is in the interval $]-1, 1[$. According to the sign of the result associated with this single neuron, an object is classified in one of the two classes that the network separates. The neural networks used by our architecture are very simple (only one hidden layer, only one neuron of output). This has several advantages. The simplicity of the task associated to each neural network simplifies the convergence of the training as well as the search for a simple structure. The generalization of their structure can be made in a dynamic way very easily. Therefore, an automatic method is used to find the number of hidden neurons that gives the best classification rate [4,12]. Once the training of a $\Re_{c_i,c_j}$ network is carried out, the classification rate $Q(\Re_{c_i,c_j})$ of this network is available. The latter is obtained on the $S_V(c_i, c_j)$ validation data set and thus relates only to data that have not been learnt. Once all the neural networks were created and trained independently, the NNIG is built. Each neural network is connected to all the other ones. This produces a graph of fully connected neural networks. The networks are not directly connected to one another: there is no link between the neurons of each neural networks. The NNIG defines an unweighted and unoriented graph with a structure that enables to know which network is directly reachable from one node of the graph. An example for the creation of the NNIG with a four class problem is given in Fig. 1. Six different neural networks are created, each one discriminating only two classes.

3.3. Branch quality index (QI)

After the training step, each neural network obtains a classification rate (denoted $Q(\Re_{c_i,c_j})$). When a neural network classifies a new pattern X, it gives the value of

the output neuron $O(\Re_{c_i,c_j},X)$. The sign of this value gives the class of X. It will be noted thereafter that if a neural network separates two classes C_i and C_j, an input pattern X is considered as class C_i if $O(\Re_{c_i,c_j},X)<0$ and C_j if $O(\Re_{c_i,c_j},X)>=0$. However, it might be beneficial to weight the decision of each neural network according to different factors. Three elements act upon the result given by a network: its potential of classification (the classification rate), its decision (the value of the output neuron) and the representativity of the data set used for the training [8]. Using only the value of the output neuron to assess the relevance of a classification performed by a neural network may cause dubious decisions. We suggest to use a (QI) which makes a trade-off between all these parameters. For a neural network $\Re_{c_i,c_j}$ and an input pattern X, we define:

$$QI(\Re_{c_i,c_j},X)=|O(\Re_{c_i,c_j},X)|Q(\Re_{c_i,c_j})\frac{|E(\Re_{c_i,c_j})|}{|\log(n)|} \tag{1}$$

$$\text{with}\quad E(\Re_{c_i,c_j})=\frac{|S_L(c_i,c_j)|}{|S_L|}\log\left(\frac{|S_L(c_i,c_j)|}{|S_L|}\right). \tag{2}$$

$|S_L|$ and $|S_L(c_i,c_j)|$, respectively, denote the size of the global learning data set and the specific one associated with the $\Re_{c_i,c_j}$ neural network. QI quantifies the relevance of a given neural network of the NNIG for a classification decision. A NNIG being a set of connected neural networks, it is therefore possible to define, as for classical graphs, a branch in the NNIG. A branch in the NNIG is an ordered set of neural networks following the connections between one network to another. Since a QI is computed with each network of a branch, we can define a branch quality index (BQI) which gives the relevance of the set of neural networks used. The Branch Quality Index is defined as the sum of all the Quality Index of the neural networks of the branch and formally given by

$$BQI(\Omega,X)=\sum_{i=1}^{|\Omega|}QI(\Omega_i,X), \tag{3}$$

where Ω is an ordered set of neural networks and Ω_i the ith network of Ω.

3.4. Classification decision by a NNIG

3.4.1. Selection by elimination principle

Once an NNIG has been created, the problem of the choice of the identity of an input pattern (its final class) arises. Indeed, if an object is proposed to a neural network of the NNIG and if the unknown input pattern does not belong to one of the two classes discriminated by a neural network, the answer is not significant and that is likely to distort the decision of classification. We set up with this intention a selection by elimination. By selection we understand determination of the identity (i.e. class) of an input pattern and by elimination we indicate the way to choose the class. If a network of the NNIG is used, the latter will classify the object in one of the two classes that it differentiates (C_i and C_j). If this network indicates the object as belonging to the class C_i then C_j is eliminated and reciprocally. This is the principle of elimination. To classify a pattern X by an NNIG, successive interrogations of the neural networks

progressively eliminate the possible classes until only one final class is available giving the identity of the unknown input pattern. This implies the following definition.

Definition 1. For n classes to differentiate, a branch in the NNIG has a depth of $(n-1)$ neural networks.

