ELSEVIER

Signal Processing 64 (1998) 271–290

SIGNAL
PROCESSING

Structured neural networks for signal classification

L. Bruzzone[a], F. Roli[b], S.B. Serpico[a,*]

[a] *Department of Biophysical and Electronic Engineering, University of Genoa, Via all' Opera Pia, 11A, I-16145, Genova, Italy*
[b] *Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi, I-09123, Cagliari, Italy*

Received 17 February 1997

Abstract

In this paper, artificial neural networks are considered as an emergent alternative to the classical 'model-based approach' to the design of signal-processing algorithms. After briefly examining the pros and cons of the neural-network approach, we propose the application of structured neural networks (SNNs) for the classification of signals characterized by different 'information sources', such as multisensor signals or signals described by features computed in different domains. The main purpose of such neural networks is to overcome the drawbacks of classical neural classifiers due to the lack of general criteria for 'architecture definition' and to the difficulty with interpreting the 'network behaviour'. Our structured neural networks are based on multilayer perceptrons with hierarchical sparse architectures that take into account explicitly the 'multisource' characteristics of input signals and make it possible to understand and validate the operation of the implemented classification algorithm. In particular, the interpretation of the SNN operation can be used to identify which information sources and which related components are negligible in the classification process. SNNs are compared with both commonly used fully connected multilayer perceptrons and the k-nearest neighbour statistical classifier. Experiments on two multisource data sets related to magnetic-resonance and remote-sensing images are reported and discussed. © 1998 Elsevier Science B.V. All rights reserved.

Zusammenfassung

Es werden künstliche neuronale Netzwerke als eine Alternative zur klassischen auf Modellen basierenden Vorgehensweise für den Entwurf von Algorithmen der Signalverarbeitung untersucht. Nach einer kurzen Besprechung der Vor- und Nachteile von Konzepten der neuronalen Netzwerke, wird die Anwendung strukturierter neuronaler Netzwerke (SNNs) zur Klassifikation von Signalen vorgeschlagen, die aus unterschiedlichen Quellen kommen, beispielsweise Multisensorsignale oder Signale, die durch Merkmale beschrieben werden, die in unterschiedlichen Bereichen berechnet wurden. Der hauptsächliche Zweck der Einführung solcher Netzwerke besteht darin, die Nachteile klassischer neuronaler Klassifikatoren zu umgehen. Diese rühren davon her, daß kein allgemeines Kriterium zur Definition der Architektur verfügbar ist, und daß die Interpretation des Netzwerkverhaltens Schwierigkeiten bereitet. Unsere strukturierten neuronalen Netzwerke beruhen auf Mehrschichtperzeptren mit hierarchischen schwach besetzten Architekturen, bei denen explizit die Mehrquellen-Charakteristik der Eingangssignale berücksichtigt wird und die es ermöglichen, die Operation der implementierten Klassifikationsalgorithmen zu verstehen und zu validieren. Die Interpretation der Funktionsweise der SNN kann insbesondere dazu verwendet werden festzustellen, welche Informationsquelle und welche damit zusammenhängenden Komponenten im Rahmen des Klassifikationsprozesses vernachlässigbar sind. Die SNNs

[*] Corresponding author. Tel.: + 39-10-3532752; fax.: + 39-10-3532134; e-mail: vulcano@dibe.unige.it.

0165-1684/98/$19.00 © 1998 Elsevier Science B.V. All rights reserved.

PII S0165-1684(97)00195-3

werden verglichen sowohl mit den üblichen voll vernetzten Mehrschichtperzeptren als auch mit dem statistischen k-ten Nachbarklassifikator. Es wird über Experimente mit zwei Mehrquellen-Datensätzen von Bildern aus dem Bereich der Magnetresonanz und der Fernerkundung berichtet und diskutiert. © 1998 Elsevier Science B.V. All rights reserved.

Résumé

Dans cet article, les réseaux de neurones artificiels sont considérés comme étant une nouvelle alternative à l'approche classique basée sur une modélisation pour la conception d'algorithmes de traitement du signal. Après un bref examen des avantages et des inconvénients de l'approche basée sur les réseaux de neurones, nous proposons l'application de réseaux de neurones structurés (SNNs) pour la classification de signaux caracterisés par différentes 'sources d'information', tels que les signaux multi-capteurs ou les signaux décrits par des paramètres calculés dans des domaines différents. Le but principal de tels réseaux de neurones est de surmonter les désavantages des classificateurs neuronaux classiques dûs à une absence de critère général pour la 'définition d'architecture' et à la difficulté d'interpréter le 'comportement du réseau'. Nos réseaux de neurones structurés sont basés sur des perceptrons multi-couches avec des architectures hiérarchiques clairsemées qui tiennent compte explicitement des caractéristiques 'multi-sources' des signaux d'entrée et rendent possible la compréhension et la validation de la mise en oeuvre de l'algorithme de classification implementé. En particulier, l'interprétation de la mise en oeuvre du SNN peut être utilisée afin d'identifier quelles sources d'information et quels composants associés sont négligeables dans le processus de classification. Les SNNs sont comparés avec les deux perceptrons multi-couches entièrement connectés généralement utilisés et avec le classificateur statistique des k plus proches voisins. Les expériences sur deux ensembles de données multi-sources associées a la résonance magnétique et aux images de télé-détection sont présentées et discutées. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: Signal classification; Pattern recognition; Neural networks

1. Introduction

Signal processing (SP) has its theoretical background in quantitative sciences, like mathematics, physics, and engineering, from which the classical 'model-based approach' (named 'parametric approach' [16]) to problem solving has been derived. In particular, in the SP field, this problem solving approach has been traditionally applied to the design of SP algorithms. Accordingly, the design of an SP algorithm aimed at performing a given processing task is usually carried out by using a two-stage procedure. The first stage is devoted to the definition of a mathematical model for the physical signal-generating 'system'. This model is then used to derive a mathematical procedure (the algorithm) that should constitute an optimal solution to the processing problem faced. Unfortunately, the model-based approach to the design of SP algorithms is often hampered by the lack of a realistic knowledge about the signal-generating system. In addition, models that allow one to obtain mathematically derived algorithms need simplified assumptions about the system (e.g., linearity, stationarity, and Gaussianity assumptions that are often not verified in real cases). Therefore, SP algorithms designed according to the model-based approach may provide suboptimal solutions to processing problems.

Artificial neural networks (ANNs) constitute an emergent alternative to the above designing approach, as they allow one to strongly simplify the two designing stages concerning the model definition and the mathematical derivation of the SP algorithm [16,18,23]. In particular, these two stages can be simultaneously performed by ANNs.

With regard to the model-definition stage, ANNs allow one to completely avoid the definition of a mathematical model based on a priori assumptions about the signal source. ANNs provide very general models (called 'nonparametric' models) that can be 'adapted' to the specific characteristics of a signal-generating system. To this end, specific 'training' algorithms can be used to tune the free parameters of a neural network model. In addition, the same capability of ANNs to adapt the neural network model to the system characteristics also allows one 'to train' the neural network to perform the desired SP task. Examples of target SP tasks are provided as inputs to the neural network. Therefore, ANNs can learn SP tasks, while taking implicitly into account the possible nonlinearity, non-stationarity and non-Gaussianity characteristics of real systems. It is worth noting that no explicit derivation of the SP algorithm from a mathematical model is therefore necessary.

However, the ANN approach to SP algorithm design exhibits some important drawbacks:

(i) The ANN operation is fundamentally based on a process of 'non-parametric statistical inference'; consequently, ANNs suffer from the well-known problems related to such a statistical process [17]. In particular, the algorithms used by ANNs to learn the desired SP tasks need a large number of training examples in order to provide satisfactory performances. Unfortunately, in real situations, available training sets are often small. Therefore, the ANN training procedure sometimes produces SP algorithms that perform well on training data but that are poorly effective when applied to data not contained in the training set (in the neural-networks literature, this is the so-called 'generalization' or 'overfitting' problem [17]);

(ii) The ANN learning capability allows one an adaptive modelling of the signal-generating system without any need for an exact knowledge of the system characteristics (e.g., linearity, stationarity, and Gaussianity). On the other hand, it is quite difficult to incorporate into the training algorithm the information about the system characteristics that is available to the SP algorithm designer;

(iii) The solution to the SP task learned by an ANN is actually embedded in the network parameters (i.e., the network 'weights'). This makes it very difficult to 'understand' and, consequently, to 'validate' the appropriateness of such a solution; therefore, it can strongly reduce the reliability of the SP algorithm used. Such an 'opacity' of neural-based algorithms can seriously limit end-users' confidence in these algorithms, in particular, when algorithm performance cannot be assessed on large data sets.

In this paper, a kind of structured neural networks (SNNs) is proposed that is mainly aimed at overcoming drawbacks (ii) and (iii) within the context of the classification of signals coming from different 'information sources', like multisensor signals and signals described by features computed in different domains [24,35]. Such SNNs are based on hierarchical architectures that allow one to easily incorporate the information available about the multisource characteristics of the signal considered. In addition, SNNs enable the SP expert to understand and validate the operation of the neural classification algorithm. The proposed SNNs also partially solve the above drawback related to the generalization capability of the network, as they allow a good compromise between architecture complexity and number of ANN parameters [2].

