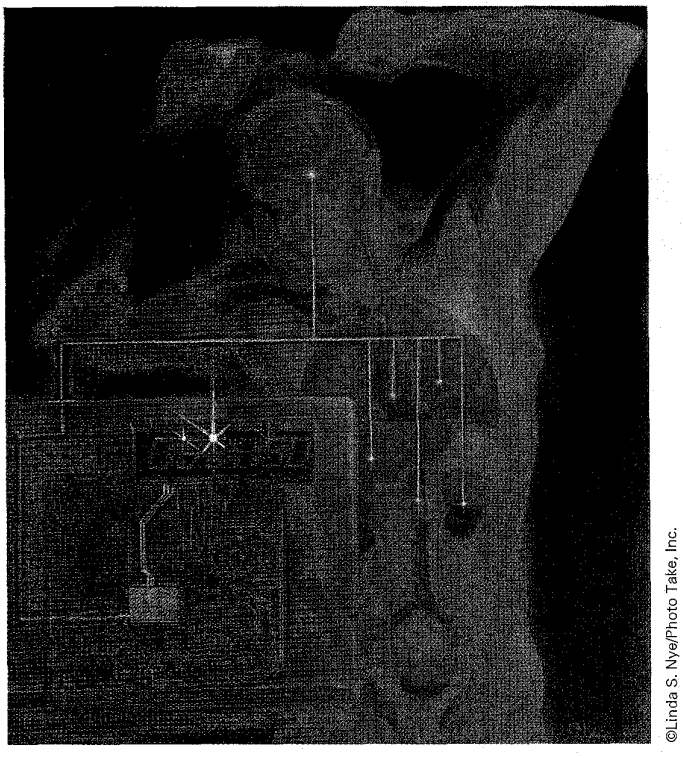
Neural Networks Expand SP's Horizons

Advanced algorithms for signal processing simultaneously account for nonlinearity, nonstationarity, and non-Gaussianity

SIMON HAYKIN



Statistical signal processing covers an area where physics and mathematics meet and interact to solve a wide range of problems. Its origins may be traced back to the 1943 classified RCA report by North, republished in [1], the 1946 classic paper [2] by Van Vleck and Middleton, and the pioneering work by Wiener [3]. In particular, the classical methods of statistical signal processing are founded on three basic assumptions: linearity, stationarity, and second-order statistics with particular emphasis on Gaussianity. These assumptions are invoked for the sake of mathematical tractability. Yet most, if not all, the physical signals that we have to deal with in real-life applications are generated by dynamic processes that are simultaneously nonlinear, nonstationary, and non-Gaussian. The end result of designing a signal-processing system along traditional lines is a suboptimal solution. One way in which the performance of the system can be improved is to consider the use of neural networks in combination with other suitable techniques (e.g., time-frequency analysis), depending on the task at hand.

Interest in neural networks, or to be more precise, artificial neural networks, has always been motivated by the fact that the human brain functions in a manner entirely different from the conventional digital computer. The human brain is a gigantic, and yet highly efficient, information-processing machine that encompasses a wide variety of complex signal processing operations. To appreciate the enormous scale of these operations, we need only look at our visual and auditory systems and be amazed at the "seamless" nature of the way in which different forms of information gathered by our eyes and ears are individually processed and then finally fused together.

Work on neural networks may be traced back to the pioneering paper [4] by McCulloch and Pitts in 1943, which was followed by Rosenblatt's development of the perceptron [5] and Widrow's development of the adaline [6] in the late 1950s. After going through a period of dormancy (in an engineering context) in the 1970s, neural networks re-emerged in the 1980s with the publication of Hopfield's

1053-5888/96/$5.00©1996IEEE

paper on recurrent networks [7] and the two-volume seminal book [8] by Rumelhart and McClelland on parallel distributed processing (PDP). We may look back on the 1980s not only as the decade of re-emergence of neural networks but also as one of consolidation.

Insofar as this article is concerned, the primary interest is in the use of neural networks as an engineering tool for signal processing applications. The aim of the article is three fold:

- Articulate a new philosophy in the approach to statistical signal processing using neural networks, which (either by themselves or in combination with other suitable techniques) account for the practical realities of nonlinearity, nonstationarity, and non-Gaussianity
- Describe three case studies using real-life data, which clearly demonstrate the superiority of this new approach over the classical approaches to statistical signal processing
- Discuss mutual information as a criterion for designing unsupervised neural networks, thus moving away from the mean-square error criterion

Rationale for Using Neural Networks

Neural networks have a number of important properties that befit their use for signal processing applications. In particular, we mention the following five properties:

Neural networks are distributed nonlinear devices
This property is a direct result of the fact that each processing unit (i.e., neuron) of a neural network has a built-in activation function (for example, in the form of a logistic function) that is nonlinear. Accordingly, neural networks have the inherent ability to model underlying nonlinearities contained in the physical mechanism responsible for generating the input data.

A neural network consists of a massively parallel processor that has the potential to be fault tolerant
For example, a multilayer perceptron, representing a popular structure for the implementation of a neural network, consists of a large number of neurons arranged in the form of layers, with each neuron in a particular layer connected to a large number of source nodes/neurons in the previous layer. This form of global interconnectivity has the potential to be fault tolerant, in the sense that the performance is degraded gracefully under adverse operating conditions. If a neuron or its synaptic links are damaged, the recall quality of a stored pattern is impaired, but owing to the highly distributed nature of the network, the damage has to be extensive before the performance is seriously degraded. Nevertheless, to be assured that the neural network is in fact fault tolerant, we may find it necessary to take proper measures in designing the algorithm used to do the training [8].

Neural networks have a natural ability to adapt their free parameters to statistical changes in the environment in which they operate.
As a rule of thumb, we may say that the more we make a nonlinear system adaptive, the more robust the performance of that system is likely to be when it operates in a nonstationary environment, subject, of course, to the requirement that the system remains stable. (We ourselves are a living example of this rule.) However, for the full benefits of adaptivity to be realized, there has to be a successful resolution to the stability-plasticity dilemma. This means that the principal time constants of the system should be long enough to ignore spurious disturbances, and yet short enough to respond to meaningful changes in the environment. Ordinary adaptive filters also have the ability to adjust their parameters automatically in accordance with statistical variations of their environment [10,11]; however, their adaptive signal processing capability is limited by their structural formulation as simple linear combiners.

Neural networks provide a nonparametric approach for the nonlinear estimation of data
The nonlinear, feedforward multilayer class of neural networks (encompassing multilayer perceptrons and radial basis-function networks) learns about its environment in a supervised manner. (The design of multilayer perceptrons and radial-basis function networks is discussed in the book by Haykin [12], and the review papers by Lippmann [13] and Hush and Horne [14].) Specifically, these neural networks undergo a training session during which their free parameters (i.e., synaptic weights and biases) are adjusted in a systematic way so as to minimize a cost function. Typically, the cost function is defined on the basis of a mean square-error criterion, with the error signal itself being defined as the difference between a desired response and the actual output of the network produced in response to a corresponding input signal. The neural network learns from examples by constructing an input-output mapping for the problem at hand, which brings to mind the notion of nonparametric statistical inference; see Table 1. The term "nonparametric" is used here in a statistical sense, meaning that no knowledge of the underlying probability distribution is required.

