Parallel Consensual Neural Networks

Jon Atli Benediktsson, *Member, IEEE,* Johannes R. Sveinsson, *Member, IEEE,*
Okan K. Ersoy, *Senior Member, IEEE* and Philip H. Swain, *Senior Member, IEEE*

***Abstract*— A new type of a neural-network architecture, the parallel consensual neural network (PCNN), is introduced and applied in classification/data fusion of multisource remote sensing and geographic data. The PCNN architecture is based on statistical consensus theory and involves using stage neural networks with transformed input data. The input data are transformed several times and the different transformed data are used as if they were independent inputs. The independent inputs are first classified using the stage neural networks. The output responses from the stage networks are then weighted and combined to make a consensual decision. In this paper, optimization methods are used in order to weight the outputs from the stage networks. Two approaches are proposed to compute the data transforms for the PCNN, one for binary data and another for analog data. The analog approach uses wavelet packets. The experimental results obtained with the proposed approach show that the PCNN outperforms both a conjugate-gradient backpropagation neural network and conventional statistical methods in terms of overall classification accuracy of test data.**

***Index Terms*— Consensus theory, wavelet packets, accuracy, classification, probability density estimation, statistical pattern recognition, time-frequency analysis, data fusion.**

I. Introduction

CLASSIFICATION of data from multiple sources (multisource data) is an important research area which is related to data fusion. In multisource classification different types of information from several data sources are used for classification in order to improve the classification accuracy as compared to the accuracy achieved by single-source classification. Conventional statistical pattern recognition methods are not appropriate in classification of multisource data since such data cannot, in most cases, be modeled by a convenient multivariate statistical model. In [1] and [2], it was shown that neural networks performed well in classification of multisource remote sensing and geographic data. The neural-network models were superior to the statistical methods in terms of overall classification accuracy of training data. However, statistical approaches based on consensus from several data sources outperformed the neural networks in terms of overall classification accuracy of test data [3]. Our conclusion from these results is that it is desirable to combine certain aspects of statistical consensus theory approaches and neural networks. However, it is very difficult to implement prior statistical information in neural networks.

In this paper, the parallel consensual neural network (PCNN) is proposed as a network which does not use prior statistical information but is somewhat analogous to the statistical consensus theory approaches. In the PCNN, the input data are transformed several times and the different transformed data are fed into different neural networks (called stage neural networks or SNN's). The final output is based on the consensus among neural networks trained on the same original data with different representations.

In the PCNN, the input data can be both binary and analog. Both these data representations are discussed in the paper. For the analog representation, a time-frequency transform based on wavelet packets is introduced. In the PCNN, the outputs from the individual stage neural networks need to be weighted when consensus is computed. The question of how they should be weighted and optimized is addressed in the paper.

The paper begins with a short overview of consensus theory which is followed by a discussion on the PCNN. Then, optimization of the weights for the individual stages is discussed, followed by an overview of how to select input data transformations. This overview includes a brief review of wavelet packets. Finally, experimental results are given.

II. Consensus Theory

Consensus theory [3]–[8] is a well-established research field involving procedures with the goal of combining single probability distributions to summarize estimates from multiple experts (data sources) with the assumption that the experts make decisions based on Bayesian decision theory. Consensus theory is closely related to the method of stacked generalization [9] where outputs of experts are combined in a weighted sum with weights which are based on the individual performance of the experts. In most consensus theoretic methods each data source is at first considered separately. For a given source an appropriate training procedure can be used to model the data by a number of source-specific densities that will characterize that source [1]. The source-specific classes or clusters are therefore referred to as data classes, since they are defined from relationships in a particular data space. In general there may not be a simple one-to-one relation between the user-desired information classes and the set of data classes available since the information classes are not necessarily a property of the data. In consensus theory, the information from the data sources is aggregated by a global membership function and the data are classified according to the usual maximum

Manuscript received December 19, 1995; revised June 26, 1996. This work was supported in part by the Icelandic Research Council and the Research Fund of the University of Iceland.

J. A. Benediktsson and J. R. Sveinsson are with the Engineering Research Institute, University of Iceland, 107 Reykjavik, Iceland.

O. K. Ersoy and P. H. Swain are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA.

Publisher Item Identifier S 1045-9227(97)00238-5.

selection rule into the information classes. The combination formula obtained is called a consensus rule.

Consensus theory can be justified by the fact that a group decision is better in terms of mean square error than a decision from a single expert (data source). To show this, let us define an indicator function

$$I_{\omega_j} = \begin{cases} 1 & \text{if } \omega_j \text{ occurs} \\ 0 & \text{if } \omega_j \text{ does not occur} \end{cases}$$

where ω_j is an information class. Now it is needed to find an estimate, r, of the "best" probability that minimizes the mean square error (summed over all Z's)

$$\varepsilon^2_{\omega_j}(r) = \sum_Z (r - I_{\omega_j})^2 p(Z)$$

where $Z = [z_1, \cdots, z_n]$ is a compound vector consisting of observations from all the data sources, n is the number of data sources, z_i $(i = 1, \cdots, n)$ is an observation from a single data source (z_i can be a vector if the corresponding data source makes a multidimensional observation), and $p(Z)$ is the probability of Z. Differentiating $\varepsilon^2_{\omega_j}(r)$ with respect to r and setting the result equal to zero gives

$$\sum_Z 2(r - I_{\omega_j}) p(Z) = 0.$$

The solution to the above equation is $r = p(\omega_j|Z)$ which implies that the group probability $p(\omega_j|Z)$ is optimal for classification in the mean square sense.

Several consensus rules have been proposed. Probably the most commonly used consensus rule is the linear opinion pool which has the following (group probability) form for the information class ω_j if n data sources are used

$$C_j(Z) = \sum_{i=1}^{n} \lambda_i p(\omega_j|z_i) \tag{1}$$

where $p(\omega_j|z_i)$ is a source-specific posterior probability and λ_i's $(i = 1, \cdots, n)$ are source-specific weights which control the relative influence of the data sources. The weights are associated with the sources in the global membership function to express quantitatively the goodness of each source [5].

The linear opinion pool has a number of appealing properties. For example, it is simple, yields a probability distribution, and the weight λ_i reflects in some way the relative expertise of the ith expert. Also, if the data sources have absolutely continuous probability distributions, the linear opinion pool gives an absolutely continuous distribution. In using the linear opinion pool, it is assumed that all of the experts observe the input vector Z. Therefore, (1) is simply a weighted average of the probability distributions from all the experts and the result is a combined probability distribution.

The linear opinion pool, though simple, has several weaknesses [6]; e.g., it shows dictatorship when Bayes' theorem is applied, i.e., only one data source will dominate in making a decision. It is also not externally Bayesian (does not obey Bayes' rule). The reason it is not externally Bayesian is that the linear opinion pool is not derived from the joint probabilities using Bayes' rule. Another consensus rule, the logarithmic opinion pool, has been proposed to overcome some of the problems with the linear opinion pool. The logarithmic opinion pool can be described by

$$L_j(Z) = \prod_{i=1}^{n} p(\omega_j|z_i)^{\lambda_i} \tag{2}$$

where $\lambda_1, \cdots, \lambda_n$ are weights which should reflect the goodness of the data sources. Often it is assumed that $\Sigma_{i=1}^{n} \lambda_i = 1$.

In [7], the logarithmic opinion pool is given a natural-conjugate interpretation and it is shown that the logarithmic opinion pool differs from the linear opinion pool in that it is unimodal and less dispersed.