3.4.2. α paths in a NNIG

Following the elimination principle, an initial neural network is firstly chosen as an entry in the NNIG. This network defines the first one which is requested to classify the input pattern and designs the first level of classification. The network classifies the pattern and a class is eliminated. Then, adjacent networks of the initial one can be further used to continue the classification until only one class is available. Using successive interrogations, the decision of classification is gradually improved and the final class remaining is the one associated to the input pattern. However, the order of the successive networks involved in the classification decision has to be precised. To this aim we defined the α-paths in the NNIG. $\alpha \in [0, (n*(n-1))/2]$ and defined the maximum number of neural networks which can be used at each level i of the successive interrogations. At the first level, α initial neural networks are chosen and define the first requested networks. For each one of these latter, a class C_k is eliminated (which is not almost the same for all these networks since they do not discriminate the same classes). Furthermore, for each initial networks, a set of adjacent networks is defined by $A(\Re_{c_i,c_j})$. This set contains all the networks directly adjacent to the $\Re_{c_i,c_j}$ network that is to say all the other networks of the graph except the one considered: therefore it does not depend on the depth i.

For the classification, a constraint is precised: for each network, adjacent networks which are considered in $A(\Re_{c_i,c_j})$ must not contain all the previously eliminated classes since they have already been eliminated. The set of reachable adjacent networks ($RA(\Re_{c_i,c_j})$) is therefore a subset of $A(\Re_{c_i,c_j})$. Maximum α networks of the subset $RA(\Re_{c_i,c_j})$ are considered for the successive interrogations.

Definition 2. At a given depth i and for $\Re_{c_i,c_j}$ a given node in the NNIG, there are $(n-i)(n-i-1)/2$ networks which compose $RA(\Re_{c_i,c_j})$ the set of reachable adjacent neural networks.

Theorem 3. *At a given depth i, the number of reachable adjacent nodes which can be used in the α-path is given by*

$$\varphi(i) = \frac{(n-i)(n-i-1)}{2} \quad \textit{and if } \varphi(i) > \alpha \quad \textit{then } \varphi(i) = \alpha.$$

Corollary 4. *A α-path contains a total of* $\phi = \sum_{i=1}^{n-1} \phi_i$ *nodes.* ϕ_i *gives the total number of nodes at a given level i in the α-path for all the different branches and is defined by* $\phi_1 = \alpha$ *and* $\phi_i = \varphi(i-1) \times \phi_{i-1}$.

Corollary 5. *A α-path is composed of* ϕ_{n-1} *different branches.*

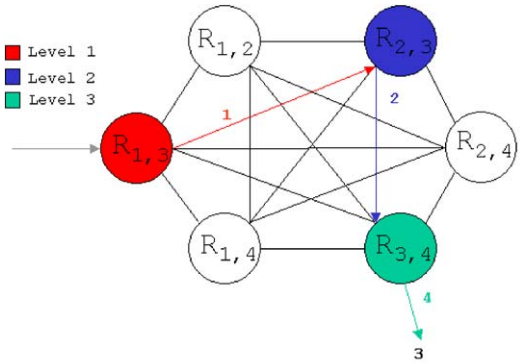


Fig. 2. Example of 1-path in an NNIG. The eliminated classes are shown on the links.

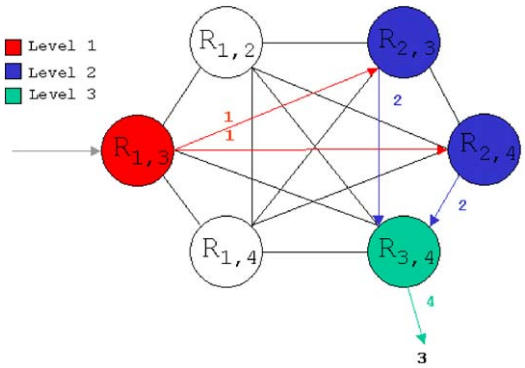


Fig. 3. Example of 2-path in an NNIG: first case, the two branches lead to the same final network $\Re_{3,4}$.

Fig. 2 presents a 1-path in an NNIG. Since $\alpha = 1$, only one network is considered at each level of successive interrogations of the path. The $\Re_{1,3}$ is considered as the initial network and the class 1 is eliminated (this is shown on the links). At this stage only networks which do not discriminate the class 1 are considered as reachable adjacent networks: $RA(\Re_{1,3}) = \{\Re_{2,3}; \Re_{2,4}; \Re_{3,4}\}$. The networks $\Re_{2,3}$ and $\Re_{3,4}$ are then used and eliminate respectively the classes 2 and 4. The input pattern is therefore designed as the class 3 since it is the last remaining one.