In the traditional approach to mathematical statistics as taught in a statistics department, the issues of primary concern are two-fold:

- The use of mathematically tractable models, assuming the idealized conditions of linearity, wide-sense stadonarity, and Gaussianity, for the derivation of parameter estimators.
- Derivation of exact properties (e.g., mean and variance) of estimators for small sample-sizes; if the exact properties are not mathematically tractable, then one would consider the asymptotic properties of the estimators as the number of samples approaches infinity.

Table 1: Statistical Inference

Quoting from Rao [15]:

"Statistical inference is in the nature of inductive logic involving generalizations from the particular, and naturally any tool (statistical method) employed for this purpose will be subject to some controversy".

Recognizing that, in reality, neural networks represent statistical techniques that are subject to well-known statistical limitations (Barron and Barron [16]; White [17]; Ripley [18,19]; Murtagh [20]; Cheng and Titterington [21]; Cherkassky et al. [22]), this statement is equally applicable to a neural network as a tool for statistical signal processing.

Statistical inference problems may be classified into two categories:

1. *Regression*, in which the outputs are continuous random variables.
2. *Classification*, in which the outputs are discrete class labels (categorical).

Consider, for example, a nonlinear multiple regression problem involving a scalar output, which is described by the relation

$$y = f(\mathbf{x}) + e$$

where the multidimensional vector x is the input, the scalar y is the output, and e is the error with unknown statistics. In nonparametric regression, none or very few assumptions are made about the function $f(\mathbf{x})$. A multilayer perceptron, for example, uses a nested form of nonlinearity to approximate the function $f(\mathbf{x})$. It does so by undergoing a training session that involves a series of input-output examples, $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$ which is representative of the environment of interest.

Consider next the example of a binary classification (hypothesis-testing) problem based on an independent and identically distributed (iid) observation sequence, as described here:

Hypothesis H_0: $\mathbf{x}_n$ is defined by the probability distribution $P \in P_0$ for $n=1,2,\ldots,N$

Hypothesis H_1: $\mathbf{x}_n$ is defined by the probability distribution $P \in P_1$ for $n=1,2,\ldots,N$

The classification problem is said to be nonparametric if the distribution P_0 or P_1 is unknown, and a finite number of parameters will not suffice to specify them [23,24]. In this case, the multilayer perceptron undergoes a training session for the implicit purpose of providing estimates of the unknown distributions P_0 and P_1.

Neural networks provide another method for solving nonparametric statistical inference problems, particularly those that are characterized by nonlinearity and non-Gaussianity. For comparisons of neural networks with other nonparametric methods, see Ripley [28] and Cherkassky [25]. For a discussion of neural network model selection, see [18,26].

In contrast, neural network-based methods are attractive for practical applications by virtue of their ability to deal with nonlinearity, nonstationarity, and non-Gaussianity. Moreover, they offer robustness with respect to parameter tuning and sample properties, which is important for a good setting of user-tunable parameters by non-expert users. It is not quite clear why, in this respect, neural networks appear to behave better than comparable statistical techniques such as projection pursuit [27,28], splines [29], and multivariate adaptive regression splines (MARS) [30]. *Projection pursuit* is similar and mathematically equivalent to the multilayer perceptron in terms of representation. Splines are closely related to radial-basis function (RBF) networks. MARS may be viewed as a tree of neurons with each leaf of the tree consisting of a neuron; the neuron may itself be modeled as a piecewise linear polynomial or a cubic polynomial with the knot of the spline treated as a variable.

A possible explanation for the superiority of neural networks may be found in the differences in the way in which the respective optimization procedures are pursued [32]. In statistical methods, particularly those that use a greedy form of optimization with the basis functions tuned one at a time, it may be difficult or perhaps impossible to recover from any wrong decisions made in some early stages of the optimization process. In contrast, in a neural network the complete set of basis functions represented by the outputs of the hidden neurons are optimized simultaneously in an iterative fashion, hence the more robust behavior.

In addition to robustness, there are other considerations to be taken into account, such as prediction accuracy in the context of dense samples or sparse samples, where "denseness" is measured with respect to the target function complexity (i.e., smoothness). Insofar as prediction accuracy is concerned, it can be said that there is no single method that provides a superior performance under all possible situations [31]. Evidence supporting this claim, using computer simulations on various statistical methods including neural networks, is presented in [25].

Neural networks, operating in a supervised manner, are universal approximators

Multilayer feedforward networks (i.e., multilayer perceptrons and radial-basis function networks) are universal approximators, in the sense that they can approximate any continuous input-output mapping to any desired degree of approximation, given a sufficient number of hidden units [33-35]. This property is also shared by classical methods based on the use of smooth functions such as algebraic or trigonometric polynomials. What is really important, therefore, is the rate of convergence with which the unknown

function is approximated for a prescribed set of basis functions. In classical approximation theory involving bounded norms of the derivatives of order s for some $s > 0$, the rate of convergence is $O(n^{-2s/(2s+p)})$, where n is the degree of the polynomial, and p is the dimensionality of the input space. The dependence of the rate of convergence on p in the exponent is a manifestation of the curse of dimensionality. The implication here is that in a high-dimensional space (i.e., large p) one can only approximate very smooth functions (i.e., large s) for a given number of samples (i.e., prescribed n). For the corresponding case of neural networks, Barron [36] has shown that it is possible to approximate any function satisfying a certain condition on its Fourier transform by a multilayer perceptron, with the rate of convergence being $O(1/\sqrt{n})$, where n is the number of sigmoid basis functions (i.e., hidden neurons). Even though this result has been (mis)interpreted as if the use of neural networks overcomes the curse of dimensionality (i.e., the rate of convergence does not depend on the dimensionality p of the input space), careful examination of the result shows that with increasing dimensionality p, the smoothness of the function being approximated would have to be increased to ensure that the condition on the bounded norm of its Fourier transform is satisfied [37].

Principle of Empirical Risk Minimization

In a real-life situation, we have to work with a finite sample size, irrespective of the statistical estimation procedure used. With the notable exception of Vapnik's pioneering work that remains largely unknown to the signal processing community, there exists no widely accepted theory for small-size nonparametric estimation.

Vapnik's work hinges on an important parameter called the Vapnik-Chervonenkis dimension, or simply the VC-dimension [38]. In the context of pattern classification, the VC-dimension provides a measure of the capacity of the family of classification functions realized by a learning machine.

The VC-dimension plays a central role in the *principle of empirical risk minimization*, an inductive principle that does not require probability density estimation. This makes it perfectly suited to the underlying premise of neural networks. The basic idea of the method is to use a set of N identically and independently distributed (iid) training examples $(\mathbf{x}_1,\mathbf{d}_1),(\mathbf{x}_2,\mathbf{d}_2),\ldots,(\mathbf{x}_N,\mathbf{d}_N)$ to construct the empirical risk functional [39]:

$$R_{emp}(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} L(\mathbf{d}_i;\mathbf{F}(\mathbf{x}_i,\mathbf{w}))$$

which does not depend on the unknown probability that pertains to the generation of the training examples. In this equation, $\mathbf{x}_i$ and $\mathbf{d}_i$ denote the input vector and the desired response vector for the ith training example, respectively, and $\mathbf{w}$ is the set of free pararneters (weights) selected by the learning machine (i.e., neural network). The function $L(\mathbf{d}_i;\mathbf{F}(\mathbf{x}_i,\mathbf{w}))$ represents the loss or discrepancy between the desired response vector, $\mathbf{d}_i$, corresponding to an input vector, $\mathbf{x}_i$, and the actual response, $\mathbf{F}(\mathbf{x}_i,\mathbf{w})$, produced by the neural network.