The logarithmic opinion pool treats the data sources independently. Zeros in the logarithmic opinion pool are vetos; i.e., if any expert assigns $p(\omega_j|z_i) = 0$, then $L_j(Z) = 0$. This dramatic behavior is a drawback if the density functions are not carefully estimated. The logarithmic opinion pool is externally Bayesian, but it is computationally more complicated than the linear opinion pool.

It is desirable to combine consensus theoretic approaches and neural networks since consensus theory has the goal of combining several opinions, and a collection of different neural networks should be more accurate than a single network in classification, at least in the mean square sense. Moreover, feedforward neural networks minimizing mean-square error at the output have been shown to approximate posterior probabilities, $p(\omega_j|z_i)$ when one output neuron is assigned to each class, ω_j [10]. Using this property, it becomes possible to implement consensus theory in the networks.

III. Neural Networks with Parallel Stages

Implementing consensus theory in neural networks involves using a collection of neural networks. This may be achieved by using neural networks with several parallel stages as depicted in Fig. 1. Each stage can be a particular neural network, here referred to as an SNN. Unlike a multilayer network, each SNN is essentially independent of the other SNN's in the sense that each SNN does not receive its input directly from the previous SNN. In the PCNN, the input data to the SNN's are obtained by applying a data transform (DT) to the original input vectors. Therefore, the stages are trained on different representations of the same input data. Each SNN has the same number of output neurons (equal to the number of data classes) and is trained for a fixed number of iterations or until the training procedure converges. When the training of all the stages has finished, the consensus for the SNN's is computed. The consensus is obtained by taking class-specific weighted averages of the output responses of the SNN's. Thus, the PCNN attempts to improve its classification accuracy by weighted averaging of the SNN responses from several different input representations. By doing this, the PCNN attempts to give highest weighting to the SNN trained on the "best" representation of input data.

Using the proposed PCNN architecture, it can be guaranteed that the PCNN should do no worse than single stage networks, at least in terms of training accuracy in the mean square sense. This is based on the argument in Section II that a group

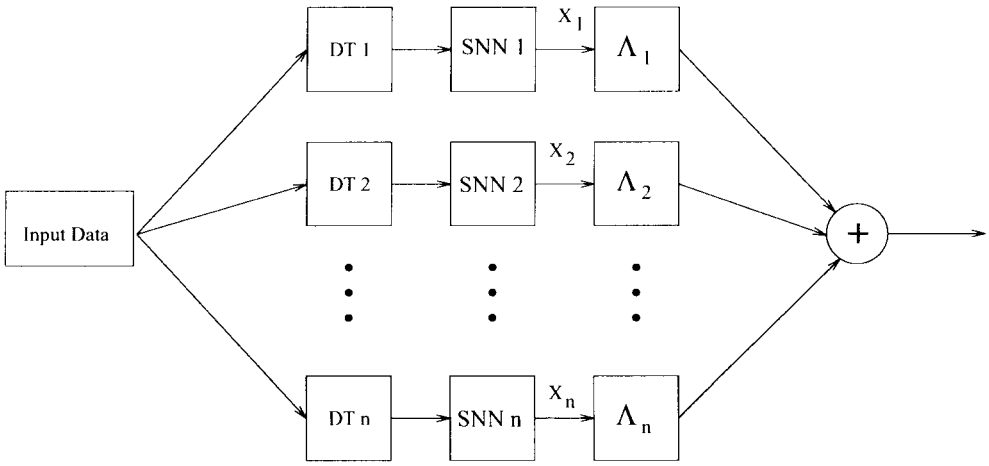

Fig. 1. The proposed PCNN with weighted individual stages.

decision is better in terms of mean square error than a decision based on the opinion of a single expert. To be able to guarantee such performance in classification of test data, cross-validation methods [9] can be used. Also, it has been shown [11] that if each of the networks in a collection of neural networks arrive at the correct classification with probability $1-p$ and the networks make independent errors, the probability of a sum of network outputs being in error is monotonically decreasing in n if $p<1/2$. This implies that using a collection of networks reduces the expected classification error if the networks make independent errors. However, the independence assumption is hard to justify in most cases.

It has also been shown [12] that the standard deviation of the classification of a collection of neural networks (such as the PCNN) decreases as the number of stage networks increase.

A. Related Neural-Network Models

Several methods have been proposed to combine multiple neural networks. In [13] and [14] it was shown that averaging separate networks improves generalization performance for the mean squared error. Tumer and Gosh [15] have also shown that substantial improvements can be achieved in difficult pattern recognition problems by combining or integrating the outputs of multiple classifiers. However, the earliest attempt at combining multiple networks can be credited to Nilsson [16] who proposed his committee machines based on a collection of single-layer networks as an attempt to design a multilayer neural network which could classify complicated data. Hansen and Salamon [11] discuss the application of an ensemble of multilayer neural networks. Their ensemble consists of several SNN's where each SNN receives the same input data, similar to Nilsson's committee machines. Each SNN is based on the backpropagation network, and the weights in different SNN's are initialized differently in order to avoid the same local minima for all the networks. The ensemble network makes the final decision (classification) based on the majority vote from all the networks. Alpaydın [17] proposed a similar architecture to the one in [11], offering the possibility of using different types of stage networks.

Battiti and Colla [18] have looked at several different ways of combining outputs of a set of neural-network classifiers, and Rogova [19] has combined the outputs of neural networks by Dempster–Shafer methods. Cho and Kim [20] have combined the outputs of multiple networks based on fuzzy logic.

The parallel self-organizing hierarchical neural network (PSHNN) proposed by Ersoy and Hong [21] is a neural network which is in some respects related to the PCNN proposed here. The PSHNN involves a self-organizing number of stages, similar to a multilayer neural network. At the output of each SNN, there is an error detection scheme. If an input vector is rejected, it goes through a nonlinear transformation before being input to the next SNN. This property is distinct from conventional neural networks. The PSHNN in [21] is based on using binary data. Deng and Ersoy [22], [23] have extended the PSHNN to apply it with analog inputs and outputs. The PSHNN is somewhat related to the method of "adaptive mixtures of local experts" [24] which is a multiple neural-network model where each network is trained on a subset of the training data. Valafar and Ersoy [25] have proposed a parallel self-organizing consensual neural network (PSCNN) which is related to the PSHNN. The PSCNN uses nonlinear transformations of the input data and creates accept and reject boundaries for each SNN in a similar fashion to the PSHNN. Pre- and postvoting are used to make decisions with the SNN's. The postvoting is in some ways similar to error boundaries in the PSHNN but is not related to consensus theory.

All the architectures discussed above are not based on consensus theory and do not offer any optimal way of computing the weights for the combination of stage networks. Of interest here is to base the total network on consensus theory, select appropriate data transforms for the inputs to different stage networks, and optimize the influence of the individual stage networks to maximize the overall accuracy in classification.

IV. Optimal Weights

The weight selection schemes in the PCNN should reflect the goodness of the separate input data, i.e., relatively high weights should be given to input data that contribute to high

accuracy. There are at least two potential weight selection schemes. The first scheme is to select the weights such that they weight the individual stages but not the classes within the stages. In this scheme one possibility is to use equal weights for all the outputs of the SNN's, $X_i, i = 1, 2, \cdots, n$, and effectively take the average of the outputs from the SNN's, i.e.,

$$Y = \frac{1}{n}\sum_{i=1}^{n} X_i$$

where Y is the combined output response. Another possibility in this scheme is to use reliability measures which rank the SNN's according to their goodness. These reliability measures might be, e.g., stage-specific classification accuracy of training data, overall separability or equivocation [1].