We can also study an example with a 2-path in the same NNIG graph. Two initial networks are chosen, each one of the latter eliminates one class and in their reachable adjacent networks two networks are chosen. The principle is repeated until only one class is available for each branch of the 2-path. Figs. 3 and 4 give an example of the branches defined by selecting two initial neural networks. The α-path generated is the following: $\{\{\Re_{1,3}; \Re_{2,3}; \Re_{3,4}\}, \{\Re_{1,3}; \Re_{2,4}; \Re_{3,4}\}, \{\Re_{1,4}; \Re_{1,2}; \Re_{2,3}\}, \{\Re_{1,4}; \Re_{2,3}; \Re_{1,3}\}\}$, it is constituted of $\phi_3 = 4$ different branches. For α-paths, with $\alpha > 1$, the

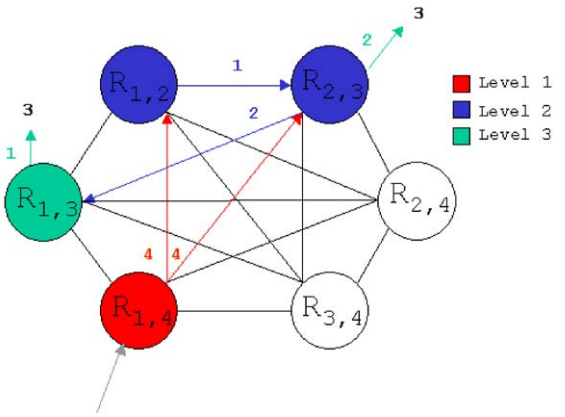


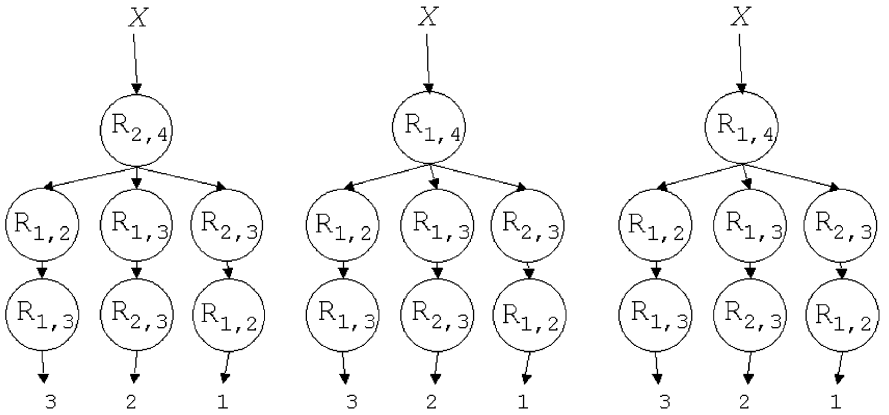
Fig. 4. Example of 2-path in an NNIG: second case, the two branches lead to different networks $\Re_{1,3}$ and $\Re_{2,3}$.

branches can use different networks and give the same final class using the same final network (first case, Fig. 3) or give the same final class using different final networks (second case, Fig. 4): it depends on the networks used. But there can be also different final classes obtained by using different branches (third case, to be further explained, Fig. 6). Therefore several key parameters needs to be defined:

- How to choose at each level of the α-path, the α neural networks?
- How to choose the final class for the input pattern if, for the different output leaves, several different final classes are associated to the input pattern ?

3.4.3. *Decision of classification*

To obtain the best decision of classification, the best neural networks as regards their ability to classify an input pattern can be used. Since at each level of a branch of the α-path, $\varphi(i)$ neural networks have to be chosen to build the latter, a quantitative measure is needed to make the choice and the *QI* associated to each neural network is used. For a node $\Re_{c_i,c_j}$, the α networks retained for the α-path are that one maximizing the *QI* criterion in the $RA(\Re_{c_i,c_j})$ subset. This defines a method to automatically build a α-path in the NNIG for an input pattern to classify. A branch is an ordered set of nodes and a node is called a leaf if it is the last of the branch (the final decision). Since a α-path is composed of several branches, several leaves are available. For each leaf corresponds a final class associated to the input pattern. Since the leaves are not necessarily identical for all the different branches, the α-path can associate several different final classes to the input pattern. Instead of choosing the leaf whose branch has the best *BQI*, we propose to choose the maximum of the sum of the decisions [2,10] associated to the different branches of the α-path. For each leaves of the α-path (denoted by $L_1,\ldots,L_{\phi_{n-1}}$) the *BQI* of the branches (denoted by $B_1,\ldots,B_{\phi_{n-1}}$) containing each leaf is associated. B_i leading to the same final class are summed and this implies the following definition.



C_i	1	2	3	4
$\Psi(C_i,X)$	2.37249	2.29464	**2.40577**	0.0

Fig. 5. Example of the developed branches for a 3-path in an NNIG for four classes with the maximum Ψ value bold faced.