Let $\mathbf{w}_{emp}$ denote the parameter vector that minimizes $R_{emp}(\mathbf{w})$ over the parameter space. According to the principle of empirical risk minimization, the functional $R_{emp}(\mathbf{w})$ converges in probability to the minimum possible value of the actual risk functional $R(\mathbf{w})$ as the size, N, of the training set is made infinitely large, provided that the empirical risk functional $R_{emp}(\mathbf{w})$ converges uniformly to the actual risk functional, $R(\mathbf{w})$. The theory of uniform convergence of $R_{emp}(\mathbf{w})$ to $R(\mathbf{w})$ includes bounds on the rate of convergence, which are, in turn, based on the VC-dimension. For a discussion of these issues, see [12, 39, 40].

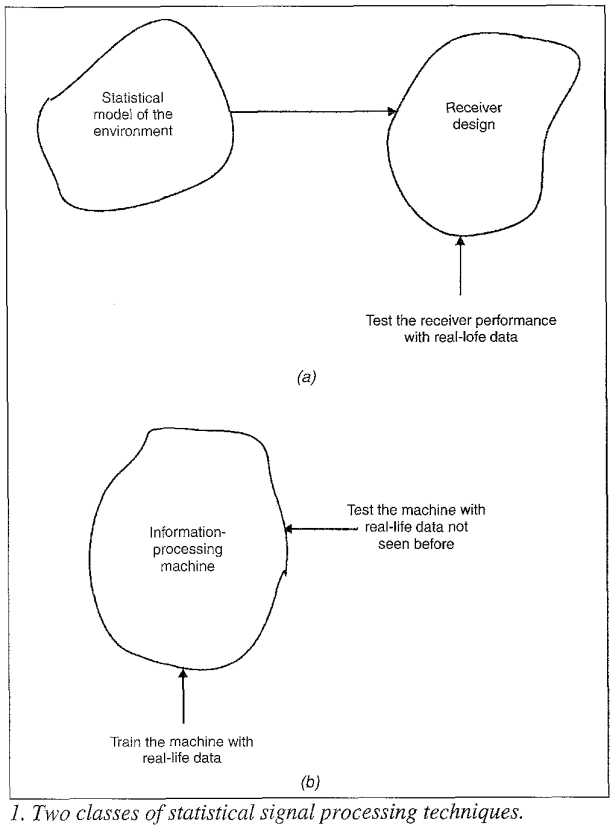
Information Preservation Rule

Neural networks may not be adequate to tackle all statistical signal processing applications by themselves. Rather, neural networks may have to be integrated with other related techniques in a principled way in order to capture the full information content of the input data and exploit the information in an efficient manner. A particular technique that lends itself to this approach is that of time-frequency (scale) analysis [41-42], by means of which a one-dimensional signal is transformed into a time-frequency (scale) image. (For example, as illustrative ways in which wavelets, a popular method for performing time-scale analysis, can be integrated with neural networks for different applications, see [43-51].) The useful feature of time-frequency transformation is that it displays the temporal localization of the signal's spectral components in a more discernible fashion than would be the case directly from the signal or its spectrum.

Whatever form of system integration is used, the design objective should be in accord with an information-theoretic rule of thumb that may be stated as follows:

In designing a receiver, the available information pertaining to a signal-processing task (e.g., target detection or parameter estimation) should be preserved optimally (in a statistical sense) and used efficiently (in a computational sense), until the receiver is ready for final decision-making. (This rule is based on one of three lessons learned from information theory; see Viterbi [52].)

In the sequel, we refer to this rule as the information preservation rule. To appreciate the practical significance of this rule, consider the case of remote sensing. In this kind of application, we typically find that the sensors (e.g., the antenna in a weather radar system and its electromagnetic accessories) represent a highly significant part of the capital investment involved in building the system. Most importantly, it is unlikely that the cost of the sensors would go down. In direct contrast, signal-processing subsystems are becoming progressively cheaper, thanks to very-large-scale integration (VLSI) technology. This is all the more reason for adhering to the information preservation rule, thereby putting



1. Two classes of statistical signal processing techniques.

the sensors to their most cost-effective use. Neural networks have the potential to preserve information by virtue of their ability to learn a model of their environment through exposure to input-output examples that are representative of the environment. For the information preservation rule to be satisfied, however, the structural complexity of the neural network should closely match the underlying complexity of the input data. This raises the issue of network complexity that has attracted the attention of many researchers, building on statistical criteria such as Rissanen's minimum-description length (MDL) criterion and cross validation; see [18,26] for a discussion of this important design issue.

Successful design of a neural network rests not only on the right selection of a network structure but also the availability of a reliable training set, (i.e., a training set that is precise and relatively noise-free). If we recognize that, irrespective of the design methodology, the statistical performance of a signal-processing system must be evaluated with real-life data prior to use in an operational setting, we may just as well start with the collection of a "labeled" dataset (i.e., ground truthed) that will be representative of the particular environment of interest. Part of the dataset is used to train the neural network, and the remaining part is subsequently used to test it. Unfortunately, the learning process is an ill-posed inverse problem for the following reasons [12]. First, the information content of the training data may not be sufficient to reconstruct the input-output mapping uniquely. Second, the unavoidable presence of noise or imprecision in the training data adds uncertainty to the reconstructed input-output mapping. To make the learning process well posed, some prior information (e.g., smoothness constraints) on the input-output mapping must be included in the formulation of the learning algorithm. This is achieved by adding a regularizing term (i.e., stabilizer) to the cost function used to derive the algorithm for training the neural network [12, 53, 54].

To sum up, we may identify two different approaches to statistical signal processing, as indicated in Fig. 1. In the parametric approach depicted in Fig. 1a, we start with a statistical model of the underlying physical mechanism responsible for generating the input data, and then use the model to design the receiver. The success of this approach depends on how closely the model describes the realities of the physical mechanism responsible for generating the data. In direct contrast, in the nonparametric approach depicted in Fig. 1b, the information-processing machine provides not only a statistical model of the environment in which it operates, but also the final receiver design. The success of this latter approach depends on how representative the training data are of the physical environment, and how adequate the size of the training data is. Neural networks, viewed in a statistical sense, belong to the approach described in Fig. 1b.