The second scheme is to choose the weights such that they not only weight the individual stages but also the classes within the stages. This scheme is depicted in Fig. 1. In this case, the combined output response, Y, can be written in matrix form as

$$Y = X\Lambda$$

where X is a matrix containing the output of all the SNN's and Λ contains all the weights. Assuming that X has full column rank, the above equation can be solved for Λ using the pseudoinverse of X or a simple delta rule. In order to find the optimal weights in Fig. 1, we define

$$X = [X_1 X_2 \cdots X_n]$$
$$\Lambda = \begin{bmatrix} \Lambda_1 \\ \Lambda_2 \\ \vdots \\ \Lambda_n \end{bmatrix}$$

where $X_i\ i = 1, \cdots, n$ are $r \times p$ matrices (r is the number of training samples, p is the number of outputs for each SNN). Each row of X_i represents an output vector for the ith SNN and $\Lambda_i, i = 1, \cdots, n$ are $p \times p$ matrices representing the weights for the ith SNN. If $Y = D$ is the desired output of the whole network we have

$$X\Lambda = D.$$

Λ is an unknown matrix, and its least square estimate Λ_{opt} is sought to minimize the squared error, i.e,

$$\Lambda_{\text{opt}} = \arg\min_{\Lambda} \|X\Lambda - D\|^2.$$

This is a well-known problem in linear regression, signal processing and adaptive filtering. The formula for Λ_{opt} uses the pseudoinverse of X, i.e.,

$$\Lambda_{\text{opt}} = (X^T X)^{-1} X^T D$$

where X^T is the transpose of X, and $(X^T X)^{-1} X^T$ is the pseudoinverse of X if $X^T X$ is nonsingular. In the case that X is not of full column rank, this solution becomes ill-conditioned. In that case one can use dummy augmentation to make X a full column rank matrix in a higher dimensional space and then solve the problem. There are at least two other suboptimal methods for solving this optimization problem. The rest of this section will be devoted to these methods.

The first method is to use sequential formulas to compute the optimal Λ [26]. Let the ith row vector of the matrix X be x_i^T and the ith row of the matrix D be d_i^T; then Λ can be calculated iteratively using the formula

$$\Lambda^{(i+1)} = \Lambda^{(i)} + P^{(i+1)} x_{i+1}(d_{i+1}^T - x_{i+1}^T \Lambda^{(i)})$$
$$P^{(i+1)} = P^{(i)} - \frac{P^{(i)} x_{i+1} x_{i+1}^T P^{(i)}}{1 + x_{i+1}^T P^{(i)} x_{i+1}} \qquad i = 0, 1, \cdots, r$$

where $\Lambda^{(r)}$ is the least squares estimate of Λ_{opt}. The initial conditions to the sequential formula are $\Lambda^{(0)} = 0$ and $P^{(0)} = \beta I$, where β is a positive large number.

The second method for solving the least squares error problem is to choose unitary Λ which minimizes $\|D - X\Lambda\|^2$ [27]. We compute

$$\|D - X\Lambda\|^2 = \|D\|^2 - 2\langle D, X\Lambda\rangle + \|X\|^2$$

where $\langle D, X\Lambda\rangle = \text{tr}(D\Lambda^T X^T)$ and tr returns the trace of its argument matrix. If

$$X^T D = V\Sigma U^T$$

is a singular value decomposition (SVD) of $X^T D$ then

$$\begin{aligned} \text{tr}(D\Lambda^T X^T) &= \text{tr}(X^T D\Lambda^T) \\ &= \text{tr}(V\Sigma U^T \Lambda^T) \\ &= \text{tr}(\Sigma U^T \Lambda^T V) \\ &= \sum_{i=1}^{p} \sigma_i(X^T D) t_{ii} \end{aligned}$$

where $T = [t_{ij}] = U^T \Lambda^T V$ is a unitary matrix and σ_i is the ith singular value of its argument matrix. This sum is maximized when all $t_{ii} = 1$, i.e.,when $\Lambda_{\text{opt}} = VU^T$.

V. Data Transforms

The major source of classification error in single stage neural networks is the nonseparability of the classes. To reduce or eliminate classification errors it is desirable to find a transformation which maps the input vectors into another set of vectors that can be classified more accurately. A variety of schemes can be used in the PCNN to transform the data. We shall consider two cases: binary input data and analog input data.

A. Binary Input Data

In the binary case, input vectors can be represented by a Gray code [2]. The Gray-code representation can be derived from the binary code representation in the following manner. If $b_1, \cdots, b_n$ is a code word in an n-digit binary code, the corresponding Gray-code word $g_1, \cdots, g_n$ is obtained by the rule

$$g_1 = b_1$$
$$g_k = b_k \oplus b_{k-1} \qquad k \geq 2$$

where $\oplus$ is the exclusive OR (XOR) operator. One simple possibility for a data transformation for the PCNN is to use

this scheme successively for the stages that follow [21]. This is done by looking at the Gray-coded input of the previous SNN as $b_1, \cdots, b_n$ and then taking the Gray code of the Gray code.

B. Analog Input Data

A general approach proposed for the transformation of analog input data is based on the wavelet packet transform (WPT). The wavelet packet transform [28] provides a transformation of a signal from the time domain to the frequency domain and is a generalized version of the wavelet transform [29]. The WPT is computed on several *levels* with different time/frequency resolutions.

The full WPT for a time domain signal can be calculated by successive application of low-pass and high-pass decimation operations. Let $h(k)$ and $g(k), k = 1, 2, \cdots, L$, be the finite low-pass and high-pass impulse responses for the WPT [28]. Let $x(m), m = 1, 2, \cdots, N$, denote the original time domain signal of finite length N, where $N = 2^{n-1}$. Define G and H as the operators which perform the convolution of $x(m)$ with $h(k)$ and $g(k)$, respectively, followed by a decimation by two (see Fig. 2). Then we have

$$\{Hx\}(m) = \sum_{k=1}^{L} x(k)h(2m-k)$$

$$\{Gx\}(m) = \sum_{k=1}^{L} x(k)g(2m-k)$$

for $m = 1, 2, \cdots, N$. Due to decimation, Hx and Gx each contain half as many samples as x. The operators H and G form a pair of quadrature mirror filters (QMF's) and satisfy the following orthogonality conditions:

$$HH^* = GG^*, \quad HG^* = GH^* \quad \text{and} \quad H^*H + G^*G = I$$

where I is the identity operator, H^* and G^* are adjoint operations of H and G, respectively. Various design criteria such as regularity, symmetry, etc., on the low-pass filter coefficients $h(k)$ can be found in [29]. Once the L-tap low-pass FIR-filter $h(k)$ is fixed, the L-tap high-pass filter can be found by $g(k) = (-1)^k h(L-1-k)$. In this work, the Daubechies four-point (D4) filters [29] were used for the WPT.