Definition 6. For each class C_i, $\Psi(C_i,X)=\sum_{j=1}^{\phi_{n-1}} B_j(X)$ if $\theta(B_j,X)=C_i$ where $\theta(B_j,X)$ denotes the final class associated to the element X by the branch B_j.

The class C_k designed as the final class of an input pattern X is the one having the maximum $\Psi(C_k,X)$ value: $C_k = argmax_{i=1}^{n} \Psi(C_i,X)$. This technique enables to choose the more plausible class according to the purity of the different branches. Ψ depends on the value of α since the sum is performed over ϕ_{n-1} which depends on it. Figs 5 and 6 resume the different α-path builded for a four-class discrimination problem with $\alpha=3$. Maximum three networks are chosen at each level of the α-path. These networks are that one maximizing the QI value. $\phi_3 = 9$ different branches are developed in the NNIG. In the proposed example, three different classes are obtained by the different branches and the value of $\Psi(C_i,X)$ is used to choose the final class of the input pattern X. In this case, the class 3 is chosen since the sum of the decisions associated to the different branches which led to this particular decision of classification is maximum. Fig. 6 presents the α-path developed in the NNIG and Fig. 5 presents the developed branches for each one of the α initial networks (this figure is given only for a better explanation using a tree-like structure).

4. Experimental results

The databases for which results will be presented here are real databases coming from the Machine Learning Data Repository of the UCI [1] and also from our own works on microscopical imaging [14]. These databases are used in various articles on

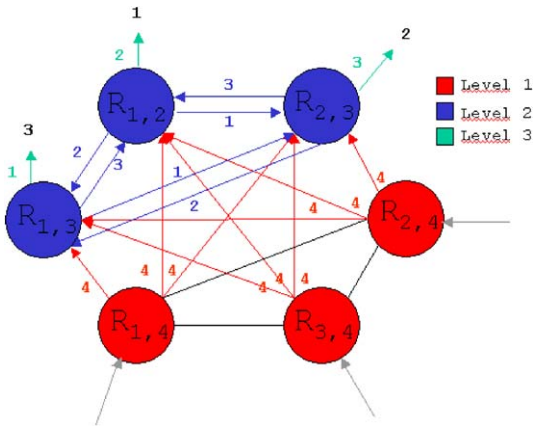


Fig. 6. All the possible developed branches in a 3-path NNIG for four classes: three possible choices as the final class.

Table 1
Databases used for the tests

Database	n	n_a	$\lvert S_T \rvert$	$\lvert S_{Test} \rvert$
Wine	3	13	144	34
Vehicle	4	18	679	167
Pageblocks	5	10	4382	1091
Segment	7	19	176	36
Glass	7	9	175	39
Shuttle	7	9	43500	14500
Pendigits	10	16	7494	3498
Optdigits	10	64	3065	760
Letter	26	16	16000	4000

classification. This will enable us to compare the performances of our architecture with the traditional MLP neural network approach.

4.1. Description of the databases

Table 1 summarizes the properties of each database. They correspond to very different problems (medical, segmentation of images, character recognition, wines, cars, etc.). Each database is characterized by the number of classes to discriminate (n), the number of attributes describing a pattern (n_a) and the number of instances in the training and test databases denoted by $|S_T|$ and $|S_{Test}|$ (see Table 1). Nine databases coming from the UCI were used. For each neural network $\Re_{c_i,c_j}$, the training is performed on S_T and the rate of classification further presented is measured on the test data base only (S_{Test}).

Table 2
Influence of α on the classification rate, the best rates are bold faced (precision of the results: 0.001)

α	1	2	3	4	5	6	7	8
Wine	**97.14**	97.14	97.14	97.14	—	—	—	—
Vehicle	68.45	**69.64**	69.64	68.45	69.05	69.05	—	—
Pageblocks	88.46	**88.55**	88.55	88.37	88.55	88.46	88.37	88.37
Segment	**91.67**	91.67	91.67	91.67	91.67	91.67	91.67	91.67
Glass	67.50	**70.00**	67.50	67.50	67.50	67.50	67.50	67.50
Shuttle	95.59	95.59	95.66	95.69	95.71	95.73	**95.75**	95.71
Pendigits	89.03	**89.14**	89.08	89.14	89.14	89.14	89.14	89.14
Optdigits	91.05	91.05	91.05	91.18	91.31	**91.44**	91.31	91.31
Letter	**76.97**	76.97	76.97	76.97	76.97	76.97	76.97	76.97