Criteria for Acceptance of Neural Networks

In assessing the engineering attributes of a "good" signal processor, we come upon two particular attributes:

- Optimal preservation of the available information, and therefore optimality of performance in some statistical sense
- Robustness of performance with respect to small variations in environmental conditions

Given these attributes, neural networks can gain acceptance as tools for solving statistical signal processing problems, in preference to traditional methods, if

(i) using a neural network makes a significant difference in the statistical performance of a system for a real-world application, or can provide a significant reduction in the cost of implementation without compromising performance

(ii) by virtue of its massively parallel and distributed structure, a neural network offers a more graceful degradation of performance due to the unavoidable failure of network components than would be possible with other nonparametric methods

(iii) the tuning of adjustable parameters in a neural network is a more straightforward task (and therefore easily accomplished by a nonexpert user) than would be the case with other nonparametric methods

(iv) through the use of a neural network, by itself or in combination with some other devices, we are able to solve difficult signal processing problems for which there are no viable solutions using standard methods

A practical limitation of neural networks is that when working with real-life data, training for an application may take a long time; the length of training would naturally have to be viewed in the context of available computing resources. The relatively long time needed to train a neural network is largely due the computer architecture (serial in nature) in current use, which is ill suited to programming neural networks. Special-purpose processors (e.g., the ANNA chip [55] and CNAPS [56]) are available today that can speed up the training process significantly for specific types of neural networks. In addition, the back-propagation algorithm (widely recognized as the workhorse for the design of neural networks) lends itself to parallelism. Indeed, many papers have been written on this issue; see [57] and the references listed herein. Through the use of parallelism, the training process of a multilayer perceptron required to tackle a large problem may be facilitated by using a large number of parallel processors and distributing the synaptic weights of the network over these processors.

Another weakness is that it is often difficult to see how knowledge gained by the neural network about its environment is actually represented inside the network. Some display/graphical tools such as the Hinton diagram and the bond diagram have been developed to remedy this difficulty [12, 58, 59].

Case Studies

Now consider three case studies (based on real-life data), with which I and some of my research colleagues have been involved for the past six years. These studies, in their own ways, testify to the computing power of neural networks in solving difficult signal processing problems.

Case Study I: Chaotic Modeling of Sea Clutter and its Cancellation

For nearly half a century, sea clutter (i.e., the radar backscatter from an ocean surface) has been modeled as a stochastic process, with a variety of probability distributions proposed for describing its stochasticity [60-63]. However, there is now strong experimental evidence that shows that sea clutter is indeed a chaotic process [64-66].

A *chaotic process* is generated by a deterministic mechanism of a relatively low dimension, and yet it generates a randomlike waveform that exhibits many of the characteristics that are normally associated with a stochastic process. Table 2 presents a summary of the important properties of a chaotic process.

The term "chaos" was coined by J.A. Yorke, an applied mathematician at the University of Maryland [87]. Yorke's paper with Li [88] probably introduced the term as a mathematical concept, though many others had previously talked about chaotic fluid behaviors. Chaos, representing a new paradigm, owes its origin to the pioneering work of Lorenz on simulated weather data [89]; some of Lorenz's personal recollections of his first computer model of the atmosphere appear in [90]. What came out of that computer simulation is now known as the Lorenz attractor. An attractor represents the equilibrium state of a nonlinear dynamical system, which may be observed experimentally after the transients have died. The Lorenz attractor is an example of a strange attractor. The strangeness comes from two important features: unlike a smooth curve or surface, a strange attractor is an object with a fractal (i.e., non-integer) dimension; unlike an ordinary attractor, the motion of a strange attractor exhibits sensitive dependence on initial conditions.

The term "strange attractor" was coined in a paper by Ruelle and Takens [91], in which they claimed that turbulent flow is not described by a superposition of many modes (as previously proposed) but by strange attractors. The existence of chaos in fluid turbulence is confirmed in [92].

Using an extensive and ground-truthed database collected by means of an instrument-quality X-band radar (called the IPIX radar) pointing along a fixed direction and dwelling onto a patch of the ocean surface, researchers have demonstrated the chaotic nature of sea clutter in light of what is known about chaos theory. The clutter-to-noise ratio of the data collected with this radar was on the order of 30 dB, and the wordlength of the A/D converter was 8 bits (equivalent to a dynamic range of 48 dB). Important aspects of the research findings reported in [65, 66] may be summarized as follows:

1. The largest Liapunov exponent, λ_1, is always positive. For the particular radar used to do the data collection, λ_1 is estimated to be about 0.03, which is normalized with respect to the pulse-repetition period of the radar. This value is essentially independent of the following (for a given radar system):

- radar parameter (i.e., amplitude, in-phase component, or quadrature component
- sea state
- radar location

Moreover, the second Liapunov exponent, λ_2, is very close to zero, and for a prescribed embedding dimension, the sum of all the Liapunov exponents is negative. The implications of these latter observations are twofold:

- Sea clutter is generated by a *coupled system of nonlinear differential equations*
- The dynamic mechanism responsible for the generation of sea clutter is a *dissipative* one

2. The correlation dimension, D_C, is fractal (i.e., non-integer), lying in the range of 6 to 9. Moreover, it is also essentially

Table 2: Characteristics of a Chaotic Process

Chaos is the very complex behavior of a dynamical system that is both nonlinear and deterministic [67-69]. It represents a powerful notion, permitting the use of a deterministic system to explain highly irregular fluctuations exhibited by many physical phenomena encountered in nature. For the definition of a chaotic system, we offer the following that builds on Newhouse's definition [70]:

"A bounded deterministic dynamical system with at least one positive Liapunov exponent is a chaotic system; a chaotic signal is an observation of a chaotic system."

The ith Liapunov exponent of a nonlinear dynamical system is defined by

$$\lambda_i = \lim_{t\to\infty} \frac{1}{t} \frac{l_i(t)}{r(0)}$$

where t denotes time, $r(0)$ is the infinitesimal radius of the initial "fiducial" volume, and $l_i(t)$ is length of the ith principal axis at time t. By convention, the Liapunov exponents are always ordered as $\lambda_1 > \lambda_2 > \lambda_3 \ldots$ The presence of a positive Liapunov exponent, a basic requirement for chaotic behavior, causes trajectories that are initially close to each other to separate exponentially. This, in turn, implies sensitive dependence of the dynamics on initial conditions, which is one of the most important characteristics of a chaotic system. Thus, contrary to noise, a chaotic signal is slightly predictable (i.e., predictable in the short term), but it is far less predictable than a purely deterministic signal. Moreover, the horizon of predictability is inversely proportional to the largest Liapunov exponent, λ_1. The estimation of the Liapunov exponents from an observed time series is by no means an easy task. The method due to Wolf et al. [71] almost always gives accurate values, but only for the largest Liapunov exponent. The method described by Brown et al. [72] uses a higher-order polynomial fit to construct the Jacobian matrix, and therefore has the potential capability to measure the complete Liapunov spectrum.

As already mentioned, for a process to be chaotic, there must be at least one positive Liapunov exponent. The sum of all positive Liapunov exponents is equal to the Kolmogorov entropy [69]. For an observed time series, Schouten et al. [73] describe a maximum likelihood procedure for estimation of the Kolmogorov entropy.

Another important characteristic of a chaotic system is the attractor dimension, which gives the amount of information necessary to specify the position of a point on the attractor to within a prescribed accuracy. An attractor represents the equilibrium state of a nonlinear dynamical system, which may be observed experimentally after the transients have died out. The attractor dimension may also be viewed as a lower bound on the number of essential variables needed to model the nonlinear dynamics of the system [74]. All of the metric dimensions of an attractor take on a common value called the fractal dimension. One such popular dimension is the correlation dimension due to Grasberge and Procaccia [75]:

$$D_C = \lim_{r\to\infty} \frac{\log C(r)}{\log r}$$

The parameter $C(r)$ in the numerator is the correlation integral, defined as the probability that two random points on the attractor are separated by a distance smaller than r. Procedures for estimation of the correlation dimension from an observed time series are described in [76,77].