2. A short overview of neural networks for signal processing applications

In recent years, ANNs have been applied to accomplish many different SP tasks. The rationale of the use of ANNs has mainly been related to their above-mentioned advantages over the model-based approach to SP algorithm design. In addition, ANNs exhibit some other properties (e.g., massively parallel processing and fault tolerant characteristics) that make their use very interesting for SP applications. A lot of work has been reported on the applications of ANNs to SP [8,21,27–29,41]. An extensive review of ANN applications to SP is beyond the scope of this paper. We refer the interested reader to the Proceedings of the 1991–1996 IEEE Workshops on Neural Networks for Signal Processing [29]. A good selection of papers dealing with the applications of ANNs to different SP tasks is also given in the September 1990 and October 1996 issues of the Proceedings of the IEEE [27,28]. The latter also contains a paper by Rao and Protopopescu that provides a new theoretical insight into the reasons for successful applications of ANNs [31].

In the following, a short overview of ANN applications to SP is provided.

At the beginning, ANNs were applied to classical SP problems related to the recovering of an analog signal after transmission over a noisy or dispersive communication channel (e.g., dispersion over telephone channels, time-invariant or very slowly varying). Widrow's pioneering work dealt with channel equalization and adaptive signal filtering [42]. ANNs were also applied to many other SP tasks involved in communication systems (e.g., design of receivers, signal detection, signal parameter estimation, error correction codes, etc. [6,26,40]). All such papers pointed out the advantages of the ANN 'non-parametric nature' and the drawbacks arising when few training data are available.

ANNs were also applied to the processing of many different kinds of one-dimensional signals. Ukrainec and Haykin reported the use of ANNs for the processing of radar signals and discussed the related pros and cons [39]. Gorman and Sejnowski applied ANNs to the task of classifying sonar returns from undersea targets, and discussed the ability of neural networks to correctly classify unknown cases [14]. Many authors proposed neural-based algorithms devoted to automatic speech processing tasks. In particular, Haton's paper pointed out the potentialities of ANNs for automatic speech recognition [15]. Chen pointed out the important role played by ANNs in improving the accuracy of non-destructive evaluation (NDE) of materials by processing ultrasonic signals [5].

More recently, ANNs were applied to various image-processing tasks [8]. With regard to 'low-level' image-processing tasks, Yin et al. proposed the use of ANNs to design adaptive image filters [43]. Zhou et al. described a neural-network model especially devoted to image restoration [44]. Some authors reported experimental results on the use of ANNs for automatic segmentation of images [1,16]. Many papers dealt with the classification of images by ANNs [10,19]. ANNs proved especially useful for image classification tasks for which a parametric approach is difficult to design due to the lack of an exact knowledge about the statistics of the 'data classes' to be recognized. Many papers also dealt with the compression of images by using ANNs [4]. Five papers on image-based applications of neural networks were published in the October 1996 issue of the Proceedings of the IEEE [28]: three deal with medical-imaging applications, one with fingerprint classification and one with image compression.

Outside the context of the SP field, ANNs were proposed as a general approach to machine vision [12,38].

3. The proposed approach to signal classification

3.1. Introduction

In this paper, we propose a signal classification technique based on the pattern recognition approach. According to this approach, classification is carried out after mapping an input signal into a 'feature space'. To this end, the signal is decomposed into 'segments' and a set of attributes ('feature vector') is computed to characterize the different signal segments. Each segment is regarded as a 'pattern' and the related feature vector corresponds to a point (or vector) in the feature space. A classifier is then utilized to assign each pattern to one among a set of predefined classes (Fig. 1). Typically, this assignment is based on the estimation of the probability density functions of the feature vectors of each class in the feature space. A critical point in the pattern recognition approach is the definition of an appropriate set of features to ensure a satisfactory separation of the patterns of the different classes in the feature space [36].

Let us consider, as a particular signal processing task, the classification of 'multisource signals' into a set of M predefined classes, denoted by $\omega_i \, (i = 1, \ldots, M)$.

By 'multisource signals', we mean signals characterized by different 'information sources': we define an information source as a set of one or several features utilized to characterize (a segment of) a signal. Usually, such features are provided directly by a sensor (e.g., the amplitude and phase of a signal), or they are attributes computed by an algorithm from a signal (such as, for example, a set of frequency domain features). If an information source is made up of several features, it can be represented as a vector, where a single feature corresponds to a 'component' of the information source. In general, we may adopt the following notation:

$$\begin{aligned} \boldsymbol{X}^i &= (\boldsymbol{X}^i_1 | \ldots | \boldsymbol{X}^i_H) \\ &= (X^i_{11}, \ldots, X^i_{1C_1}, \ldots, X^i_{H1}, \ldots, X^i_{HC_H}), \end{aligned} \tag{1}$$

where $\boldsymbol{X}^i$ is the feature vector utilized to characterize the ith segment of the multisource signal; $\boldsymbol{X}^i_j$ is

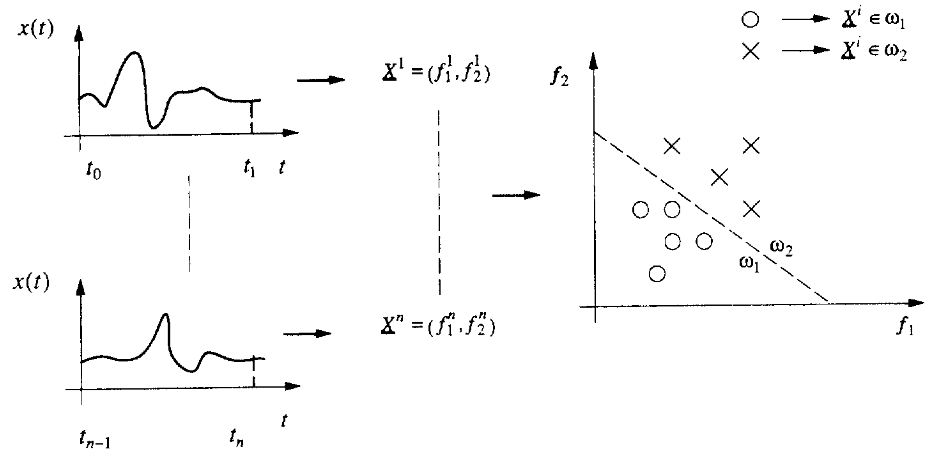

Fig. 1. Classification of a one-dimensional signal by using the pattern recognition approach: a simple example is shown where two features, f_1 and f_2, are computed for each segment of the signal $x(t)$, and the classification into two classes, ω_1 and ω_2, is performed by linear separation in the feature space.

the projection of X^i in the subspace related to the features of the jth information source; X^i_{jk} is the kth component (i.e., the kth feature) of the jth information source; H is the number of available information sources; C_j is the number of components of the jth information source.

The kind of classifier we propose to adopt is based on multilayer perceptron neural networks [17] with particular architectures, which will be described in the next subsection.

Multilayer perceptrons (Fig. 2) are among the most widely used supervised non-parametric neural classifiers. For the kind of signals we consider, it is especially useful to adopt a non-parametric classifier because, when dealing with signals from different sensors or characterized by quite heterogeneous features, it is often difficult to define a statistical model of the distribution of the patterns to be classified.

According to the supervised classification paradigm, a 'training set' is required for which the desired behaviour of the classifier is known: a set of patterns, with the related feature vectors and the information about the classes they belong to. A learning algorithm is used to train the adopted classifier to classify the training set correctly (to some acceptable approximation). Afterwards, such a classifier can be applied to the unknown patterns derived from the signals that have to be classified.

During the training phase, the learning algorithm (in our case, the well-known error backpropagation algorithm [34]) computes appropriate values of the parameters the classifier depends on in order to adapt to the characteristics of the training set. The training phase can therefore be considered as a process of parameter estimation from noisy data: in order to obtain reliable estimates, the number of training patterns should be much larger than the number of parameters to be estimated [2]. For a multilayer perceptron, the number of parameters to be estimated corresponds to the number of connections plus the number of neurons in the network, as a weight per connection and a bias per neuron are to be estimated (Fig. 2).

An evaluation of the performances of a supervised classifier can be obtained as follows. The set of known patterns is first subdivided into two distinct sets: a training set and a 'test set'. After training with the training set, the supervised classifier is applied to the test set (which should not be used during the training phase) and the accuracy yielded on such a set is used as an estimate of the classifier performances. This process may be repeated several times, with different training and test data, and the average accuracy is taken as an estimate.

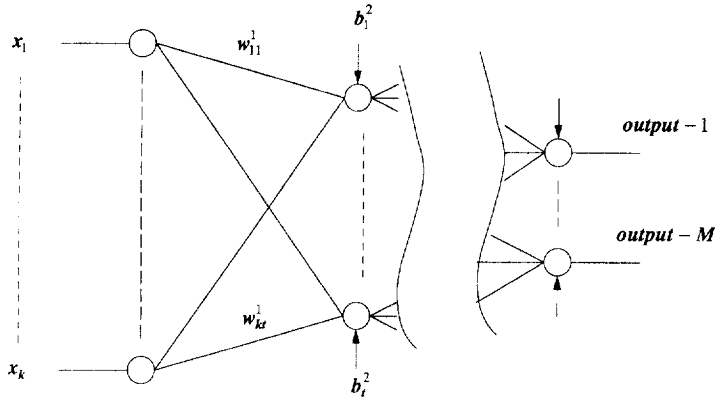

Fig. 2. A multilayer perceptron neural network. (w_{ij}^l denotes the weight of the connection between the ith neuron of the lth layer and the jth neuron of the $(l+1)$th layer; b_j^l denotes the bias of the jth neuron of the lth layer).