4.2. Result analysis

In this section, the influence of the α parameter on the generalization ability of the NNIG according to the databases is studied. Table 2 summarizes the results obtained with the variation of α and presents the rate of classification on the S_{Test} test databases. We recall here that $\alpha \in [0, (n * (n-1))/2]$ since at the first level of the NNIG, the maximum number of neural networks that can be used corresponds to the total number of networks of the NNIG. Table 2 is presented in the following way: the results are given on a line for each database and for different values of α. One can note first of all that the rate of classification does not progressively increase with α and that some variations occur along its evolution. But for each database, the best rate of classification is obtained before these variations appear: the rate of classification gradually increases with α until an upper limit then fluctuates. A second analysis of the results shows that the best rate of classification is obtained for various values of α according to the databases. In certain cases, $\alpha = 1$ is enough, which represents the use of only one branch in the NNIG to carry out the decision of classification: that implies a good generalization of the NNIG which provides few dubious cases as it will be further discussed.

We can now compare (see Table 3) the results obtained between a NNIG and traditional MLP (trained with the same method as the networks of the NNIG, see in [13]). The first column gives the database, the second the classification rate obtained with an NNIG, the third the classification rate with a traditional MLP, the fourth the learning time of the whole NNIG, the fifth the mean learning time over the networks of the NNIG, the sixth the learning time of the traditional MLP. Times are given in seconds. One can note that for all the bases, the NNIG makes it possible to obtain better results in all the cases ranging from simpler (few classes) to more complex (many classes) databases. The rate of classification of the NNIG used for the comparison corresponds to the best rate obtained among the various values of α. The NNIG performs between 0% and 17.5% better than a traditional MLP. But the NNIG has several other advantages [13]. As the number of classes increases the traditional MLP presents difficulties of generalization and its learning is longer. As compared to a traditional large MLP,

Table 3
Comparison between an NNIG and an MLP in terms of classification rate and learning time convergence with best rates bold faced

Database	NNIG	MLP	L. time	Mean l. time	MLP l. time
Wine	**97.14**	**97.14**	0.	0.	0.04
Vehicle	**69.64**	66.67	19.24	3.21	4.08
Pageblocks	**88.55**	84.80	26.77	2.68	5.92
Segment	**91.67**	80.56	6.26	0.3	4.57
Glass	**70.00**	52.50	9.12	0.43	6.92
Shuttle	**95.75**	79.15	132.19	6.29	44.3
Pendigits	**89.14**	83.82	75.69	1.68	76.18
Optdigits	**91.44**	90.39	104.75	2.33	138.49
Letter	**76.97**	62.45	1167.86	3.59	1680.68

Note: l. time = learning time.

Table 4
Breakdown of the study of uncertain cases for an NNIG

Database	Id-Classes	Correct Id-Classes	Correct Diff-Classes
Wine	100	97.14	—
Vehicle	99.40	69.46	100
Pageblocks	99.54	88.59	80
Segment	100	91.67	—
Glass	90	66.67	100
Shuttle	99.56	95.96	48.44
Pendigits	99.34	89.56	26.09
Optdigits	96.58	93.05	46.15
Letter	100	76.97	—

the NNIG structure is simpler, it enables incremental learning and its interpretation is easier since a branch provides a grading of the classification decision. Owing to the mean learning time of each network of the NNIG, it can be assessed that their learning is fast since their classification task is easier. The incremental learning is therefore of big interest since adding new data to the learning set implies only the learning of the involved network (which learn fast). The divide-and-conquer strategy employed coupled with a graph of neural networks thus proves to be very efficient for the classification of data. However no automatic method is provided to automatically find the best value of α, this will be further investigated.

We focus now on the study of the dubious cases for an NNIG. A dubious case corresponds to the appearance of several different final classes on the level of the leaves of an α-path.

Table 4 summarizes the study of the dubious cases. An α-path used to classify an input pattern X can bring to identical classes (pure case) or to different classes (dubious case). The use of the Ψ criterion makes it possible to manage the dubious cases. Table 4 gives the percentage of identical final classes (Id-Classes) obtained on the test databases for the best α-path. It is noted that the cases where all the final classes are identical always correspond to $\alpha = 1$. That is easily explained. If an α-path is pure (all

Table 5
The worst neural networks leading to bad decisions and penalizing the whole NNIG architecture