The WPT may be calculated using a recursion of the above mentioned filter decimation operations. The top level, called Level 0, of the full WPT contains the original time domain signal and thus has one bin. Level 1 of the WPT has two bins where the first bin contains Hx and the second bin contains Gx. The Level 1 representation has two degrees of frequency resolution, i.e., the low- and the high-frequency portions of the original signal have been separated into two bins, but due to decimation, each bin has only half the time resolution that exists at Level 0. Level 2 of the WPT contains four bins, where each bin contains sequences generated by the operations H^2x, GHx, HGx and G^2x. Hence, Level 2 has four degrees of frequency resolutions, but each bin has only half the time resolution that existed at Level 1. This process can be repeated $n-1$ times where $N = 2^{n-1}$. By proceeding down through the levels of the WPT, the tradeoff between time resolution and

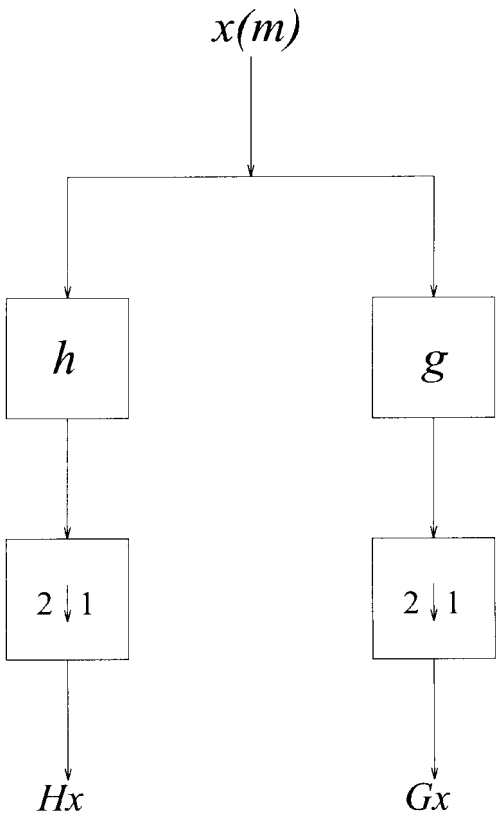

Fig. 2. Low-pass decimation and high-pas s decimation of a time domain signal.

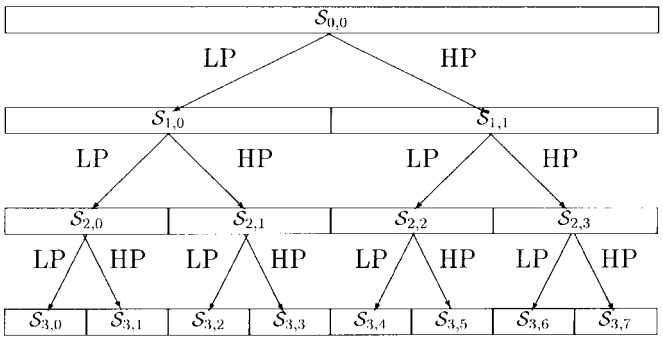

Fig. 3. Levels and bins for the full WPT.

frequency resolution is observed (see Fig. 3). The sequence, $\mathcal{S}_{l,b}x$, at bin b and level l of the WPT can be written as

$$\mathcal{S}_{l,b}x = H^*\mathcal{S}_{l+1,2b}x + G^*\mathcal{S}_{l+1,2b+1}x$$
$$\text{for } l = 0, 1, \cdots, n-1, \quad b = 0, 1, \cdots, 2^{n-1}-1$$

and the low-pass and high-pass filter decimations of the sequence $\mathcal{S}_{l,b}x$ are

$$\mathcal{S}_{l+1,2b}x = H\mathcal{S}_{l,b}x$$

and

$$\mathcal{S}_{l+1,2b+1}x = G\mathcal{S}_{l,b}x.$$

The WPT provides a systematic way for transforming the input data for the PCNN. Each level of the full WPT consists

of data for the different stage networks. Therefore, the stages will have the same original input data with different time-frequency resolutions. Thus, the PCNN attempts to find the consensus for these different representations of the input data, and the optimal weighting method will consequently give the best representation the highest weighting.

An advantage of the WPT is that it is not computationally intensive, i.e., the computations are $O(N \log_2 N)$. This property is very important for the PCNN, especially if the number of stages (n) is large.

VI. Experimental Results

Two experiments were conducted with the PCNN on multisource remote sensing and geographic data. The results of the experiments are discussed below.

A. Experiment 1: Colorado Data Set

The PCNN was used to classify a data set consisting of the following four data sources:

1) Landsat MSS data (four spectral data channels);
2) elevation data (in 10 m contour intervals, one data channel;
3) slope data (0–90° in one-degree increments, one data channel);
4) aspect data (1–180° in one-degree increments, one data channel).

Each channel comprised an image of 135 rows and 131 columns, and all channels were spatially coregistered. The area used for classification is a mountainous area in Colorado. It has ten ground-cover classes which are listed in Table I. One class is water; the others are forest types. It is very difficult to distinguish among the forest types using the Landsat MSS data alone since the forest classes show very similar spectral response [1]. Reference data were compiled for the area by comparing a cartographic map to a color composite of the Landsat data and also to a line printer output of each Landsat channel. By this method 2019 reference points (11.4% of the area) were selected comprising two or more homogeneous fields in the imagery for each class. Approximately 50% of the reference samples were used for training, and the rest were used to test the neural networks. Two versions of the PCNN were applied in classification of the Colorado data, i.e., PCNN with equal weights and with optimized weights. (The optimal approach reported here was the pseudoinverse method but the suboptimal methods gave similar results.) The PCNN algorithms were implemented using one-layer conjugate-gradient delta rule neural networks [2], [30] for the SNN's. The conjugate-gradient versions of the feedforward neural networks are computationally more efficient than conventional gradient descent neural networks. The original input data were Gray-coded since that representation has previously given the best results for this particular data set [2]. Using the Gray code with eight bits for each input variable expanded the dimensionality of input data to 56 dimensions. Therefore, each SNN had 57 inputs (one extra input for computing bias for the neurons), and ten outputs. All the neural networks used the sigmoid activation function. Since the input data were binary, the Gray code of the Gray code was the data transformation selected for the PCNN. Each SNN was trained for 200 iterations.

TABLE I
Training and Test Samples for Information Classes in the Experiment on the Colorado Data Set

Class #	Information Class	Training Size	Test Size
1	Water	301	302
2	Colorado Blue Spruce	56	56
3	Mountane/Subalpine Meadow	43	44
4	Aspen	70	70
5	Ponderosa Pine 1	157	157
6	Ponderosa Pine/Douglas Fir	122	122
7	Engelmann Spruce	147	147
8	Douglas Fir/White Fir	38	38
9	Douglas Fir/Ponderosa Pine/Aspen	25	25
10	Douglas Fir/White Fir/Aspen	49	50
Total		1008	1011

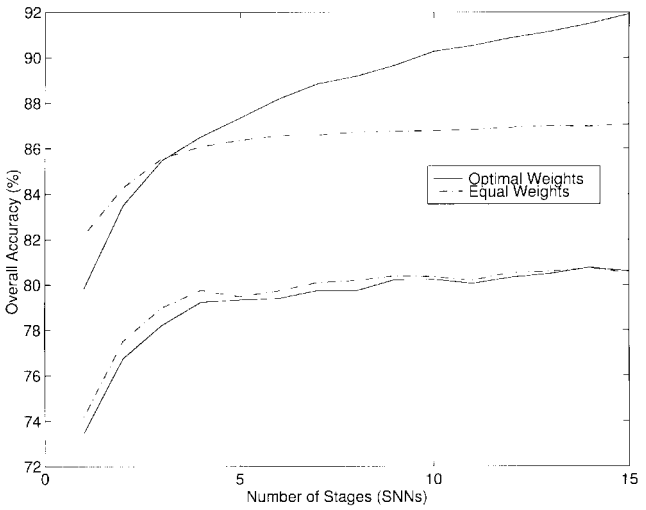

Fig. 4. Colorado data. Average results for the PCNN with equal and optimal weights as a function of the number of SNN's. The upper curves represent training results and the lower curves test results.