3.2. Structured neural networks and tree-like networks

The classifier we propose is based on a set of multilayer perceptron neural networks, one for each class, and on a 'winner-takes-all' (WTA) block that makes the final decision about classification (Fig. 3). We use this global architecture (which, in the following, is referred to as a 'structured neural network' – SNN) to implement the Bayes rule for the minimum classification error [11]. To this end, we train each neural network to estimate the posterior probability $p(\omega_i/X)$ of a class ω_i, given the feature vector X of a pattern [18]. The WTA block assigns each pattern to the class whose posterior probability estimate is the maximum one:

$$\hat{p}(\omega_k/X) = \max_i \{\hat{p}(\omega_i/X)\} \quad \Rightarrow \quad X \in \omega_k. \tag{2}$$

The class-related networks in Fig. 3 are called 'tree-like networks' (TLNs) for their architectures, which are defined below. Each TLN provides a non-parametric model that can approximate the posterior probability of a class in the feature space. This approximation is computed as a non-linear function of the feature vector X and depends on the network architecture, on the input/output response of neurons (the so-called 'activation function') and on the network parameters (weights of connections and biases of neurons).

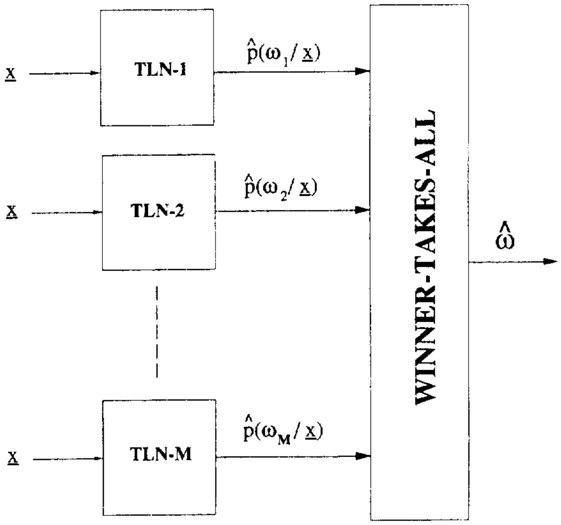
x
TLN-1
$\hat{p}(\omega_1/\underline{x})$
x
TLN-2
$\hat{p}(\omega_2/\underline{x})$
x
TLN-M
$\hat{p}(\omega_M/\underline{x})$
WINNER-TAKES-ALL
$\hat{\omega}$

Fig. 3. Architecture of an SNN for a number of data classes equal to M: just as many TLNs are used (i.e., multilayer perceptrons with particular tree-like architectures) as the data classes.

As is usually done for MLPs, as a neuron activation function, we use a sigmoid function:

$$S(x) = (1 + \mathrm{e}^{-x})^{-1}, \tag{3}$$

which is a nonlinear function that takes on values over the range 0 to 1. The output o_i^l of the ith neuron of the lth layer is then

$$o_i^l = S\left(\sum_k \omega_{ki}^{l-1} o_k^{l-1} + b_i^l\right). \tag{4}$$

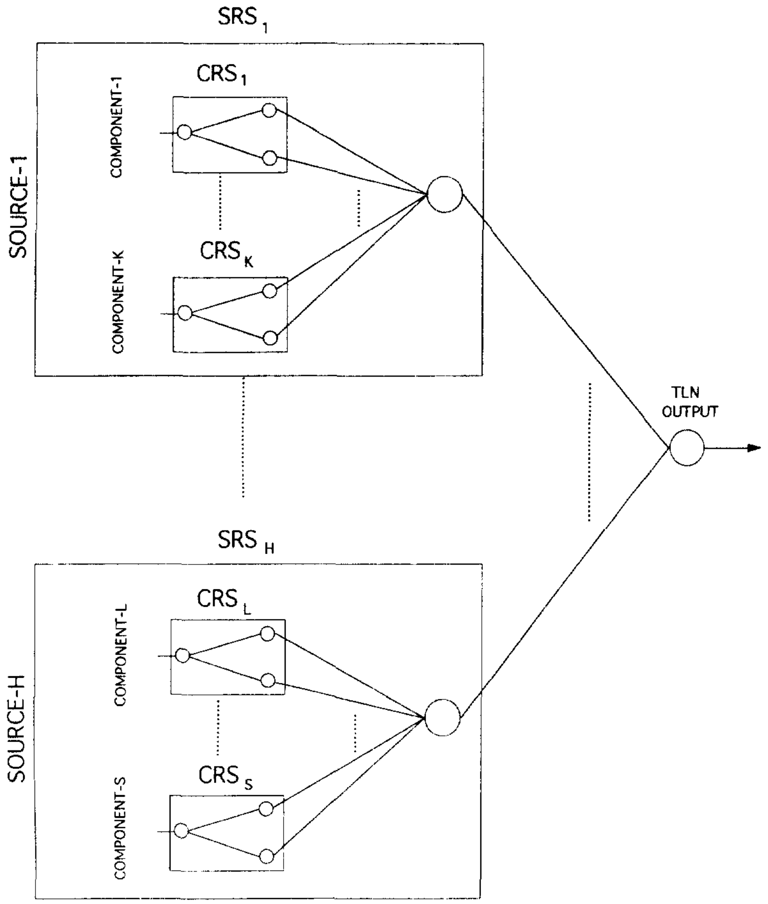

Fig. 4. TLN architecture based on source-related subnets (SRSs) and component-related subnets (CRSs).

where we have adopted the notation given in Fig. 2, that is, b_i^l stands for the bias of the ith neuron of the lth layer, and w_{ij}^l stands for the weight of the connection between such a neuron and the jth neuron of the $(l+1)$-th layer. The output of a TLN can be computed by propagating, from the input layer up to the output layer, the contributions of the neurons of a layer toward the neurons of the next layer.

All the TLNs in Fig. 3 have the same architecture, which is tailored to the classification task considered. Such an architecture is not fully connected and includes an input layer, two hidden layers, and an output layer (Fig. 4). A TLN architecture may be hierarchically decomposed into an output neuron and a set of just as many subnets as the information sources available to describe a signal. In turn, each source-related subnet (SRS) may be decomposed into an output neuron and just as many component-related subnets (CRSs) as the components included in the related information source. Finally, the CRSs we consider in this paper include an input neuron (corresponding to a source component) and two neurons in the first hidden layer (Figs. 4 and 5(a)).

The neuron activation function and the architecture are a priori fixed characteristics that determine the generality of the non-parametric model provided by each TLN. On the contrary, the network weights are free parameters that are tuned during the training phase to adapt the non-parametric

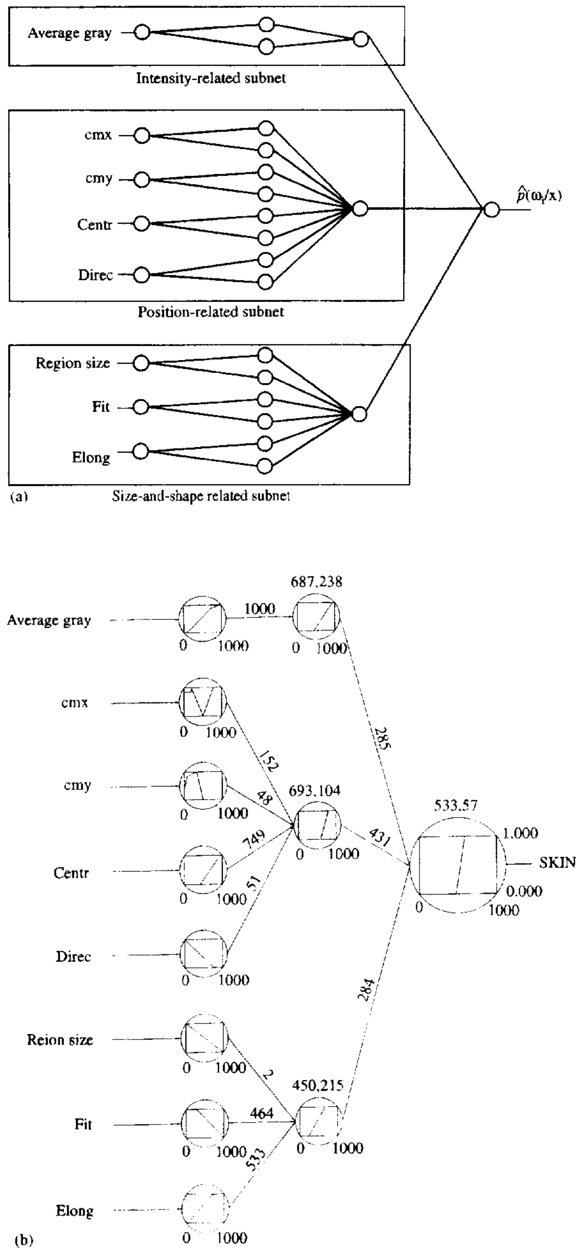

Fig. 5. TLNs used for the recognition of the magnetic-resonance images: (a) architecture based on the organization of features into three information sources; (b) simplified representation of the skin class-related TLN obtained by training with images 1, 2, 3, 4 and 5. Voting powers are shown on top of connections, whereas voting thresholds and delta votes are shown on top of neurons.

model to perform the classification task on the specific kind of signal considered.