Database	Networks		
Wine	$\Re_{1,2}$	—	—
Vehicle	$\Re_{0,1}$	$\Re_{1,2}$	—
Pageblocks	$\Re_{0,1}$	$\Re_{0,2}$	—
Segment	$\Re_{2,4}$	$\Re_{2,3}$	—
Glass	$\Re_{0,1}$	$\Re_{0,2}$	—
Shuttle	$\Re_{1,4}$	—	—
Pendigits	$\Re_{1,2}$	—	—
Optdigits	$\Re_{1,8}$	$\Re_{8,9}$	—
Letter	$\Re_{18,25}$	$\Re_{6,16}$	$\Re_{5,15}$

the final classes obtained are identical), an increase of α will not influence the rate of classification on the test database. On the other hand if the α-path is more dubious (less pure cases), an increase of α makes it possible to better take into account the database and thus allows a better generalization. That explains why for some databases, $\alpha = 1$ is sufficient whereas for other ones it is necessary to gradually increase α to improve the management of the dubious cases. Table 4 also presents the rate of identical cases which were correctly classified (Correct Id-Classes). Indeed, an α-path for which all the paths bring to the choice of the same final classes can appear correct or not (the chosen class is not necessary the right one). The percentage of the correctly classified dubious cases is also presented (Correct Diff-Classes). Several remarks are essential. The rate of identical classes is generally high whatever the bases (higher than 96%), which reveals a certain homogeneity of the learning of the networks with respect to the data. On the other hand, one can note that all these identical classes are not inevitably well classified and one can suppose that an increase of α will change nothing. For the cases where the rate of identical classes correctly classified is weaker, the management of the dubious cases appears very effective and for the cases where the rate of identical classes correctly classified is high, it is much less efficient.

In order to explain why some pure cases are badly classified, it is interesting to know which neural networks eliminated the reference class of the input pattern for the identification of the networks performing detrimental classification errors. The detection of these networks identifies the weaknesses of the NNIG and enables to precisely point out the networks to be improved. This improvement necessarily implies an improvement of the learning database and a new learning of the selected neural networks. The advantage of the NNIG is that it allows an incremental learning [20]: the inducer is not completely rebuilded to learn from new data bringing further information and thus allows an easier improvement of the inducer performances. Table 5 gives an example of the worst neural networks retained for each database and having to be improved.

4.3. Feature selection

The networks used in the NNIG have a simple structure with a reduced number of neurons in the hidden layer. However, we can simplify these networks again, because

Table 6
Influence of the SFFS feature selection on the NNIG in terms of classification rate

Database	NNIG	SFFS
Wine	**97.14**	**97.14**
Vehicle	69.64	**75.00**
Pageblocks	88.55	**90.38**
Segment	91.67	**94.44**
Glass	**70.00**	**70.00**
Shuttle	95.75	**97.10**
Pendigits	89.14	**90.74**
Optdigits	91.44	**92.11**
Letter	76.97	**77.68**

Table 7
Comparison of the number of attributes for each neural network of the NNIG

Base	Initial	Min	Max	Mean	Std. dev.
Wine	13	13	13	13	0
Vehicle	18	4	10	7.66	2.05
Pageblocks	10	2	6	3.1	1.3
Segment	19	2	16	4.09	4.12
Glass	9	2	9	4.52	2.97
Shuttle	9	1	9	4.33	3.03
Pendigits	16	2	13	4.82	2.38
Optdigits	64	2	64	40.68	19.30
Letter	16	2	16	9.11	3.22

one of the remaining layers can be simplified: the input layer. Indeed, better results in term of classification rate can be obtained by removing the irrelevant attributes and therefore reducing the uncertainty. All the more, the relevant attributes can be selected for each neural network and therefore some attributes can be used in one network and not in the other ones: that makes it possible to know the relevance of each attribute according to the classes to differentiate. That concerns a well-known classification research topic: relevant feature selection. In this paper, the SFFS feature selection algorithm [15] has been used. This algorithm is known as wrapper heuristic [11] since the classification rate of the inducer is used to select the relevant features.

The results are given in Table 6. The latter gives the rate of classification using the NNIG alone and with the SFFS feature selection algorithm. The selection of attributes enables to increase once again the rate of classification of the NNIG. The feature selection improves the classification of the identical classes which were incorrectly classified and decreases the number of dubious cases by reducing the global uncertainty. To illustrate the utility of the selection of relevant attributes, Table 7 gives, for the SFFS algorithm and several databases, the minimum, the maximum, the mean and the standard deviation of the number of attributes used by all the networks of the NNIG. The selection of attributes is very useful since some networks finally have a very low