The PCNN was tested with randomly ordered stages in 11 different experiments. Up to 15 SNN's were used in each PCNN and the average overall classification accuracies were computed as a function of the number of SNN's in the PCNN. The average results of the experiments with the PCNN are shown in Fig. 4 for the two weight selection schemes and the standard deviation of the training accuracy for the PCNN is shown in Fig. 5.

The results using the PCNN were compared to the results obtained with three statistical methods in [3]: The minimum Euclidean distance (MED) classifier [31], the linear opinion pool (LOP), and the statistical multisource classifier (SMC) which is a version of the logarithmic opinion pool. Our original intent was also to use the Gaussian maximum likelihood (ML) method [31]. However, the ML method could not be applied since the whole data set cannot be modeled by Gaussian

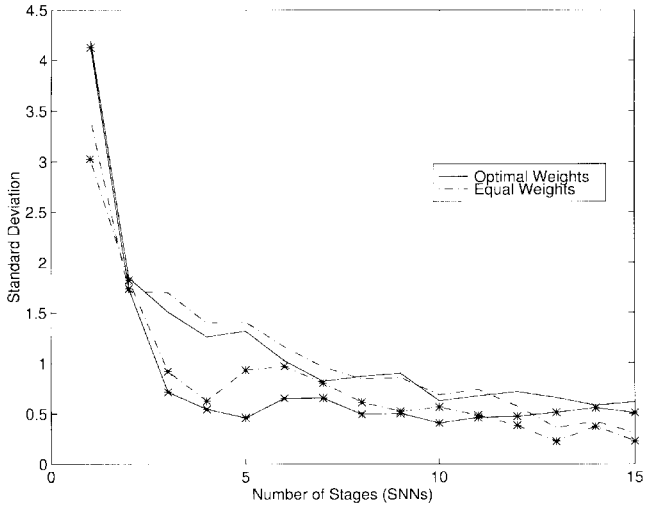

Fig. 5. Colorado data. Standard deviation for the training and test results of the PCNN mehtods. The * curves indicate training standard deviaitons.

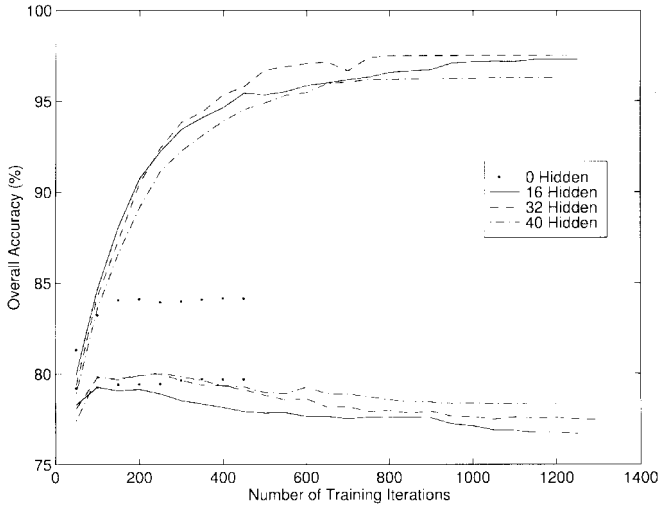

Fig. 6. Colorado data. Average results for the CGBP with a variable number of hidden neurons. The upper curves represent training results and the lower curves test results.

multivariate density functions. The reason for this is that several of the class-specific covariance matrices were singular due to low variation in the topographic data sources.

Also, the single-stage conjugate-gradient backpropagation (CGBP) algorithm with two layers [30] was trained on the same data with a variable number of hidden neurons. The CGBP neural networks had 57 inputs, zero, 16, 32, and 40 hidden neurons, and ten output neurons (the network with zero hidden neurons is the network which was used for the stages in the PCNN). Each version of the CGBP was trained six times with different initializations. The average results achieved with the CGBP (for different number of hidden neurons) are shown in Fig. 6 as a function of the number of training iterations.

The classification results are summarized in Table II. There it can be seen that the PCNN methods outperformed the single stage CGBP and the statistical methods in terms of overall classification accuracy of test data. Also, the difference between the equal weight selection and the optimal

TABLE II
OVERALL TRAINING AND TEST ACCURACIES FOR THE CLASSIFICATION METHODS APPLIED TO THE COLORADO DATA SET

Method	Training Accuracy	Test Accuracy
MED	40.28%	37.98%
LOP	76.19%	73.79%
SMC	81.25%	80.02%
CGBP (0 hidden neurons)	84.15%	79.70%
CGBP (40 hidden neurons)	91.16%	80.06%
PCNN (equal weights)	87.06%	80.74%
PCNN (optimal weights)	91.93%	80.77%

weighting method became very clear in the experiments. The optimal approach clearly outperformed the equal weighting approach in terms of training accuracy. In fact, for training data, the optimal weighting approach did show monotonically increasing overall accuracy as a function of the number of stages (see Fig. 4). This result was expected since the weights in the PCNN were optimized based on the training data. On the other hand, the PCNN methods showed very similar test accuracies after 15 stages. On the average, the optimal approach achieved 80.77% overall accuracy for test data as compared to 80.74% for the equal weighting approach. In comparison, the CGBP method achieved the maximum accuracy of 80.06% for test data (both for 32 and 40 hidden neurons at 250 iterations but the test accuracy was lower when the CGBP converged), whereas the SMC result was 80.02% and the LOP result 73.79% for the same data. Although the test accuracy difference between the PCNN methods, on one hand, and the SMC and the CGBP, on the other, seems small, this difference is statistically significant. Therefore, in the experiment both versions of the PCNN outperformed not only the CGBP but also the best statistical consensus theory method (SMC) in terms of classification accuracy of test data. Also, as expected, the standard deviation of the classification accuracy of the PCNN went down as the number of stages increased (Fig. 5).

B. Experiment 2: Anderson River Data Set

The data used in the second experiment, the Anderson River data set, are a multisource remote sensing and geographic data set made available by the Canada Centre for Remote Sensing (CCRS), Ottawa, Ontario

[32]. The imagery involves a 2.8 km by 2.8 km forestry site in the Anderson River area of British Columbia, Canada. The area is characterized by rugged topography, with terrain elevations ranging from 330 m to 1100 m above sea level. The forest cover is primarily coniferous, with Douglas fir predominating up to approximately 1050 m elevation, and cedar, hemlock, and spruce types predominating at higher elevations [32]. Six data sources were used:

1) airborne multispectral scanner (AMSS) with 11 spectral data channels (ten channels from 380–1100 nm and one channel from 8–14 μm);

TABLE III
TRAINING AND TEST SAMPLES FOR INFORMATION CLASSES IN THE EXPERIMENT ON THE ANDERSON RIVER DATA

Class #	Information Class	Training Size	Test Size
1	Douglas Fir (31-40m)	971	1250
2	Douglas Fir (21-30m)	551	817
3	Douglas Fir · Other Species (31-40m)	548	701
4	Douglas Fir · Lodgepole Pine (21-30m)	542	705
5	Hemlock · Cedar (31-40m)	317	405
6	Forest Clearings	1260	1625
Total		4189	5503

TABLE IV
AVERAGE PAIRWISE JM-DISTANCES FOR THREE OF THE DATA SOURCES (MAXIMUM JM-DISTANCE IS 1.414)

Data Source	Average JM-Distance
AMSS	1.19877
SAR Shallow	0.46305
SAR Steep	0.43109

2) steep mode synthetic aperture radar (SAR) with four data channels (X-HH, X-HV, L-HH, L-HV);
3) shallow-mode SAR with four data channels (X-HH, X-HV, L-HH, L-HV);
4) elevation data (one data channel, where elevation in meters $= 61.996 + 7.2266$ * pixel value);
5) slope data (one data channel, where slope in degrees $=$ pixel value);
6) aspect data (one data channel, where aspect in degrees $= 2$ * pixel value).