TLN architectures appear to be interesting from several viewpoints, as specified in the following discussion.

They can be easily defined once the kinds of information sources used to characterize signals are known. This makes it possible to avoid the classical trial-and-error process for architecture definition [17].

The information derived from a component of an information source is first processed separately from all the other components, then the result obtained is merged with the results related to the other components of the same information source, at the level of the SRS output neuron. Similarly, we can observe that all the information derived from an information source is first processed separately from the other information sources inside an SRS, then the results obtained by all the SRSs are merged at the level of the TLN output neuron. This peculiarity of TLNs makes it possible to interpret the network behaviour, after appropriate transformations, as we explain in the next subsection. By contrast, in fully connected multilayer perceptrons, each neuron of a layer contributes to the inputs of all the neurons of the next layer. This hinders the efforts to interpret the network behaviour.

The above-mentioned separate processing of the information coming from different sources and from their components enhances the feature-selection capability of multilayer perceptrons. In addition, it makes it easier to evaluate the importance of the sources and the related components. Both these aspects will be reconsidered in Section 3.4.

The fact that the architecture is not fully connected limits the generality of the model provided by our SNNs but, on the other hand, it allows a significant reduction in the number of network parameters. In addition, in typical situations, the components of different information sources are quite independent. This reduces the importance of the loss of model generality due to the separate processing of such components inside each SRS because of the absence of connections towards the other SRSs. In conclusion, thanks to the use of a priori knowledge about the multisource characteristics of a signal, TLNs may provide an interesting compromise between the two requirements for the generality of the non-parametric model and a limited number of network parameters. This is useful, in particular, when dealing with small training sets.

3.3. Interpretation of the network behaviour

Each TLN is trained separately by the back-propagation algorithm, then it is combined with all the other TLNs by the WTA block to classify data. For the purpose of interpreting the network behaviour, a transformation is applied to each TLN in order to obtain a 'simplified representation' of the network [35]. This transformation consists of the following five steps, which are applied in sequence to each TLN:

(i) Each TLN is transformed into an equivalent network with positive weights. This procedure is not applied to the weights of the connections between the neurons of the input layer and those of the first hidden layer, as our method does not require that these weights be positive.

(ii) Inside the CRSs, each pair of neurons of the first hidden layer is replaced with an 'equivalent neuron' whose activation function is approximated by a piecewise-linear function. This function is made up of five (or fewer) linear pieces and is normalized so as to take on values over the range 0 to 1.

(iii) For all possible values of the inputs to a TLN, the outputs of some neurons of the second hidden layer may exhibit variations over a narrow subrange ([0,1] in our case) of the full output dynamics of these neurons. This step is aimed at avoiding an overestimation of the importance of the contributions of such neurons to the inputs to the neuron of the output layer. To this end, each TLN is transformed into an equivalent network in which the output of each neuron of the second hidden layer is expanded to the whole range of the neuron activation function (i.e. [0,1]), and the weights of the output connections from each of these neurons are reduced to compensate for the expansion.

(iv) A normalization procedure is applied to the weights of the input connections and to the biases of the neurons of the second hidden layer and of the output layer. As a result, an equivalent network is obtained in which the weights of all the connections entering such neurons are normalized by a multiplicative factor such that their sum may be equal to a prefixed positive value N ($N = 1000$, in our case); the biases of the above neurons are also normalized by using the same multiplicative factor as for the input weights to the neurons.

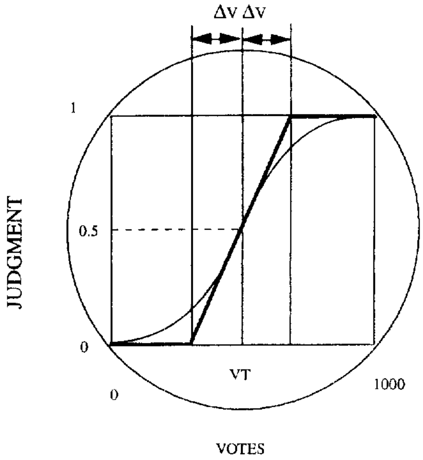

Fig. 6. The activation functions of the neurons of the second hidden layer and of the output neuron are interpreted as 'majority rules' of 'vote-taking units' and approximated by piecewise-linear functions. ΔV are the 'delta votes' and VT is the 'voting threshold'.

(v) Finally, the expanded activation functions of the neurons of the second hidden layer and the activation function of the output neuron are approximated by piecewise-linear functions (Fig. 6). In particular, we utilize the straight lines tangent to the (expanded) activation functions at the point where these functions are equal to 0.5. Then, we flatten these straight lines so as to obtain the same dynamics as the activation functions they approximate. This dynamics ranges from 0 to 1 for the neurons of the second hidden layer; a lower dynamics, corresponding to a subrange of 0 to 1, can be found for the output neuron. It is worth noting that the activation function of the output neuron is not expanded to avoid introducing modifications to the probability estimate it provides.

After this transformation, all the weights from the first hidden layer up to the output are made positive, normalized in a given range (i.e., 0 to

1000), and are such that the importance of the contribution of a neuron to the input to a neuron of the next layer corresponds directly to the weight of the connection between the two neurons. In addition, the activation functions of all the neurons (including the above-mentioned equivalent neurons) are approximated by piecewise-linear functions. Within the limits of the piecewise-linear approximations, the above transformation saves the input-output response of the TLN.

The simplified representation of a TLN (see, for example, Fig. 5(b)) makes it possible to interpret the network as a hierarchical arrangement of committees that judge the hypothesis that a pattern belongs to a given class.

The input neurons provide the information about all the source components to the next layers of the network. Each equivalent neuron of the first hidden layer expresses its judgment on the basis of a constraint on the values of an information-source component. This equivalent neuron is a member of a 'source-related committee' (SRC) that corresponds, from the viewpoint of the network architecture, to a source-related subnet. Each neuron of the second hidden layer is a 'vote-taking unit' (VTU) of an SRC, that is, it combines the votes of the members of its SRC. In addition, the neurons of the second hidden layer are members of the 'global committee' whose votes are taken by the neuron of the output layer to compute the final decision about the probability that a pattern belongs to a given class. The global committee corresponds to the whole TLN.

The importance of both information sources and of their components for the output of each TLN depends on their 'voting powers' in the committees they belong to, i.e., on the positive normalized weights of the connections between the VTUs of the SRCs and the output neuron, and between the equivalent neurons and the VTUs of the SRCs, respectively (Fig. 5(b)). The decision is made by each VTU on the basis of majority rules applied to the sums of the votes of all the members of the related committee. These rules are defined by piecewise-linear functions in the simplified representation (Fig. 6).

The representation described above allows a user to understand if the network operation is in agreement with his a priori knowledge and with the visual analysis of the input signal. If there is no agreement, the network may be trained again with different random starting weights. An example of network interpretation is given in Section 4.

3.4. *Selection of information sources and their components*

In this section, we consider the feature-selection capability of the proposed neural networks. This aspect is quite important, since, in general, the reduction in the number of features also reduces the number of free parameters of the classifier to be estimated. In addition, the variability of some features may not only provide no useful information for classification, but it may also affect classification accuracy by acting as noise.

The global 'one-net-one-class' scheme (Fig. 3) we adopted makes it possible to enhance the intrinsic capability of neural networks to select features by operating on various classes independently of one another. This makes a class-specific rejection of features possible, which can be very effective, especially when both classes and information sources are very heterogeneous.

The separate processing of the information coming from different sources and their components (see Section 3.2) enhances the feature-selection capability of multilayer perceptrons. In a TLN, the contribution of a source component may be selectively reduced not only by lowering the weights of the connections between the input and the first hidden layer, but also by lowering the weights of the layer of connections between the first and second hidden layers. Moreover, the reduction in the weight of a connection between a neuron of the second hidden layer and the output neuron may allow the contributions of all the components in an information source to be rejected altogether. In this case, the information source itself (a feature set or a sensor) may be disregarded.