By the same token, all of the frequency-dependent dimensions of an attractor take on a common value called the dimension of the natural measure [74]. Furthermore, the dimension of the natural measure is typically equal to the Liapunov dimension, which is also referred to as the Kaplan-Yorke dimension [58]. this quantity is defined in terms of the Liapunov spectrum as follows:

$$D_L = K - \frac{\sum_{i=1}^{K} \lambda_i}{\lambda_{K+1}}$$

where K is the integer for which we have:

$$\sum_{i=1}^{K} \lambda_i > 0 \quad \textbf{and} \quad \sum_{i=1}^{K+1} \lambda_i < 0$$

Note that the Liapunov exponent λ_{K+1} is negative. The Liapunov dimension provides a lower bound on the fractal dimension (i.e., correlation dimension) of the attractor [74, 78].

From a signal processing perspective, an issue of paramount importance is the reconstruction of dynamics from measurements made on a single coordinate of the system. The motivation here is to make "physical sense" from the resulting time series, bypassing a detailed mathematical knowledge of the underlying dynamics. The reconstruction of dynamics using independent coordinates from a time series was first advocated by Packard et al. [79]; however, this paper does not give a proof, and uses "derivative" embeddings rather than time-delay embeddings. The idea of time-delay or delay coordinate embeddings is attributed to Ruelle and Takens. Specifically, in 1981, Takens [80] published a mathematically profound paper on time-delay embeddings, which applies to attractors that are surfaces, like a torus. Takens' paper is difficult to read by non-mathematicians. The idea of delay coordinate mapping was refined in 1991 by Sauer et al. [81]. The approach taken in this latter paper integrates and expands on previous results due to Whitney [82] and Takens [80]. The delay embedding theorem, or more precisely, the fractal delay embedding prevalence theorem, for dynamic reconstruction derived in [81] holds not only in the Takens sense but also a probabilistic sense. To do so, Sauer et al. introduce a new theory called "prevalence" theory.

Suppose now we have a time series denoted by $x(nT_S)$, $x(nT_S - T_S)$, $x(nT_S - 2T_S)$,..., where T_S is the sampling period.

To proceed with the dynamic reconstruction, we first set up a vector in D dimensions by using a set of time delays, as shown by

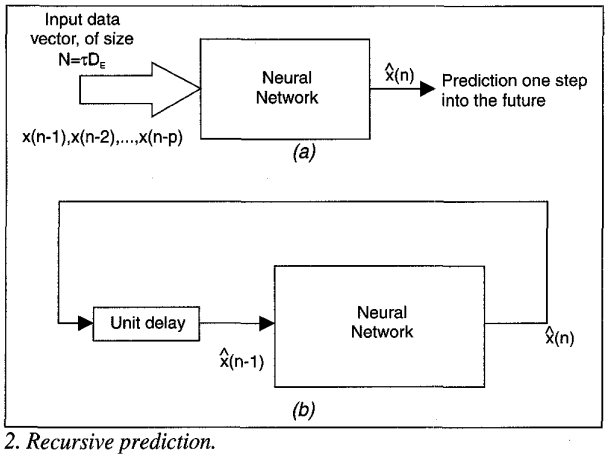
$$\mathbf{y}(n) = [y(n), y(n-1),..., y(n-D+1)]$$

The kth element in the vector $\mathbf{y}(n)$ is related to an element in the original time series as follows:

$$y(n-k) = x((n-k\tau)T_S), \qquad k = 0, 1,..., D-1$$

where τ is a positive integer called the normalized embedding delay. According to the delay embedding theorem, dynamic reconstruction by means of a delay coordinate map is possible provided that the dimension D is large enough. The procedure of finding a suitable D is known as embedding, and the minimum integer D that achieves dynamic reconstruction is called the embedding dimension; it is denoted by D_E. The delay coordinate map takes the form of a nonlinear predictive model that operates on the input vector $\mathbf{y}(n)$ to make a prediction of the next sample y(n+1). Procedures for estimating the embedding delay, τ, based on mutual information, are described in [76,83]. For estimating the embedding dimension D_E, one may use the method of false nearest neighbors described in [84].

The time series must be large enough for estimating the attractor dimensions and the Liapunov exponents, and computing the delay coordinate map. For an authoritative review of chaotic dynamics, see the paper by Aberbanel et al. [85]; for additional information of interest, see the review paper by Tong [86].



2. Recursive prediction.

independent of radar parameter, sea state, and radar location. However, unlike the Liapunov spectrum, the correlation dimension is essentially independent of the radar system used to perform the data collection.

Sea clutter is indeed generated by a chaotic process. But the mechanism by which this chaotic process actually arises in physical terms is unknown.

With this background on the chaotic nature of sea clutter, we may now turn attention to its signal processing implications. Specifically, we wish to (1) demonstrate that sea clutter permits a nonlinear predictive model with a significant horizon of predictability, and (2) describe a novel radar application exploiting this predictive capability.

The nonlinear predictive modelling of sea clutter involves the use of recursive (iterated) prediction [12, 93], illustrated in Fig. 2. This is a difficult procedure designed to test the generalization capability of the model. For the case study presented here, a neural network is used as the predictive model. There are two separate operations to be considered:

- The neural network is trained to operate as a one-step predictor, as depicted in Fig. 2a. A set of p samples $x(n-1)$, $x(n-2),...,x(n-p)$ is applied to the input layer of the network, and its synaptic weights are adjusted to minimize the prediction error [i.e., the difference between the actual sample value $x(n)$ and the predicted value, $\hat{x}(n)$] in a mean-square sense. For this to be attained, the size of the training set has to be large enough, and the training session would have to be continued until the synaptic weights of the network reach steady-state values, whereafter they are fixed.

- The neural network is next tested for its generalization performance, as depicted in Fig. 2b. The network is initialized by presenting it a set of samples $x(p-1)$, $(p-2),...,x(1)$ that have not been seen by the network before. The resulting prediction $\hat{x}(p)$ is delayed by one time unit and then fed back to the input. Correspondingly, the samples $x(p-1),...,x(2)$ are each delayed by one time unit, and the oldest sample $x(1)$ is dropped to make room for the delayed prediction $\hat{x}(p)$. The set of p samples so obtained is used to make a new prediction, and the process is repeated until all the original samples used to do the initialization have been removed from the recursive prediction process. From that point on, the neural network operates in a completely autonomous fashion, producing a time series that is representative of the dynamics learned by the neural network as a result of the training process.

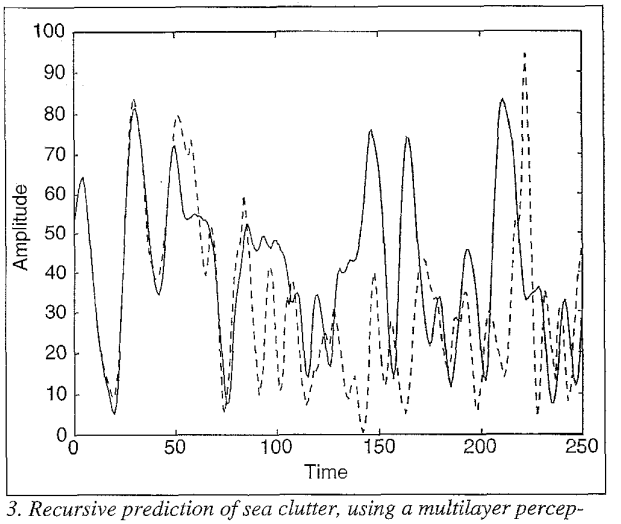
For the predictive modelling of sea clutter, we used a multilayer perceptron trained with the backpropagation algorithm. The size of the input layer, denoted by p, is chosen in accordance with the formula $p \geq \tau D_E$, where D_E is the embedding dimension, and τ is the embedding delay (normalized with respect to the pulse repetition period). In practice, it is inadvisable to choose p much larger than the lower bound, τD_E, as the effect of additive noise contaminating the input radar data would become more pronounced. Based on measurements on real-life data reported in [66], the embedding dimension D_E for sea clutter is estimated to be 10. Also, for the particular radar (operating at a pulse repetition fre-

quency of 1 kHz) used in those measurements, the normalized embedding delay, τ, is estimated to be 5. Thus, for the problem at hand, choosing $p \geq 50$ is logical.