The AMSS and SAR data were detected during the week of July 25–31, 1978. Each channel comprises an image of 256 lines and 256 columns. All of the images are spatially coregistered with pixel resolution of 12.5 m.

There are 19 information classes in the ground reference map provided by CCRS. In the experiments, only the six largest ones were used, as listed in Table III. Here, training samples were selected uniformly, giving 10% of the total sample size. Test samples were then selected randomly from the rest of the labeled data.

To estimate the separabilities between the information classes for the AMSS and SAR data sources, Jeffries–Matusita (JM) distances [31] were computed. The average pairwise JM-distance separabilities are shown in Table IV for the AMSS and SAR data sources. The values in Table IV indicate that the Anderson River data is very difficult to classify. The AMSS source is apparently the most separable of the multidimensional data sources. Although it only has an average separability of 1.199, it is much more separable than the SAR data sources which are not very separable at all. Since these three multidimensional data sources are not very separable for this forest area, the topographic data may be expected to help in classifying the data more accurately than can be achieved using the remote sensing data alone.

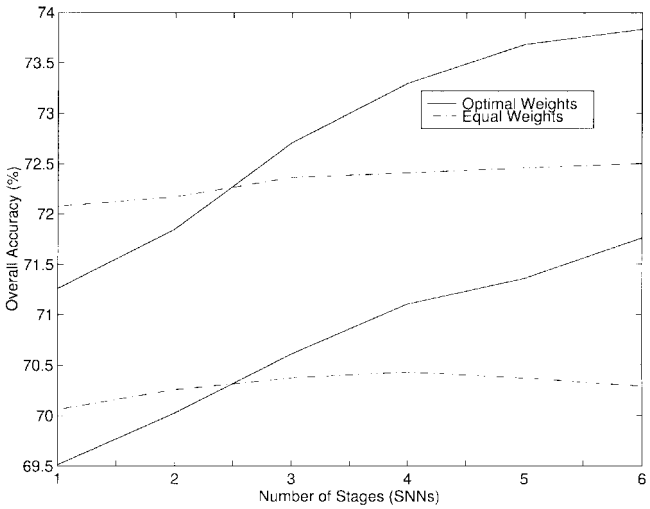

Fig. 7. Anderson River data. Average results for the PCNN with equal and optimal weights as a function of the number of ANN's. The upper curves represent training accuracies and the lower curves test accuracies.

Analog representation was used for the data and, therefore, the WPT was applied to obtain input vectors for the different stages. Zero-filling was used in the WPT to achieve vectors of length 32 (nearest power of two). Here, the PCNN with the WPT was implemented using two-layer conjugate-gradient backpropagation neural networks (CGBP) [2], [30] for the SNN's. All the neural networks used the sigmoid activation function. Each SNN had 33 inputs (one extra input for computing bias in the neurons), 15 hidden neurons and six output neurons. Both versions of the PCNN were used in the experiments, i.e., the equal weighting method and the optimal weighting method. In the experiments, the optimal weighting approach was again the pseudoinverse method. Each SNN was trained for 600 iterations.

The PCNN was tested with randomly ordered stages in fifteen different experiments. Up to six SNN's were in each PCNN (corresponding to the number of levels in the full WPT) and the average overall classification accuracies were computed as a function of the number of SNN's in the PCNN. The average classification results for the experiments with the PCNN's are shown in Fig. 7, and the standard deviations of the training and test accuracies for the PCNN's are shown in Fig. 8.

The results of the PCNN, were compared to results for four statistical methods used to classify the 22 band data [3], [31]: the minimum Euclidean distance (MED) classifier, the Gaussian maximum likelihood method (ML), the linear opinion pool (LOP), and the SMC. For the LOP and SMC 6 data classes were defined in each data source and the remote sensing data sources were modeled to be Gaussian but the topographic data sources were modeled by the maximum penalized likelihood method [33].

The single-stage CGBP algorithm with two layers [2], [30] was also trained on the same data with 15, 20, and 25 hidden neurons. Each version of the CGBP network was trained six times with different initializations. Then the overall average accuracies were computed for each version. The average

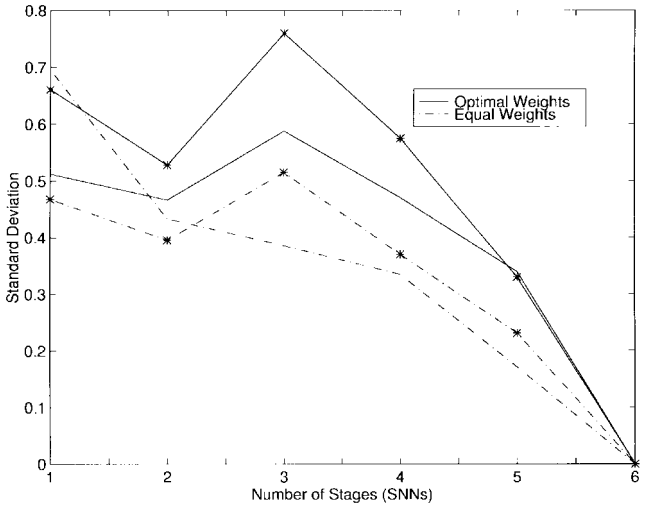

Fig. 8. Anderson River data. Standard deviation for the training and test results for the PCNN methods. The * curves indicate training standard deviations.

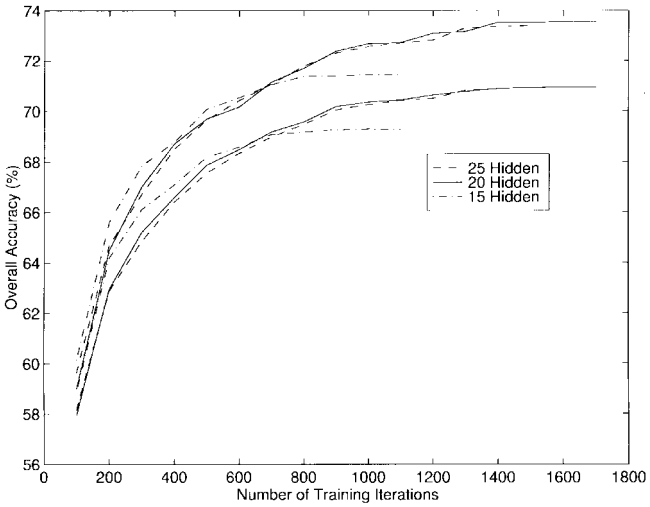

Fig. 9. Anderson River data. Average experimental results for the CGBP with a variable number of hidden neurons. The upper curves represent training results and the lower curves test results.

results for the CGBP networks are shown in Fig. 9 as a function of the number of training iterations.