It is worth noting that, for the purpose of deciding if an information source or a component may be removed, the behaviour of the majority rules must also be taken into account. In particular, the

ith source of a TLN can be disregarded, for classification purposes, if its votes do not change the output of the TLN appreciably. To this end, it is easy to show that the maximum change induced by the ith information source is about equal to

$$\mathrm{SC}_i = \min\{(\mathrm{VP}^2_{i1}/2\Delta V^3_1);\Delta J^3_1\}, \tag{5}$$

where VP^2_{i1} is the voting power of the ith information source; ΔJ^3_1 is the width ($\leqslant 1$) of the output range of the VTU of the global committee; ΔV^3_1 are the 'delta votes' of such a VTU. If one considers, for simplicity, ΔJ^3_1 to be equal to 1, then, according to the piecewise-linear approximation, $2\Delta V^3_1$ corresponds to the minimum number of votes to make the output of the VTU vary from 0 to 1 (Fig. 6), and Eq. (5) can be interpreted as follows: the maximum effect of the votes available to an information source can be observed in the central steepest part of the activation function of the output neuron (Fig. 6). According to the piecewise-linear approximation, the maximum induced change is equal to about the ratio between the number of votes available to an information source (i.e., VP^2_{i1}) and the minimum number of votes necessary to make the output of the VTU vary from 0 to 1 (i.e., $2\Delta V^3_1$). If this ratio exceeds the output range of the VTU (i.e., ΔJ^3_1), the maximum change is equal to the output range itself.

Analogously, the maximum change in a TLN output due to the votes of the jth component of the ith information source is about equal to

$$\mathrm{CC}_j = \min\{(\mathrm{VP}^1_{ji}/2\Delta V^2_i);1\} \cdot \mathrm{SC}_i, \tag{6}$$

where VP^1_{ji} is the voting power of the jth component of the ith information source and ΔV^2_i are the delta votes of the VTU of the ith SRC.

The two parameters SC and CC can be used to select information sources and the related components, respectively.

4. Experimental results

In order to assess the performances of the proposed SNNs, we considered two data sets related to different applications: one data set was composed of magnetic-resonance images, the other was composed of remote-sensing images. The choice of these two applications was suggested by the different types of signals involved that allow one to test the proposed SNNs under different conditions. For the magnetic-resonance data set, we defined three information sources, which corresponded to three different sets of features extracted from the images; in the case of the remote-sensing data set, we considered two information sources, which corresponded to the optical and radar sensors utilized for the image acquisition. In both cases, the performances provided by the proposed SNNs were compared with those of multilayer perceptron (MLP) neural networks and with those of one of the most widely used nonparametric statistical classifiers, i.e., the k-nearest neighbour (k-nn) classifier [9].

4.1. Magnetic-resonance image classification

A data set consisting of six magnetic resonance images was considered (Fig. 7). These images (256 × 256-pixel size) represented axial sections of the human head at eye level. We applied the classification process to the elementary regions obtained by segmenting the images. Therefore, we were able to exploit some useful features related to the position, the size and the shape of each elementary region. The images were preprocessed in order to reduce the influence of noise by applying an edge-preserving smoothing filter; then, they were segmented by using a region-growing algorithm [25] (Fig. 8). The segmentation threshold was automatically adjusted in order to have about 90 elementary regions for each image. An expert checked on the segmentation correctness and provided the correct recognition for each elementary region obtained by the segmentation. We considered three information sources to describe the elementary regions: the *intensity*, *position*, and *size-and-shape* sources. The components of each information source are given in Table 1.

Our experiments were aimed at recognizing thirteen classes: twelve classes associated with different organs (i.e., the nose, the left eye, the right eye, the left ocular fat, the right ocular fat, the skin, the chiasma, the left ear, the right ear, the mesencephalon, bone, and the brain) plus the

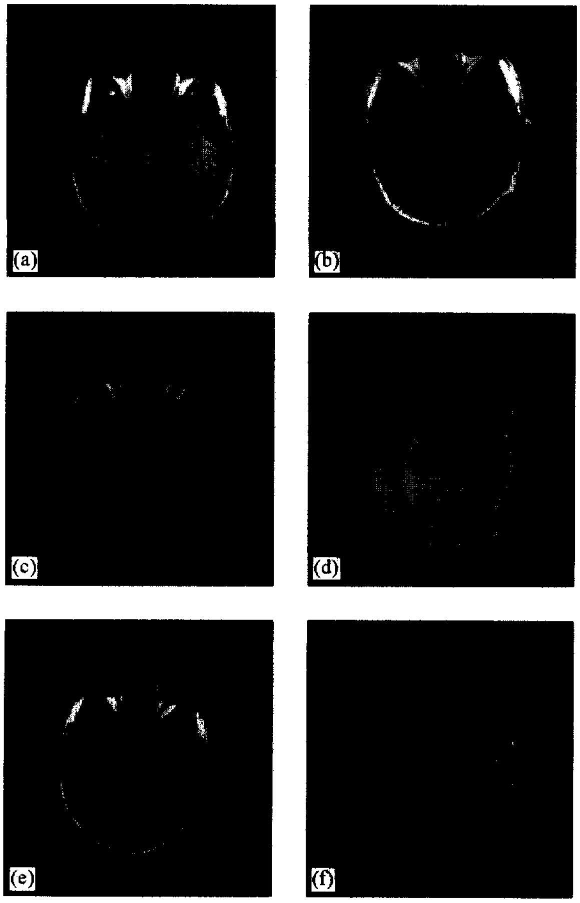

Fig. 7. The six magnetic-resonance images used in our experiments: (a) image 1, (b) image 2, (c) image 3, (d) image 4, (e) image 5, (f) image 6.

background. The feature vectors of the elementary regions were the inputs to the classifiers.

To plan the performance evaluation process, we took into account that, in medical applications, when a new image is to be classified, it is desirable not to ask for new training data from the new image. Consequently, we decided to separate the images into training and test images (including all the regions of an image either in the training set or in the test set, during the same experiment). In addition, we considered that, when few known patterns are available, rather than consider a single partitioning of known patterns into training and test sets, it is more reliable to repeat partitioning several times, then to average error rates [9]. Every time a different partitioning should be considered, including as many patterns in the training set as possible. According to these two observations, we decided to perform the training and test processes by adopting the k-fold cross validation method [37], with k equal to the number of images (i.e., $k = 6$). We performed the training process on the elementary regions of one of the possible image quintuples (about 450 patterns); then, we estimated the accuracies on the elementary regions of the left-out image (about 90 patterns). This procedure was applied to all possible k combinations. The average accuracy over the k testing processes was taken as a global accuracy estimate.

The SNN applied to the magnetic-resonance images had an architecture composed of thirteen TLNs (i.e., one for each class) and the WTA block. The TLNs had not fully connected architectures organized into three SRSs. The architecture of each TLN was of the '8-16-3-1' type, i.e., it was composed of 8 input units, 16 units in the first hidden layer, 3 units in the second hidden layer, and 1 output unit, as shown in Fig. 5(a).

The TLNs were initialized with random weights and trained by the error backpropagation learning procedure. As a convergence criterion, we required a mean square error (MSE) smaller than 0.0005. Learning was carried out at a learning rate $\eta = 0.01$ and was stopped when the convergence was reached or after 10.000 epochs. The learned TLNs were used to classify the test set data. To this end, the WTA rule was adopted in order to combine the outputs of the thirteen TLNs. The classification accuracies provided by the SNN on the six test sets considered in our experiments and the average accuracy are given in Table 2. Results are expressed in terms of percentage of correctly classified image area (this percentage was computed with respect to the pixels belonging to the head only, excluding the background, which is very easy to recognize and covers a large part of the image).

In order to compare the performances of the above SNN with those of fully connected MLPs, we carried out experiments by using four different MLP architectures. Also in this case, we applied the backpropagation learning procedure. The best

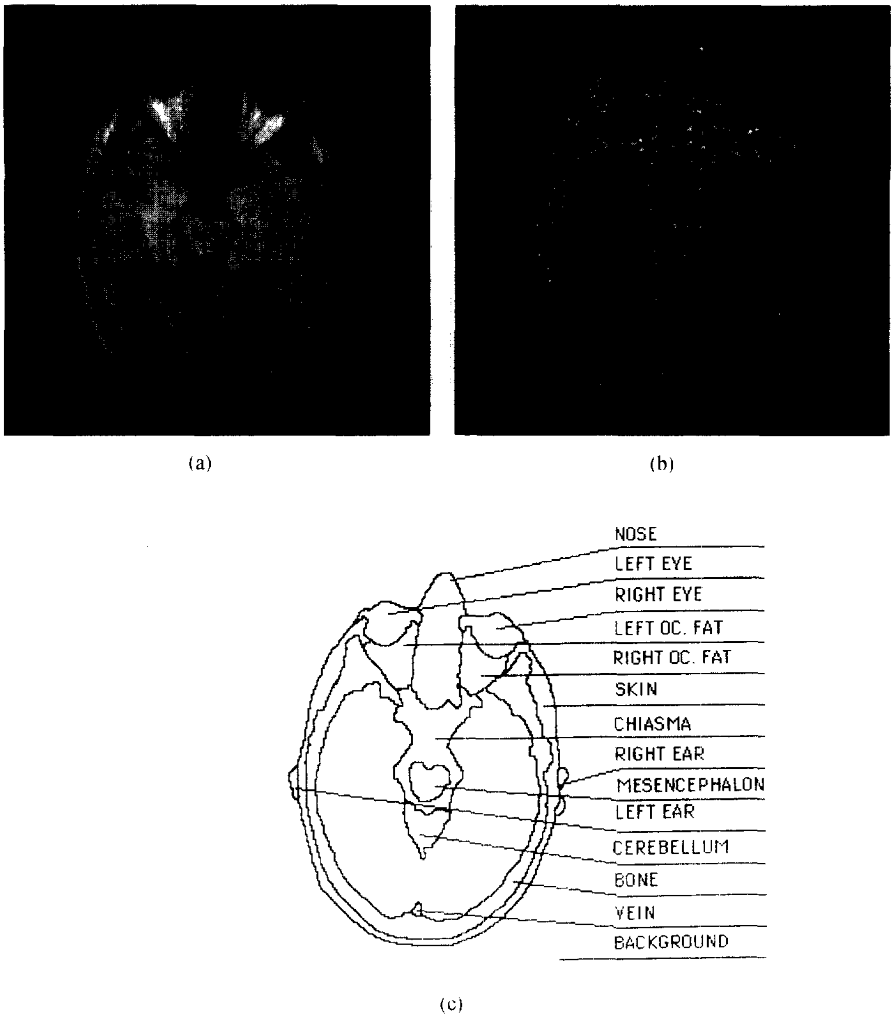

Fig. 8. Examples of: (a) original image; (b) segmented image (contours surround elementary regions); (c) recognized image (contours surround the groups of elementary regions that correspond to organs).

average accuracy, given in Table 3, was obtained by using an 8-6-13 architecture.