To illustrate the importance of this lower bound on p, Fig. 3 shows the results obtained using recursive prediction performed on a multilayer perceptron that had already been trained (using the back-propagation algorithm) on actual sea clutter data. The size of the training dataset was 10^4 clutter samples. The multilayer perceptron had two hidden layers:

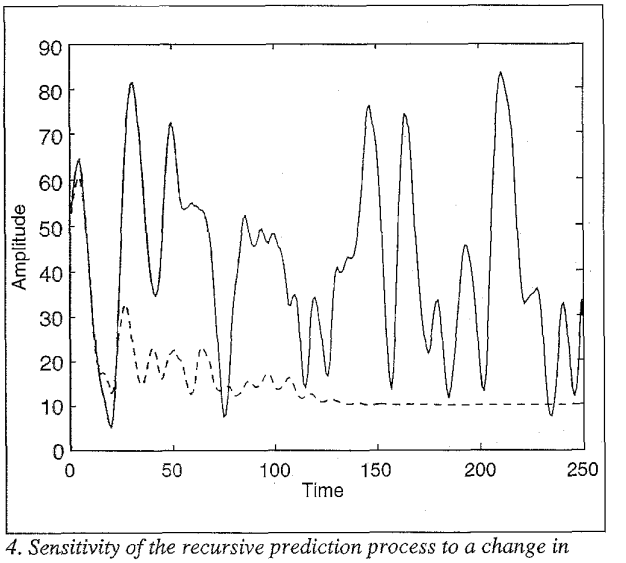
Input layer: 50 source nodes
First hidden layer: 80 neurons
Second hidden layer: 55 neurons
Output layer: 1 neuron

The neurons in both hidden layers used a logistic function for their activation functions, whereas the output neuron was linear. The solid line in Fig. 3 is the actual sea clutter waveform, and the dashed curve is the result of recursive prediction. The origin in this figure corresponds to the end of the initialization procedure. We see that for about the first 50 points shown in Fig. 3, the predicted and actual waveforms of sea clutter match fairly closely and thereafter they diverge.



3. Recursive prediction of sea clutter, using a multilayer perceptron with 50 source nodes, two hidden layers with 80 and 55 neurons, respectively, and one output neuron. The solid curve refers to the original sea clutter waveform. The dashed curve refers to the recursive predicted waveform, for which the first 50 points of the sea clutter set (not shown in the figure) are used as the initial starting point.

This result confirms that sea clutter produced by a noncoherent radar (i.e., one that relies on amplitude information alone) is locally predictable. Moreover, the horizon of predictability, namely, 50, is approximately equal to the inverse of the largest Liaponuv exponent, 0.03. (For an accurate calculation of the horizon of predictability, see [66].) The fact that a neural network with an input layer of the right size can be trained to learn the underlying nonlinear dynamics of sea clutter is further testimony for the important observation



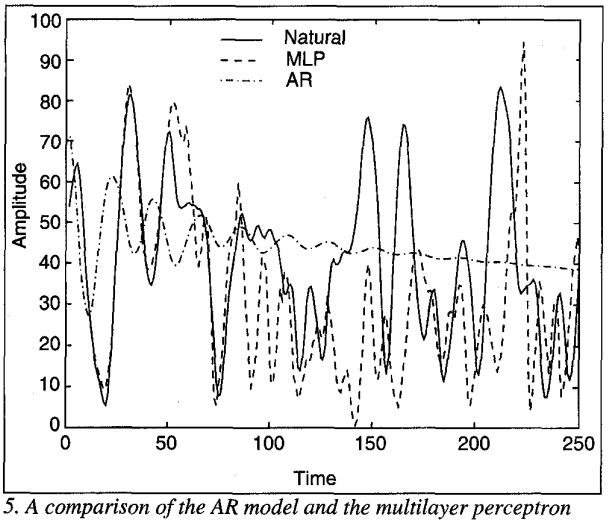
4. Sensitivity of the recursive prediction process to a change in the neural network design. The network used for this experiment consists of an input layer with 45 source nodes, two hidden layers with 80 and 55 neurons, respectively, and one output neuron.

made previously: the generation of sea clutter is governed by a coupled system of nonlinear differential equations. In effect, the neural network provides an approximation to such a system.

Figure 4 shows the result of a recursive prediction performed by a multilayer perceptron with $p = 45$, which is slightly smaller than the lower bound of 50 defined above. Except for this difference, the model has two hidden layers, with 80 neurons in the first one and 55 neurons in the second one, and a single linear output neuron as before. Moreover, the model is trained with the same data set and of the same size used to obtain the result shown in Fig. 3, and the recursive prediction procedure is used to test the model after completing the training session in exactly the same way as before. There is a dramatic difference between the results shown in Figs. 3 and 4. In particular, when the size of the input layer of the multilayer perceptron model is not large enough, the model fails to capture the underlying dynamics of sea clutter. Clearly, the choice of a neural network predictor that violates the lower bound on the size of p is unacceptable.

To emphasize the need for a nonlinear predictive model, we show Fig. 5, the recursive prediction results obtained using (a) an autoregressive (AR) model, and (b) a multilayer perceptron model. Both models used 50 delay taps for the input, in accordance with the lower bound of $p \geq 50$. Clearly, the AR model, which is linear, fails completely to capture the underlying dynamics of sea clutter.