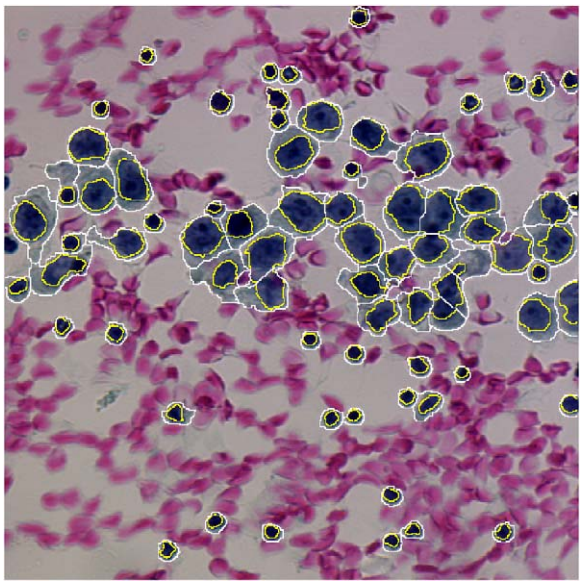


Fig. 7. A cytological color image and the corresponding segmentation.

number of features after the selection and for some databases, several attributes can be totally eliminated since they are irrelevant. The mean number of attributes is also lower than the initial number of attributes and that proves that for some networks few attributes are relevant. The feature selection for each network is therefore very interesting to simplify the complexity of each neural network and therefore of the whole NNIG.

4.4. Microscopical imaging

To illustrate the ability of the NNIG for pattern recognition, it is applied to the recognition of cells in serous cytology. In pathological anatomy and cytology, there are two types of examinations. The histology is the observation of the tissue and the cytology is the examination of a smear of cells. We are interested more particularly in the serous cytological examination. The samples are smeared over slides, fixed and colored in order to recognize the cells. Smears are examined under a microscope by a cytotechnician in order to locate cells of interest. That reading slide stage consists in a visual evaluation of the cells present on a cytological slide and is called screening. The goal of that stage is either the detection of abnormal or suspect cells, or the quantification of cells. That is thus of capital interest for the pathologist who must establish a reliable and valid diagnosis. We suggest to use the NNIG to build an automatic cellular sorting system for serous cytology. That system is called HERCULS (*HE*lp to the *R*esearch in cytology by *C*omputer cell *UL*ar *S*orting). Images are firstly segmented using color Mathematical Morphology operators (Fig. 7) and all the objects (the cells)

extracted in the images are described by 46 attributes ranging from size, shape, color and texture [14]. The choice of the attributes (i.e. descriptors) is very significant and is guided by the vision task which can be expressed as a priori information on the cells. For that experimentation, the various types of objects that can be met in serous cytology were indexed, that represents a rather important number of classes of cells to be recognized [13] and an NNIG can help in the recognition.

We are therefore interested in the isolated cells (which can be normal, abnormal or suspect cells). They must be distributed in the 18 different classes of objects. The isolated cells are classified by an NNIG with the SFFS feature selection method. The training of our architecture is carried out on a learning database of 3870 cells and tested on a database of 1967 cells. The total rate of recognition of the NNIG after the learning with feature selection by the SFFS wrapper method is 83.54% for the cells of the test database (the recognition rate of the NNIG without feature selection is 72.36% which is higher than the one obtained using a single large neural network: 55.90%). The percentage of identical classes is of 100% which reveals the homogeneity of the database. However, a work of extension of the base remains to be carried out in order to balance and to increase the number of cells of each class. Certain classes are few represented and thus under learned by the NNIG, which does not make it possible to give enough significant results for these classes. This can be verified by an analysis of the worst neural networks which correspond exactly to the difficulties encountered by the experts for the tagging of the objects. According to the aims of our system HERCULS, namely detection of abnormal or suspect cells, our system is very satisfactory because (after grouping of the same categories of cells) 94.5% of the abnormal cells and 99% of the normal cells are recognized, which greatly exceeds the success rate of an expert. From that point of view and since our system operates as an assistance to screening, one can think, within sight of the results obtained, that an abnormal cell omitted by the cytotechnician will be detected by the NNIG.

5. Conclusion

We suggested a new neural network architecture based on induction graphs and binary neural networks according to a divide-and-conquer principle. The properties of an NNIG for classification problems and mainly pattern recognition problems has been studied and this new architecture has proved its superiority compared to a traditional MLP. In addition to its strength of classification, an NNIG is able to identify and manage dubious decisions using the mean decision of several classification decisions. Another interest of the NNIG is their particular properties for incremental learning: the whole inducer is not totally rebuilded once new training data are available.

Future works will concern the extension of the NNIG to weighted NNIG. It might be interesting to add another step of learning in order to weight the links between the different networks of the NNIG. This will enable to better manage the uncertain cases by penalizing the worst networks of the NNIG. Other techniques coming from graph theory might be applied to the NNIG structure in order to automatically find

the best branch in the graph without an exhaustive exploration of all the different branches.