With regard to the k-nn classifier, we performed several trials to select the optimal value of k. Given the small number of samples available for some organs (e.g., the brain and the mesencephalon were segmented into a single region in all the images considered), the best average classification accuracies were obtained by small k values. In particular, the best average accuracy was obtained by $k = 3$ (see Table 3). We used $k = 5$ only for tie situations where there was no majority class in the set of 3-nearest neighbours of a pattern.

The classification accuracy provided by the SNN is much higher than those provided by both MLPs and the k-nn. This is due to the intrinsic feature-selection capability of SNNs (see Section 3.4) which makes them less sensitive to noisy information-source components, as compared with MLPs and the k-nn.

Table 1
Definitions of the information sources and the related components considered in the magnetic-resonance data set (MBR stands for 'minimum bounding rectangle')

Information source	Component	Definition
Intensity	*Average gray*	Mean gray-level value of the pixels belonging to the region
Position	*cmx, cmy*	Region centroid coordinates
	Centr	Polar distance between the region centroid and the slice centroid
	Direc	Orientation of the longest side of the MBR
Size-and-shape	*Region size*	Number of pixels belonging to the region
	Fit	Ratio between the region size and the size of its MBR
	Elong	Ratio between the longest and the shortest sides of the MBR

Table 2
Classification accuracies provided by the SNN classifier for the six test sets considered in our experiments on the magnetic-resonance image data set

Training set images	Test set image	Correctly classified image area
2 3 4 5 6	1	86.9%
1 3 4 5 6	2	88.6%
1 2 4 5 6	3	95.3%
1 2 3 5 6	4	83.7%
1 2 3 4 6	5	86.1%
1 2 3 4 5	6	96.7%
Average value		89.6%

Table 3
Average classification accuracies provided by the SNN, MLP (architecture 8-6-13) and k-nn (with $k = 3$) classifiers for the magnetic-resonance image data set

Classifier	Average value of correctly classified image area
SNN	89.6%
MLP	82.2%
k-nn	83.6%

In order to understand the roles of each component and of each information source in the classification process, after the training phase, we applied the transformations described in Section 3.3 to the obtained TLNs. As an example, let us consider the TLNs trained on the quintuple composed of images 1, 2, 3, 4 and 5. Fig. 5(b) shows the simplified representation of the TLN related to the skin class. In order to distinguish between the regions belonging to the skin class and the other regions, our network gives the greatest importance to the source related to *position* (i.e., VP = 431), whereas both the *intensity* and *size-and-shape* sources give smaller but non-negligible contributions (i.e., VP = 285 and VP = 284, respectively). Considering the voting threshold for the global committee (i.e., VT = 533), it is easy to observe that at least two information sources out of three should vote in favour of the 'skin' hypothesis in order to get a high output from this TLN. It is also interesting to observe the behaviours of the single information-source components in their source-related committees. As an example, in the *position*-related committee, the *centr* component has the highest voting power (i.e., VP = 749); the *cmx* component is smaller but not negligible (i.e., VP = 152), whereas the *direc* and *cmy* components give only small contributions (i.e., VP = 51 and VP = 48, respectively). By analyzing, for example, the constraint imposed by the equivalent neuron related to the *centr*

Table 4
SC and CC values computed for the magnetic-resonance data set. Stars indicate the source and components that may be disregarded

(a) SC values

TLN source	Chiasma	Left oc.fat	Right oc.fat	Bone	Left eye	Right eye	Skin	Left ear	Right ear	Nose	Brain	Mesen-ceph.	Back-ground
Intensity	0.879	1.000	1.000	0.731	0.753	1.000	1.000	0.000*	0.074*	1.000	0.953	0.653	0.009*
Position	1.000	1.000	1.000	0.731	1.000	1.000	1.000	1.000	1.000	1.000	0.953	0.999	0.003*
Size and shape	1.000	1.000	0.047*	0.731	1.000	1.000	1.000	1.000	1.000	1.000	0.953	0.073*	0.989

(b) CC values

TLN component	Chiasma	Left oc.fat	Right oc.fat	Bone	Left eye	Right eye	Skin	Left ear	Right ear	Nose	Brain	Mesen-ceph.	Back-ground
Average gray	1.000	1.000	1.000	0.731	0.753	1.000	1.000	0.000*	0.074*	1.000	0.953	0.653	0.009*
cmx	1.000	1.000	1.000	0.731	1.000	1.000	0.725	1.000	1.000	1.000	0.011*	0.155	0.000*
cmy	1.000	1.000	1.000	0.731	1.000	1.000	0.229	0.673	1.000	1.000	0.243	0.999	0.000*
Centr	1.000	1.000	1.000	0.731	1.000	0.784	1.000	0.185	1.000	1.000	0.953	0.999	0.003*
Direc	0.296	0.656	0.632	0.465	1.000	0.606	0.243	0.329	1.000	0.296	0.133*	0.163	0.000*
Region size	0.015*	0.036*	0.002*	0.089*	0.015*	0.026*	0.005*	0.007*	0.034*	0.015*	0.953	0.070*	0.989
Fit	1.000	0.731	0.044*	0.706	1.000	1.000	1.000	0.185	1.000	1.000	0.029*	0.001*	0.006*
Elong	1.000	1.000	0.002*	0.731	1.000	1.000	1.000	1.000	1.000	1.000	0.953	0.002*	0.047

Note: *Indicate the source and components that may be disregarded.

component, it is easy to observe that it shows a high activation only at the peripheral positions in the images. This constraint is in agreement with the typical position of the skin. Regarding the *size-and-shape*-related source, it is easy to see that the *elong* and *fit* components have high degrees of importance (i.e., VP = 533 and VP = 464, respectively), whereas the *size* is really negligible (i.e., VP = 2). Also in this case, the constraints of the equivalent neurons of the first hidden layer agree with the a priori knowledge on the application. Elementary regions belonging to the skin class usually have a high *elong* and a small *fit*, while the *size* is not useful to distinguish between the regions of the skin class and the other ones. The conclusions for the skin class are that all three information sources considered are important. However, two information-source components (*cmy* and *direc*) have low degrees of importance, whereas the *size*, having a negligible impact on the classification provided by the skin-related network, can be neglected at all.

Table 4 gives the SC and CC values computed for all the classes considered, as described in Section 3.4. These values can be compared to a threshold fixed on the basis of training data to decide which sources and which components can be disregarded for each class. In our experiments, we chose a threshold equal to 0.15. In Table 4, negligible sources and components are marked with stars. As can be seen, the *intensity* source is negligible for the left-ear, right-ear, and background classes; the *position* source is negligible for the background class; the *size-and-shape* source is negligible for the right-ocular-fat and mesencephalon classes. With regard to the source components, for example, Table 4 shows that, in order to identify the background class, only the *region size* component is sufficient. The *region size* component is also very useful to recognize the brain class, whereas it can be disregarded for all the other classes. This is in agreement with the medical knowledge on the segmented magnetic-resonance images.

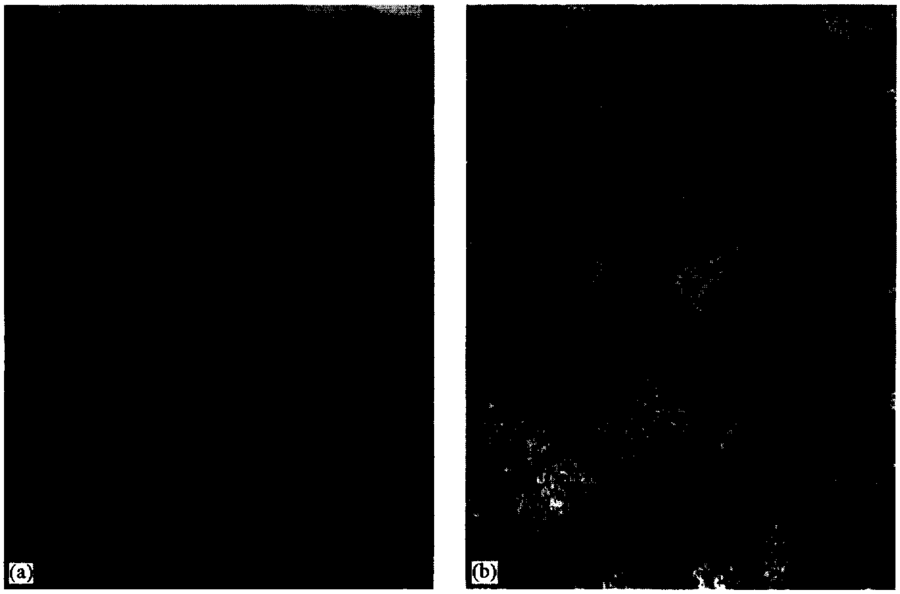

Fig. 9. Examples of images of the remote-sensing data set: (a) a band of the ATM sensor; (b) a channel of the SAR sensor.