The overall classification accuracies for the different methods are summarized in Table V. There it can be seen that the PCNN methods outperformed both the single stage CGBP's, and all the statistical methods in terms of overall classification accuracy of test data. Also, the optimal approach outperformed the equal weighting approach in terms of both training and test accuracy. The optimal method achieved, on the average, 73.84% overall accuracy for training data. In contrast, the corresponding accuracy for the equal weighting method was 72.50%. This difference between the methods was expected, since the weights in the optimal PCNN were optimized based on the training data. On the average, the optimal approach achieved 71.76% overall accuracy for test data as compared to 70.29% for the equal weighting approach. In comparison, the maximum overall accuracies for the CGBP method were 73.56% for training data and 70.94% for test data, and the maximum overall accuracies with the statistical methods were achieved by the SMC which gave 70.47% overall accuracy for training data and 68.20% overall accuracy for test data. The differences in test classification accuracies for the optimal PCNN and the CGBP can be shown to be statistically significant. Therefore, in the experiments the optimal weighting PCNN outperformed all other methods in terms of classification accuracies of test data. Also, as Fig. 8 displays, the standard deviations of the clasification for the PCNN's went down as the number of stages used increased, as expected.

TABLE V
OVERALL TRAINING AND TEST ACCURACIES FOR THE CLASSIFICATION METHODS APPLIED TO THE ANDERSON RIVER DATA SET

Method	Training Accuracy	Test Accuracy
MED	50.51%	50.83%
ML	68.23%	64.30%
LOP	54.00%	53.89%
SMC	70.47%	68.20%
CGBP (15 hidden neurons)	71.44%	69.28%
CGBP (20 hidden neurons)	73.56%	70.94%
PCNN (equal weights)	72.50%	70.29%
PCNN (optimal weights)	73.84%	71.76%

VII. CONCLUSIONS

In this paper, a new type of neural network-architecture, the PCNN, was proposed. The PCNN architecture is based on statistical consensus theory and its significance lies in using a collection of SNN's trained with different representations of input data in order to form a consensual decision. The PCNN takes advantage of the fact that a neural-network group decision is more accurate in the mean square sense than the decision of a single neural network. Also, classification performance of neural networks is very dependent on representation of input data. The PCNN provides a way of making a consensual decision for networks trained on different input representations and give the most weights in classification to the SNN's trained on the "best" representation of input data.

In the PCNN, the input data are transformed several times and the different transformed data are used as if they were independent inputs. The independent inputs are first classified using SNN's. The output responses from the SNN's are then weighted and combined to make a consensual decision.

Two methods were used to weight the outputs from the stage networks in the PCNN architecture. The simpler approach used equal weights for all the stages; the other used optimized weights, an approach which can also be used for other similar neural-network architectures.

An approach based on wavelet packets was also proposed for the selections of data transformations for PCNN's with analog inputs. Wavelet packets provide a systematic way of computing input data for the PCNN. Wavelet packets give different time-frequency resolutions of the original input data for the different stages. A more heuristic method based on

Gray coding was also suggested for PCNN's with binary inputs.

The results obtained showed that the PCNN performed very well in the experiments in terms of overall classification accuracy. In fact, the PCNN with the optimal weights outperformed both conjugate-gradient backpropagation and the best statistical methods in classification of multisource remote sensing and geographic data in terms of overall classification accuracy of test data. On the other hand, the PCNN uses multiple neural networks and improves the overall classification accuracy by using both more parameters and longer training time than single neural networks. However, when the data sets are difficult to model and accuracy is the most important factor, the PCNN with optimal weights should be considered a desirable alternative to other methods.

Some of the future research issues concerning the PCNN involve the weight selection. In this paper the weights for the stage neural networks were only based on the training set. Using the same data for training the classifiers and estimation of the weights can lead to overtraining by the optimal PCNN. This type of overtraining was seen in experiment 1 (see Fig. 4) where the optimal PCNN clearly outperformed the equally weighted PCNN in terms of training accuracies but the test accuracies for both methods were very similar. This behavior leads to the conclusion that it may be appropriate to use a different training set to train the classifiers than the one used to compute the weights for the stage neural networks. When the weights are computed it is desirable to know which network is the best one in general and not the best on the training set. A possible strategy is to take the training set, divide it into two, and use one half to train the classifiers and the other half to compute the weights for the stages.[1] If the training set is not large enough, one can use the leave-one out method or k-fold cross-validation [9].

ACKNOWLEDGMENT

This work was done in part while the first author was a visiting scholar at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN. The Colorado data set was originally acquired, preprocessed, and loaned to the authors by Dr. R. Hoffer of Colorado State University, Fort Collins. Access to the data set is gratefully acknowledged. The Anderson River SAR/MSS data set was acquired, preprocessed, and loaned by the Canada Centre for Remote Sensing of the Department of Energy, Mines, and Resources of the Government of Canada. The authors also thank the anonymous reviewers for their constructive comments which helped improve this paper.

REFERENCES

[1] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Neural-network approaches versus statistical methods in classification of multisource remote sensing data," *IEEE Trans. Geosci. Remote Sensing*, vol. 28, pp. 540–552, July 1990.

[2] ——, "Conjugate-gradient neural networks in classification of multisource and very-high dimensional remote sensing data," *Int. J. Remote Sensing*, vol. 14, no. 15, pp. 2883–2903, Oct. 1993.

[3] J. A. Benediktsson and Philip H. Swain, "Consensus theoretic classification methods," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, pp. 688–704, July/Aug. 1992.

[4] C. Berenstein, L. N. Kanal, and D. Lavine, "Consensus rules," in *Uncertainty in Artificial Intelligence*, L. N. Kanal and J. F. Lemmer, Eds. New York: North Holland, 1986.

[5] R. F. Bordley, "Studies in mathematical group decision theory," Ph.D. dissertation, Univ. California-Berkeley, 1979.

[6] C. Genest and J. V. Zidek, "Combining probability distributions: A critique and annotated bibliography," *Statist. Sci.*, vol. 1, no. 1, pp. 114–118, 1986.

[7] R. L. Winkler, "Combining probability distributions from dependent information sources," *Management Sci.*, vol. 27, pp. 479–488, 1981.

[8] R. A. Jacobs, "Methods for combining experts' probability assessments," *Neural Computa.*, vol. 3, pp. 867–888, 1995.

[9] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, pp. 241–259, 1992.

[10] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxley, and B. W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discrimination function," *IEEE Trans. Neural Networks*, vol. 1, pp. 296–298, 1990.

[11] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 993–1001, 1990.

[12] G. Mani, "Lowering variance of decisions by using artificial neural-network portfolios," *Neural Computa.*, vol. 3, pp. 484–486, 1991.

[13] M. P. Perrone and L. N. Cooper, "When networks disagree: Ensemble Methods for hybrid neural networks," in *Neural Networks for Speech and Image Processing*, R. J. Mammone, Ed. London: Champman-Hall, 1993.

[14] M. P. Perrone, "Improving regression estimation: Averaging methods for variance reduction with extensions to general convex measure optimization," Ph.D. dissertation, Dep. Physics, Brown Univ., Providence, RI, 1993.

[15] K. Tumer and J. Ghosh, "Theoretical foundations of linear and order statistics combiners for neural pattern classifiers," *IEEE Trans. Neural Networks*, to appear.

[16] N. Nilsson, *Learning Machines*. New York: McGraw-Hill, 1965.

[17] E. Alpaydin, "Multiple networks for function learning," in *Proc. 1993 IEEE Int. Conf. Neural Networks*, vol. 1, San Francisco, CA, pp. 9–14, 1993.