References

[1] C. Blake, C. Merz, UCI repository of machine learning databases (1998). URL http://www.ics.uci.edu/~mlearn/MLRepository.html.
[2] L. Breiman, Bagging predictors, Mach. Learning 24 (1996) 123–140.
[3] L. Breiman, J. Friedman, R. Olshen, Classification and Regression Trees, Wadsworth and Brooks, California, 1984.
[4] C. Campbell, Constructive Learning Techniques for Designing Neural Network Systems, Academic Press, San Diego, 1997.
[5] E. Diday, Optimisation en classification automatique Tomes 1 et 2, INRIA, Paris, France, 1979.
[6] D. Driankov, H. Hellendoorn, M. Reinfrank, An Introduction to Fuzzy control, Springer, Berlin, 1993.
[7] U. Fayyad, D. Haussler, P. Stolorz, Kdd for science data analysis: issues and examples, in: Proceedings of International Conference on KDD, Portland, OR, 1996, pp. 50–56.
[8] D. Fournier, B. Cremilleux, Using impurity and depth for decision trees pruning, in: Proceedings of EIS, Paisley, Scotland, 2000, pp. 320–326.
[9] J. Hérault, C. Jutten, Réseaux neuronaux et traitement du signal, Traité des nouvelles technologies (série traitement du signal), Hermès, Paris, France, 1994.
[10] J. Kittler, On combining classifiers, IEEE Trans. PAMI 3 (20) (1998) 226–239.
[11] R. Kohavi, G. John, Wrappers for feature selection, Artif. Intell., special issue on relevance 97 (1994) 273–324.
[12] T.-Y. Kwok, D.-Y. Yeung, Constructive algorithms for structure learning in feedforward neural networks for regression problems, IEEE Trans. Neural Networks 8 (3) (1997) 630–645.
[13] O. Lezoray, H. Cardot, A neural network architecture for data classification, Int. J. Neural Syst 11 (1) (2001) 33–42.
[14] O. Lezoray, A. Elmoataz, H. Cardot, M. Revenu, Arctic: an automatic system for cellular sorting by image analysis, in: Proceedings of Vision Interface, Trois-Rivières, Qué., Canada, 1999, pp. 312–319.
[15] P. Pudil, F. Ferri, J. Novovicovà, J. Kittler, Floating search methods in feature selection, Pattern Recognition Lett. 15 (1994) 1119–1125.
[16] R. Rakotomalala, Graphes d'induction, Ph.D. Thesis, Université Claude Bernard-Lyon I, 1997.
[17] A. Ribert, A. Ennaji, Y. Lecourtier, Building and evaluation of a distributed neural classifier, in: Proceedings of Vision Interface, Trois-Rivières, Qué., Canada, 1999, pp. 582–585.
[18] I. Sethi, J. Yoo, Structure-driven induction of decision tree classifiers through neural learning, Pattern Recognition 30 (11) (1997) 1893–1904.
[19] A. Sirat, J. Nadal, Neural trees: a new tool for classification, Network 1 (1990) 198–209.
[20] E. Stocker, A. Ribert, Y. Lecourtier, A. Ennaji, An incremental distributed classifier building, in: Proceeding of ICPR, Vol. 4, IEEE Computer Society Press, Silver Spring MD, 1996, pp. 128–132.
[21] D.A. Zighed, R. Rakotomalala, Graphes d'induction, Hermes Science publications, Paris, 2000.

O. Lezoray has received his M.Sc. degree in Computer Science in 1995 and the Ph.D. degree in Computer Science from the University of Caen in 2000. From September 2000 to August 2001 he worked as a 1 year assistant lecturer in the Computer Science Department of the University of Caen. Since September 2001, he joined the Communication Networks and Services Department of Technology of the University of Caen as a senior lecturer. His research focus on image segmentation techniques for color images and data classification methods based on the cooperation of machine learning methods. He is a member of the GDR-ISIS French national research group.

D. Fournier has received his M.Sc. degree in Computer Science in 1996 and Ph.D. degree in Computer Science from the University of Caen in 2001. Between September 2000 and August 2002 he was assistant lecturer at the Computer Science Department of the University of Caen. Last September, he was recruited at the University of Le Havre where he works now as Senior Lecturer in Computer Science. During his thesis, his research focused on machine learning and knowledge discovery applied to medical problems. Now, he has reoriented his interest on emergence in multi-agent systems.

H. Cardot is an engineer of the National School of Engineers of Caen and has a Ph.D. in signal processing. Until August 2003, he was a senior lecturer at the Electrical and Computer Science Department of Technology of the University of Caen. Since September 2003 he is a full professor at the Polytechnic School of Engineers of Tours and head of the Pattern Recognition and Image Analysis group of the LI research laboratory. His research focuses on neural networks and their applications in the fields of Image Processing.