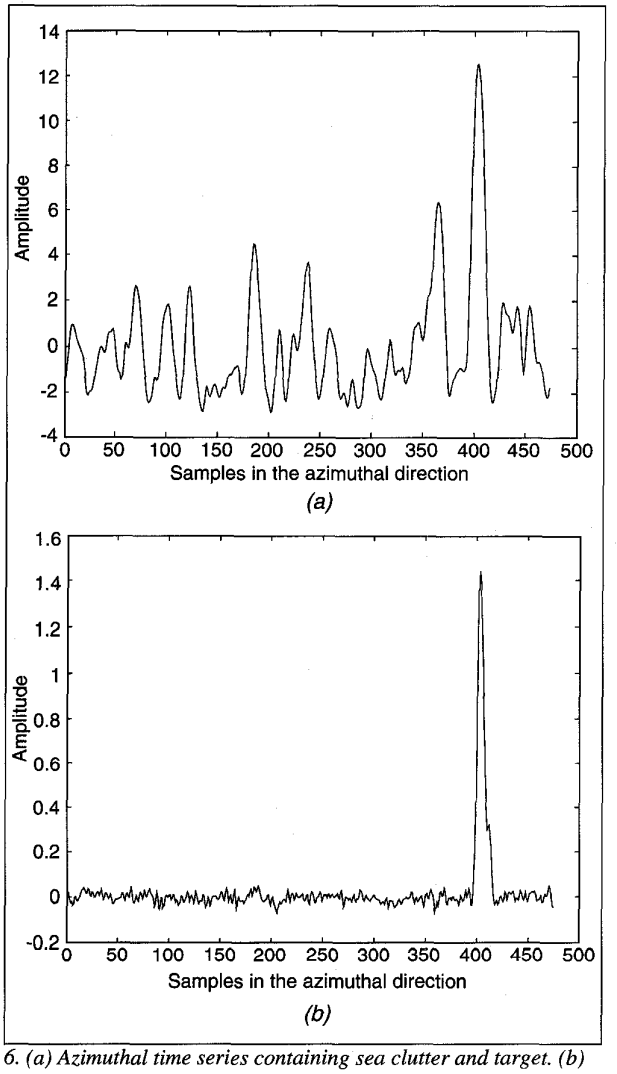
Turning next to the radar application of the predictive modelling of sea clutter, Fig. 6 shows the results of another experiment involving an off-the-shelf commercial noncoherent marine radar operating in a scanning mode [94]. In this application, the multilayer perceptron, trained on examples drawn from sea clutter and then having its synaptic weights fixed, acts as a clutter (interference) canceller. In particular, through training, the network acquires the function of a



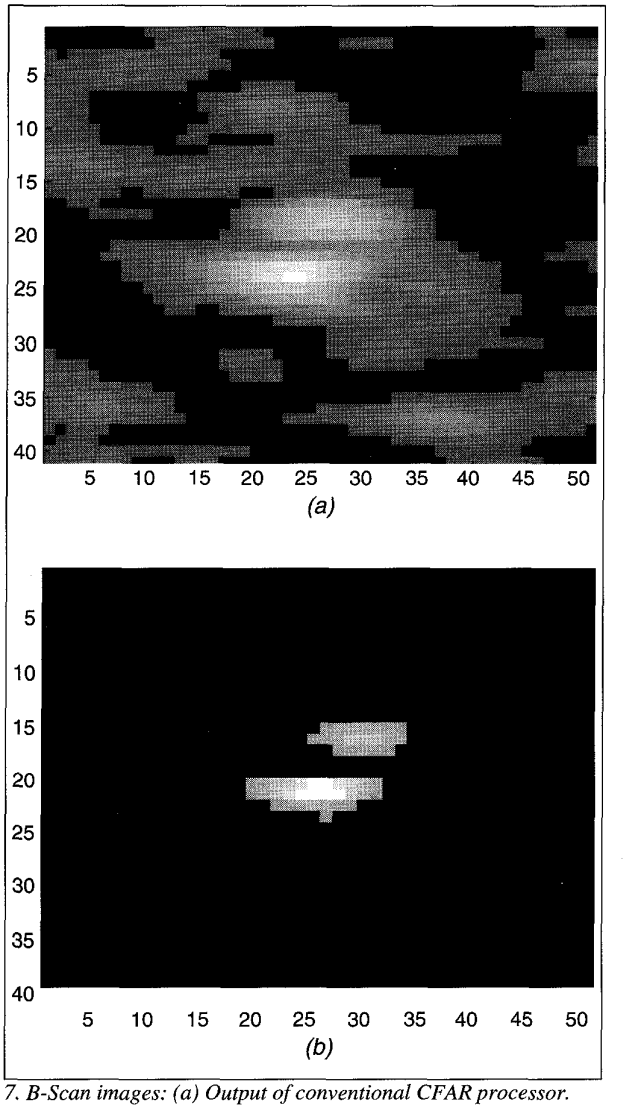
5. A comparison of the AR model and the multilayer perceptron (MLP) model for reconstruction of the underlying dynamics of sea clutter.

clutter model. When it is fed with a received signal that consists of a target signal plus clutter, as in Fig. 6a, the presence of the target signal causes a corresponding perturbation in the output of the network. That is, the network suppresses the clutter component and thereby enhances the presence of the target signal at its output, as illustrated in Fig. 6b.

Figure 7 demonstrates another interesting property of the clutter canceller (nonlinear predictive model), using data collected with the same noncoherent marine radar employed for Fig. 6. The so-called B-scan (azimuth versus range) images shown in the two parts of Fig. 7 represent (a) the output of a conventional constant false-alarm rate (CFAR) processor, and (b) the output of the clutter canceller. In both cases, the images show the respective processor outputs prior to the application of a detection threshold. The input radar data set contains two closely spaced targets. While the echoes from these two targets are blurred together in the conven-



6. (a) Azimuthal time series containing sea clutter and target. (b) Prediction error at output of neural network. Target is clearly evident.



7. B-Scan images: (a) Output of conventional CFAR processor. (b) Output of neural network-based clutter canceller.

tional CFAR processor image, they are clearly separable in the neural-network processor image.

Case Study II: Modular Learning Strategy for Signal Detection in a Nonstationary Environment

In the first case study, we showed how a neural network, used as a nonlinear predictive model, can exploit prior knowledge, namely, the fact that sea clutter is chaotic. The only information used in that case study was the information contained in the amplitude of the received signal, which is provided by a noncoherent radar. This second case study pertains to the detection of a weak target signal corrupted by an interfering signal. Here, one or the other or both of these signals may be nonstationary, and no prior knowledge about the environment is invoked. However, these problems are ameliorated through the use of Doppler information in addition to amplitude information, which requires the use of a coherent radar. Case Studies I and II do have one thing in common: in both cases, the radar operates in an ocean environment, with sea clutter being the primary source of interference.

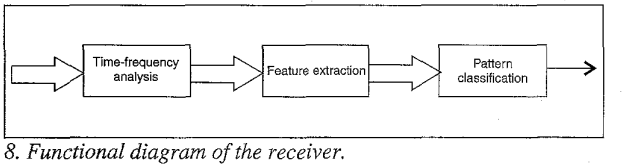
Now consider a novel modular learning strategy for signal detection that is motivated by the echo-location (sonar) of a bat, which detects, pursues, and captures its target (e.g., an insect) with a facility and success rate that is the envy of every radar or sonar engineer [95]. We are not suggesting that the modular detection strategy described in this article involves all the signal processing functions performed in the bat's echo-location system. What we are saying is that the principal functions that characterize the modular learning strategy are found in one form or another in the bat's echo-location system.

Figure 8 shows a block diagram of the basic detection strategy consisting of three fundamental functional blocks that are designed to perform time-frequency analysis, feature extraction, and pattern classification, in that order. This form of front-end processing is commonly used in pattern recognition tasks [106].