4.2. Remote-sensing image classification

A set of multisensor remote-sensing images related to an agricultural area were considered [30]. Images were acquired with two sensors: a multiband optical sensor (an airborne thematic mapper (ATM)) with eleven bands [32], and a synthetic aperture radar (SAR) sensor with three bands and four polarizations for each band [30]. For our experiments, we selected six bands for the ATM sensor and nine channels for the SAR sensor. Fig. 9 shows an example of both the ATM and SAR images. The two sensors considered were associated with two information sources and each image was described as an information-source component.

In order to prepare a reference map aimed at assessing the classification accuracies provided by the classifiers, we digitized the available ground truth. For our experiments, we selected the five numerically most representative agricultural classes, i.e., sugar beets, carrots, potatoes, bare soil and stubble. The image classification process was carried out by considering each single pixel as an input pattern: the fifteen components described above were associated with each pixel to form a 'feature vector' to be used as input to the classifiers considered.

In the present case, a very large number of known patterns was available. Therefore, it was possible to estimate the accuracies provided by the classifiers by considering a single partitioning of the known patterns into a training set and a test set [9]. In particular, the agricultural fields were randomly subdivided into two sets: 5124 training pixels were taken from the fields of one set and 5820 test pixels from the fields of the other set.

The SNN used in our experiments on the remote-sensing data set was based on five TLNs, each with a 15-30-2-1 architecture (Fig. 10(a)). The TLNs were initialized with random weights and then trained by the backpropagation learning procedure. As a convergence criterion, we required a mean square error (MSE) smaller than 0.005. Learning was carried out at a learning rate $\eta = 0.01$ and was stopped when the convergence was reached or after 400 epochs. The class-by-class classification accuracies and the overall classification accuracy obtained by the SNN on the test set are given in Table 5.

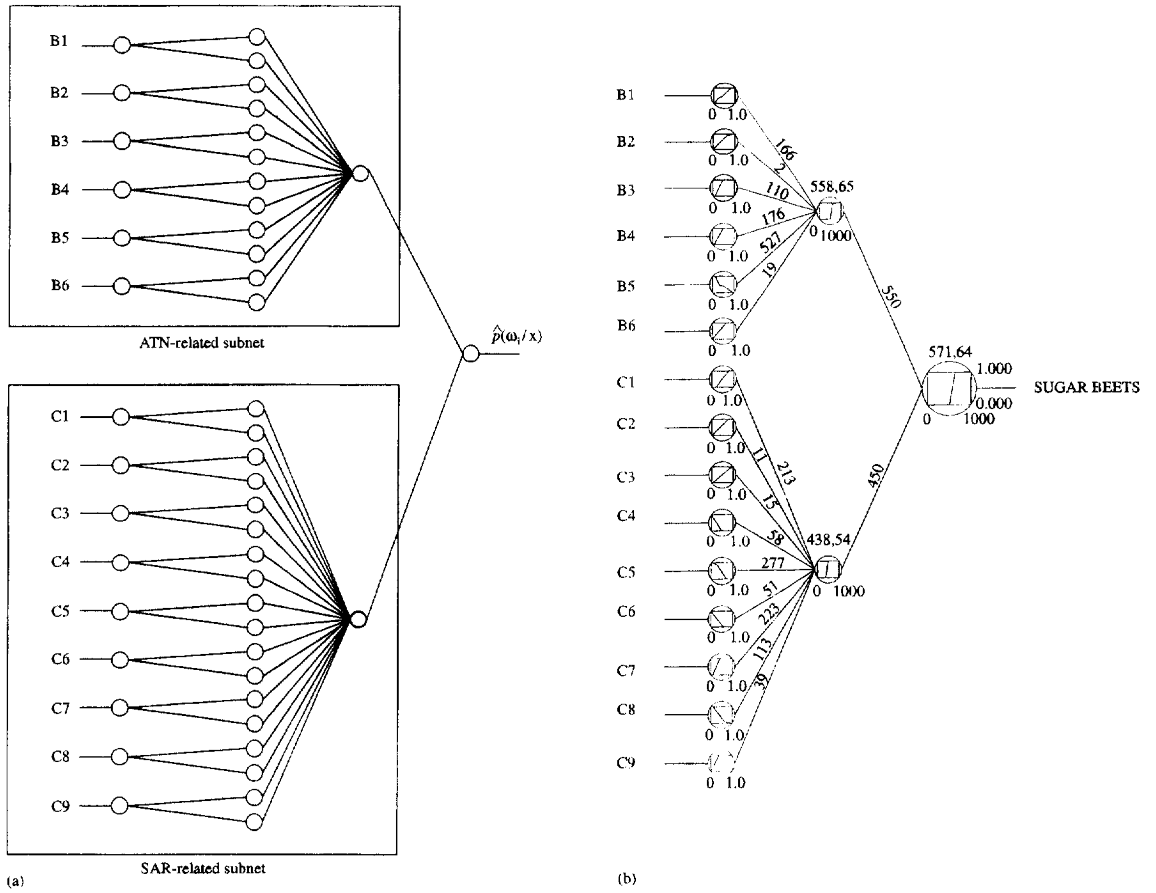

Fig. 10. TLNs used for the recognition of the remote-sensing images: (a) architecture based on the organization of features into two information sources; (b) simplified representation of the sugar beets class-related TLN.

With regard to the fully connected MLPs, we carried out experiments by using six different architectures. Table 6 gives the value of the best accuracy on the test set, which was obtained by the 15-8-5 architecture and $\eta = 0.01$.

Concerning the k-nn classifier, we performed several experiments to identify the best value for k. The best classification accuracy (see Table 6) was obtained by $k = 25$.

By comparing the three classifiers considered, it is easy to verify that the SNN provided a little lower accuracy than those achieved by both the MLP with the best architecture and the k-nn classifier (i.e., there were differences of 3.1% and 3.3%, respectively). However, unlike fully connected MLPs, SNNs allow one to interpret the network behaviour and to avoid the problem of architecture definition.

Also in this case, we carried out the transformations described in Section 3.3 on the trained TLNs in order to understand the network behaviour. As an example, Fig. 10(b) depicts the TLN obtained for the class 'sugar beets'. For an interpretation of the behaviour of such a TLN and, in general, for a more accurate description of the results obtained on this data set, we refer the reader to [35].

Table 5
Class-by-class classification accuracies provided by the SNNs for the remote-sensing image data set

Class	Clasification accuracy
Sugar beets	99.5%
Stubble	85.9%
Bare soil	79.3%
Potatoes	74.2%
Carrots	75.2%
Overall accuracy	86.5%

Table 6
Overall classification accuracies provided by the SNN, MLP (architecture 15-8-5) and k-nn (with $k = 25$) classifiers for the remote-sensing image data set

Classifier	Overall classification accuracy
SNN	86.5%
MLP	89.6%
k-nn	89.8%

5. Discussion and conclusions

In this paper, we have presented a kind of structured neural networks (SNNs) for the non-parametric supervised classification of multisource signals. These neural networks are characterized by hierarchical architectures that allow one to easily incorporate the information available about the multisource characteristics of the signal considered. In addition, they make it possible to understand the network behaviour and, consequently, to validate the appropriateness of the signal classification performed. The proposed SNNs have been compared with standard fully connected MLPs and with the statistical non-parametric k-nn classifier on both magnetic-resonance and remote-sensing data sets. It is worth noting that other neural models can also be used for a non-parametric signal classification, such as 'radial basis functions' and 'probabilistic neural networks' (a comparison between these classifiers and SNNs is made in [33]).

The main advantage of the proposed SNNs is the possibility of interpreting the network behaviour. In the literature, this problem has been usually addressed by developing new neural-network models [7,13,20,22]. Only very few works dealt with the interpretation of MLPs. In particular, Bishof et al. [3] proposed the 'weight-visualization' technique. However, due to the use of a fully connected architecture, this technique cannot separate the contribution of each component to each network output. By implementing a not fully connected neural network, our approach overcomes this difficulty and provides a clearer interpretation of the network behaviour.

The proposed technique can also be used to perform feature selection on the information sources and their components. As shown in Section 4, it is possible to quantify the contributions of features to the classification process as well as to identify features that provide no contributions at all.