[18] R. Battiti and A. M. Colla, "Democracy in neural nets: Voting schemes for classification," *Neural Networks*, vol. 7, pp. 691–707, 1994.

[19] G. Rogova, "Combining the results of several neural-network classifiers," *Neural Networks*, vol. 7, no. 5, pp. 777–781, 1994.

[20] S.-B. Cho and J. H. Kim, "Multiple network fusion using fuzzy logic," *IEEE Trans. Neural Networks*, vol. 6, pp. 497–501, 1995.

[21] O. K. Ersoy and D. Hong, "Parallel, self-organizing, hierarchical neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 167–178, 1990.

[22] S.-W. Deng and O. K. Ersoy, "Parallel, self-organizing, hierarchical neural networks with forward–backward training," *Circuits, Syst., Signal Processing*, vol. 12, no. 2, pp. 223–246, 1993.

[23] O. K. Ersoy and S.-W. Deng, "Parallel, self-organizing, hierarchical neural networks with continuous inputs and outputs," *IEEE Trans. Neural Networks*, vol. 6, pp. 1037–1044, 1995.

[24] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computa.*, vol. 3, pp. 79–87, 1991.

[25] H. Valafar and O. K. Ersoy, "Parallel, self-organizing, consensual neural network," School Electr. Eng., Purdue Univ., West Lafayette, IN, Rep. TR-EE 90-56, 1990.

[26] D. Graupe, *Time Series Analysis, Identification, and Adaptive Filtering*. Melbourne, FL: Robert E. Krieger, 1984.

[27] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[28] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. 38, pp. 713–718, 1992.

[29] I. Daubechies, *Ten Lectures on Wavelets*, CBMS-NFS Regional Conf. Series Appl. Math.s, vol. 61. Philadelphia, PA: SIAM, 1992.

[30] E. Barnard, "Optimization for training neural nets," *IEEE Trans. Neural Networks*, vol. 3, pp. 232–240, Mar. 1992.

[31] P. H. Swain, "Fundamentals of pattern recognition in remote sensing," *Remote Sensing—The Quantitative Approach*, P. H. Swain and S. Davis, Eds. New York: McGraw-Hill, 1978.

[32] D. G. Goodenough, M. Goldberg, G. Plunkett, and J. Zelek, "The CCRS SAR/MSS Anderson river data set," *IEEE Trans. Geosci. Remote Sensing*, vol. GE-25, pp. 360–367, May 1987.

[33] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Monographs on Statist. Appl. Probability. New York: Chapman and Hall, 1986.

[1] Suggested by an anonymous reviewer.

Jon Atli Benediktsson (S'84–M'90) received the bachelor's degree in electrical engineering from the University of Iceland, Reykjavik, in 1984 and the M.S.E.E. and the Ph.D. degrees from the School of Electrical Engineering at Purdue University, West Lafayette, IN, in 1987 and 1990, respectively.

He was with the University of Iceland's Laboratory for Information Technology and Signal Processing from 1984 to 1985. From 1985 to 1991 he was affiliated with the School of Electrical Engineering and the Laboratory for Applications of Remote Sensing (LARS) at Purdue University. He is currently Associate Professor of Electrical and Computer Engineering at the University of Iceland. His research interests are pattern recognition, neural networks, remote sensing, image processing, and signal processing.

Dr. Benediktsson received the 1991 Stevan J. Kristof Award from LARS as outstanding graduate student in remote sensing. Dr. Benediktsson is a Member of the International Neural Network Society (INNS), the Institute for Nonlinear Analysts (IFNA), and Tau Beta Pi.

Johannes R. Sveinsson (S'86–M'90) received the B.S. degree from the University of Iceland, Reykjavik, and the M.Sc.(Eng.) and the Ph.D. degrees from Queen's University, Kingston, Ontario, Canada, all in electrical engineering.

He was a Visiting Research Student at the Department of Electrical Engineering, Imperial College, London, England from 1985 to 1986. From 1981 to 1982, he was with the Laboratory for Information Technology and Signal Processing, University of Iceland. At Queen's he held teaching and research assistantships. Since November 1991 he has been with the Engineering Research Institute, University of Iceland as a Member of Research Staff and a Lecturer in the Department of Electrical and Computer Engineering. His current research interests include systems theory.

Dr. Sveinsson is a Member of SIAM and received Queen's University Graduate Awards.

Okan K. Ersoy (M'86–SM'90) received the B.S.E.E. degree from Robert College, Istanbul, Turkey, in 1967, and the M.S., Certificate of Engineering, a second M.S., and Ph.D. degrees from the University of California, Los Angeles (UCLA), in 1968, 1971, and 1972, respectively.

He was a Teaching and Research Assistant in the Department of Electrical Sciences and Engineering, UCLA, from 1968 to 1972, Assistant Professor in the Department of Electrical Engineering, Bosphorus University, Istanbul, Turkey, from 1972 to 1973, and Associate Professor in the second semesters at the same university from 1976 to 1980. He joined the Center for Industrial Research, Oslo, Norway, as a Researcher in the Computer Science Division in 1973. He was a Visiting Scientist at the University of California, San Diego, in 1980 to 1981. He has been with Purdue University, School of Electrical Engineering, West Lafayette, IN, since August 1985. His current interests include neural networks, digital signal/image processing and recognition, spectral methods, related parallel and fast algorithms, diffractive optics, and optical pattern recognition. He has published approximately 150 papers. He also holds three patents which are separately patented in the USA, Norway, Denmark, and Sweden.

Dr. Ersoy is a Member of the Optical Society of America and the Society for Photo-Optical Instrumentation Engineers. He was a Fulbright Fellow in 1967 to 1968. He is also an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, and the *International Journal for Smart Engineering Design*.

Philip H. Swain (S'66–M'69–SM'81) received the B.S. degree in electrical engineering from Lehigh University, Bethlehem, PA, in 1963 and the M.S. and Ph.D. degrees from Purdue University, West Lafayette, IN, in 1964 and 1970, respectively.

During the 1984–1985 academic year, he was Honorary Visiting Fellow at the University of New South Wales, Sydney, Australia. He has been affiliated with the Laboratory for Applications of Remote Sensing (LARS) at Purdue since its inception in 1966. Much of that time he served LARS as Program Leader for Data Processing and Analysis Research, responsible for the development of methods and systems for the management and analysis of remote sensing data. He has been employed by the Philco-Ford Corporation and the Burroughs Corporation and served as a consultant to the National Aeronautics and Space Administration (NASA), the Universities Space Research Association and IBM. He is currently Professor of Electrical and Computer Engineering and Director of Continuing Engineering Education at Purdue. He is Coeditor and Contributing Author of the textbook *Remote Sensing: The Quantitative Approach* (New York: McGraw-Hill, 1978). His research interests include theoretical and applied pattern recognition, methods of artificial intelligence, geographic information systems, and the application of advanced computer architectures to image processing.

Dr. Swain is a Member of the American Society for Engineering Education (ASEE), the Universities Continuing Education Association, Phi Beta Kappa, Sigma Xi, and Eta Kappa Nu. As Vice Chairman of the Technical Committee on Remote Sensing (TC7) of the International Association for Pattern Recognition, he helped organize a Workshop on Analytical Methods in Remote Sensing for Geographic Information Systems, held in Paris, France, in 1986 and published its proceedings as a special issue of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. In 1994, he received the Meritorious Achievement Award in Continuing Education from the IEEE Educational Activities Board.