In [35], SNNs were proposed for the more specific problem of the classification of remote-sensing images acquired with multiple sensors. In this paper, we have extended the SNN approach to the more general case of multisource signals. This case includes very common situations in the SP field, such as time signals and 2 D images for which the sources are obtained by computing various sets of features from the same signal (e.g., features from time and frequency domains). As an example, we carried out experiments on magnetic-resonance images for which three kinds of features (i.e., *intensity*, *position* and *size-and-shape* features) were considered as information sources. These experiments confirmed that the SNNs utilized reasonable criteria for classifying the input data. This may, in general, increase the confidence of the user in the classifier. Furthermore, the SNNs provided a much higher classification accuracy than MLPs and the k-nn. This may be ascribed to the fact that, for this data set, some information-source components exhibit, for some classes, a completely random behaviour that may be misleading in the classification process. The large number of classes (i.e., thirteen) and the presence of such misleading components favour the use of SNNs, which show a better capability of class-dependent feature selection. On the contrary, the k-nn classifier gives the same importance to all the features considered; therefore, it is necessarily influenced by misleading components.

The results obtained on the remote-sensing data set show a slight superiority of both MLP and k-nn classifiers over SNNs. These results can be explained by the different characteristics of such data. In particular, the smaller number of classes and the absence of the kind of misleading components described above do not allow SNNs to utilize all their feature-selection potential. However, also in this case, SNNs have the important advantage of allowing the network behaviour to be interpreted. Concerning the k-nn classifier, the availability of a large number of training patterns for all classes makes it possible to use larger k values, thus increasing the robustness of the classifier to noisy patterns.

Finally, SNNs are a good compromise between a model-based approach and a non-parametric approach. Although they are non-parametric classifiers that are capable of adaptively learning their parameters from a training set, they exploit the a priori knowledge on a multisource problem in order to define the network architecture and to validate the network operation.

Acknowledgements

The authors wish to thank Esaote Biomedica SpA (Italy) and Hunting Technical Services Ltd. (UK) for providing the magnetic-resonance images and the remote-sensing images, respectively.

References

[1] S.C. Amartur, D. Piraino, Y. Takefuji, Optimization neural networks for the segmentation of magnetic resonance images, IEEE Trans. on Medical Imaging 11 (2) (June 1992) 215–220.

[2] E.B. Baum, D. Haussler, What size network gives valid generalization, Neural Computation 1 (1989) 151–160.

[3] H. Bishof, A. Pinz, W.G. Kropatsch, Visualization methods for neural networks, in: Proc. 11th Internat. Conf. Pattern Recognition, The Hague, The Netherlands, 30 August–3 September 1992, pp. 581–585.

[4] S. Carrato, Neural networks for image compression, in: E. Gelenbe (Ed.), Neural Networks: Advances and Applications 2, Elsevier North-Holland, New York, 1992, pp. 177–198.

[5] C.H. Chen, Application of neural networks and fuzzy logic in nondestructive evaluation of materials, in: C.H. Chen (Ed.), Fuzzy Logic and Neural Network Handbook, McGraw-Hill, New York, 1996, pp. 20.1–20.8.

[6] J. Cid-Sueiro, A. Artés-Rodriguez, A.R. Figueiras-Vidal, Recurrent radial basis function networks for optimal symbol-by-symbol equalization, Signal Processing 40 (1994) 53–63.

[7] K.J. Cios, N.L. Ning, A machine learning method for generation of a neural network architecture: A continuous ID3 algorithm, IEEE Trans. on Neural Networks 3 (2) (March 1992) 280–291.

[8] Darpa Neural Network Study, AFCEA Int. Press, 1990.

[9] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

[10] M. Fukumi, S. Omatu, F. Takeda, T. Kosaka, Rotation-invariant neural pattern recognition system with application to coin recognition, IEEE Trans. on Neural Networks 3 (2) (March 1992) 272–279.

[11] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd edition, Academic Press, New York, 1990.

[12] K. Fukushima, A neural network for visual pattern recognition, IEEE Computer 21 (3) (March 1988) 65–75.

[13] R.M. Goodman, C.M. Higgins, J.W. Miller, P. Smyth, Rule-based neural networks for classification and probability estimation, Neural Computation 4 (1992) 781–804.

[14] R.P. Gorman, T. Sejnowski, Learned classification of sonar targets using massively parallel network, IEEE Trans. Acoust. Speech Signal Process. (Special Section on Neural Networks) 36 (7) (July 1988) 1135–1140.

[15] J.P. Haton, Neural networks for speech recognition, in: C.H. Chen (Ed.), Fuzzy Logic and Neural Network Handbook, McGraw-Hill, New York, 1996, pp. 14.1–14.16.

[16] S. Haykin, Neural networks expand SP's Horizons, IEEE Signal Processing Magazine 13 (2) (March 1996) 24–49.

[17] J. Hertz, A. Krogh, R.G. Palmer, Introduction to the Theory of Neural Computation, Addison Wesley, Reading, MA, 1991.

[18] D.R. Hush, B.G. Horne, Progress in supervised neural networks. What's new since Lippmann?, IEEE Signal Processing Magazine 10 (1) (January 1993) 8–39.

[19] A. Khotanzad, J. Lu, Classification of invariant image representations using a neural network, IEEE Trans. Acoust. Speech Signal Process. 38 (6) (1990) 1028–1038.

[20] R. Krishnapuram, J. Lee, Fuzzy-set-based hierarchical networks for information fusion in computer vision, Neural Networks 5 (2) 62–72.

[21] IEEE Transactions on Acoustics, Speech, and Signal Processing, Special Section on Neural Networks 36 (7) (July 1988).

[22] R.C. Lacher, S.I. Hruska, D.C. Kunicky, Back-propagation learning in expert networks, IEEE Trans. on Neural Networks 3 (1) (January 1992) 62–72.

[23] R.P. Lippmann, An introduction to computing with neural nets, Signal Processing Magazine 4 (2) (April 1987) 4–22.

[24] R.C. Luo, M. Gray, multisensor integration and fusion in intelligent systems, IEEE Trans. on Systems Man Cybernet 19 (5) (September/October 1989) 901–931.

[25] M. Nagao, T. Matsuyama, A Structural Analysis of Complex Aerial Photographs, Plenum, New York, 1980.

[26] V. Neagoe, Applying pattern recognition principles for intelligent detection of FSK signals, Signal Processing 32 (1993) 257–261.

[27] Proceedings of the IEEE, Special Issue on Neural Networks 78 (9) (September 1990).

[28] Proceedings of the IEEE, Special Issue on Artificial Neural Network Applications 84 (10) (October 1996).

[29] Proceedings of the 1991, 1992, 1993, 1994, 1995, 1996 IEEE Workshops on Neural Networks for Signal Processing, IEEE Press.

[30] Proc. Final Workshop MAESTRO-1 European SAR Campaign, ESTEC, Noordwijk, The Netherlands, March 1992, European Space Agency Publication.

[31] N.S.V. Rao, V.A. Protopopescu, On PAC learning of functions with smoothness properties using feedforward sigmoidal networks, Proceedings of the IEEE 84 (10) (October 1996) 1562–1569.

[32] J.A. Richards, Remote Sensing Digital Image Analysis, Second edition, Springer, New York, 1993.

[33] F. Roli, S.B. Serpico, L. Bruzzone, G. Vernazza, Classification of multispectral remote-sensing images by neural networks, in: Proc. of the VIII European Signal Processing Conference, Trieste, Italy, 10–13 September 1996, Vol. 1, pp. 172–175.

[34] D.E. Rumelhart, J.L. McClelland, and the PDP Research Group, Parallel Distributed Processing, Vols. 1–2, MIT Press, Cambridge, 1986.

[35] S.B. Serpico, F. Roli, Classification of multisensor remote-sensing images by structured neural networks, IEEE Trans. Geosci. Remote Sensing 33 (3) (May 1995) 562–578.

[36] S.S. Soliman, S.Z. Hsue, Signal classification using statistical moments, IEEE Trans. Commun. 40 (5) (May 1992) pp. 908–916.

[37] M. Stone, Cross validation choice and assessment of statistical predictions, J. Royal Statistical Society 36 (1974) 111–147.

[38] Toward an Artificial Eye, IEEE Spectrum, May 1996.

[39] A. Ukrainec, S. Haykin, A mutual information-based learning strategy and its application to radar, in: C.H. Chen (Ed.), Fuzzy Logic and Neural Network Handbook, McGraw-Hill, New York, 1996, pp. 12.3–12.26.

[40] G. Veciana, A. Zakhor, Neural net-based continuos phase modulation receivers, IEEE Trans. Commun. 40 (8) (August 1992) 1396–1408.

[41] B. Widrow, S. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.

[42] B. Widrow, M.A. Lehr, 30 Years of adaptive neural networks: Perceptron, madaline, and backpropagation, Proc. IEEE 78 (9) (September 1990) 1415–1442.

[43] L. Yin, J. Astola, Y. Neuvo, Adaptive neural filters, Proceedings of the 1991 IEEE Workshops on Neural Networks for Signal Processing, IEEE Press, pp. 503–512.

[44] Y.T. Zhou, R. Chellapa, A. Vaid, B.K. Jenkins, IEEE Trans. on Acoust. Speech Signal Process., Special Section on Neural Networks 36 (7) (July 1988) 1141–1151.