For the time-frequency analysis, we have chosen the Wigner-Ville distribution (WVD); Table 3 presents a summary of the important properties of the WVD. Among the family of bilinear time-frequency distributions, the WVD possesses two distinct advantages over other members of the family for signal detection [107]:

1. It is always a real-valued function.
2. It exhibits the least amount of spread in the time-frequency plane.

One criticism that is often made against the WVD is the generation of cross-terms, or more precisely, cross Wigner-Ville distributions, due to the combined presence of two (or more) components in the received signal. Various procedures have been developed in the literature for dealing with the cross Wigner-Ville distributions. In [108,109], for example, an algorithm known as the reduced interference distributions (RID) is described, which is designed to flatten the cross-terms so that they bounce up and down less, when compared to the standard form of the WVD. In so doing, the RID provides an "almost" positive distribution, which is what a time-frequency energy distribution should be, particularly for those applications that require the analysis of signals. For signal detection, it would be tempting to do the opposite (i.e., retain the cross Wigner-Ville distributions and suppress the auto-terms). The detection strategy would then focus solely on the presence or absence of the cross Wigner-Ville distributions. Such a procedure would, however, violate the information preservation rule by removing useful information contained in the auto-terms; its use is therefore not recommended for signal detection.



8. Functional diagram of the receiver.

In a clutter-dominated environment, which is the environment of interest in Case Study 2, the cross-terms arise only when a target signal is present. Thus, the presence of such terms is in fact an asset. We say this because the terms provide another feature that can enhance the visibility of a target in the time-frequency image resulting from the application of the WVD. Indeed, the cross-terms are essential to the optimal information-preserving property of the WVD. To appreciate the importance of the WVD for the radar detection problem, we present three sample WVD images of real-life radar returns. These represent three different situations pertaining to an ice-infested ocean environment using a coherent radar [110-111]:

- Strong radar return from a large ice target, shown in Fig. 9a.
- Relatively weak radar return from a small ice target, shown in Fig. 9b.
- Sea clutter alone, shown in Fig. 9c.

The WVD images presented in this figure significantly differentiate between these three scenarios. Unfortunately, the use of the WVD leads to a significant increase in the amount of redundant information contained in the time-frequency image of a radar signal. To improve computational efficiency, it is therefore necessary to follow up the WVD with some form of data compression. (This point is also made in [112, 113], where singular value decomposition is used for the extraction of features from the WVD image of a signal for the purpose of signal detection or classification. However, the scheme described therein is primitive compared to the modular learning strategy embodied in Fig. 10, as it lacks a learning capability and does not address the two fundamental questions raised later in this case study.)

Table 3: The Wigner-Ville Distribution

Consider an analytic signal $x(t)$, whose real and imaginary parts are related by Hilbert transformation [96,97]. The Wigner-Ville distribution (WVD) of the signal $x(t)$ is defined by [98-100]:

$$W(t,f)=\int_{-\infty}^{\infty} x\left(t+\frac{\tau}{2}\right)x^*\left(t-\frac{\tau}{2}\right)\exp(-j2\pi f\tau)\,d\tau$$

where the integration is carried out with respect to the delay parameter τ. In effect, the signal $x(t)$ acts as its own window function. In simple terms, the WVD is said to be bilinear because x(t) is used twice in its computation. Formulation of the WVD makes it directly applicable to complex baseband signals that are commonly encountered in coherent radar, sonar, and communication systems. The Wigner-Ville distribution is the prototype of joint time-frequency distributions that are different from the Fourier-based spectrogram. Wigner [101], the originator of the distribution, was motivated by ideas in quantum mechanics. Some fifteen years after Wigner's paper, Ville [102] introduced the distribution into signal analysis. In [103], Cohen provided a consistent set of definitions for joint time-frequency distribution, which has played a key role in guiding and clarifying research in this area. For a more detailed historical account of the Wigner-Ville distribution, see Cohen [99].

The WVD provides a high-resolution representation of the signal $x(t)$. Its important properties are summarized here [98-100]:

- The WVD is always real, even if the signal x(t) is complex.
- The WVD satisfies the time-frequency marginals, in terms of the instantaneous power in time:

$$\int_{-\infty}^{\infty} W(t,f)\,df=|x(t)|^2$$

and in terms of the energy spectral density in frequency:

$$\int_{-\infty}^{\infty} W(t,f)\,dt=|X(f)|^2$$

where

$X(f)$ = Fourier transform of $x(t)$

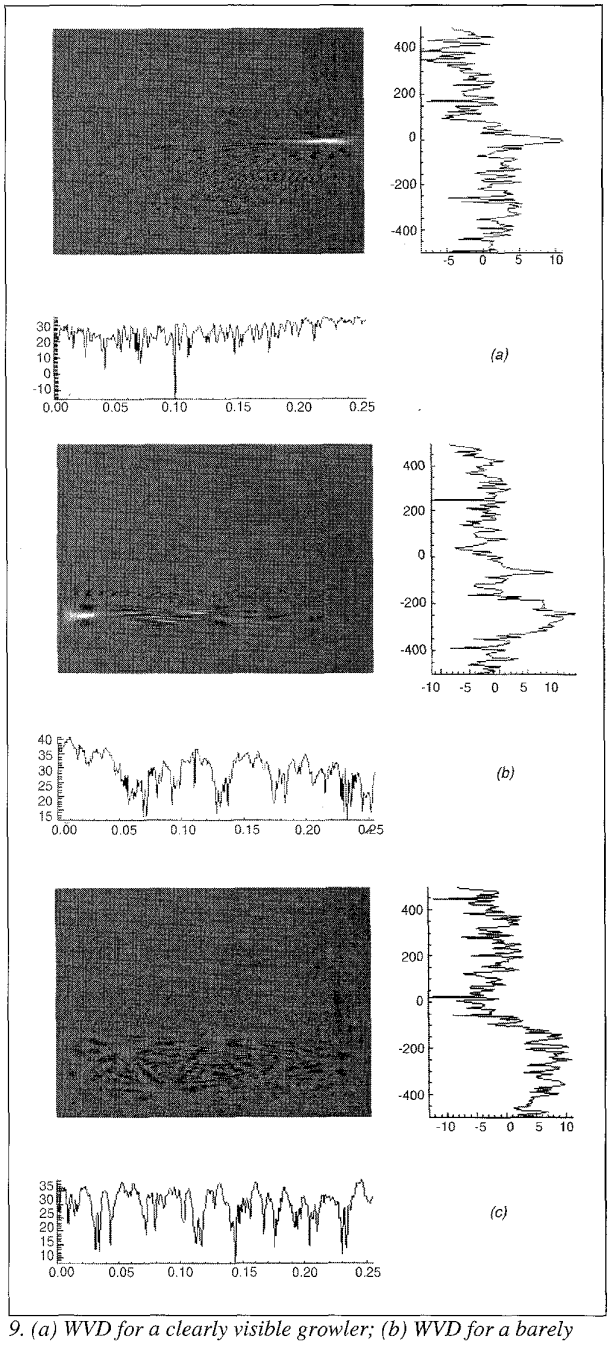
$$=\int_{-\infty}^{\infty} x(t)\exp(-j2\pi ft)\,dt$$

- If the signal $x(t)$ is shifted by time t_0 and/or its Fourier transform $X(f)$ is shifted by frequency f_0, then the shifted WVD is defined by $W(t-t_0, f-f_0)$.
- The WVD is an information-preserving transformation in that the original signal $x(t)$ can be uniquely recovered from the time-frequency representation $W(t,f)$ to within a constant phase factor; see also [104,105].
- A consequence of the property that the WVD satisfies the marginals is the fact that it cannot be guaranteed to be positive throughout the time-frequency plane; that is, the WVD violates the condition of positivity for a proper time-frequency energy distribution, which may therefore lead to results that cannot be interpreted.
- When the signal $x(t)$ consists of multicomponents, the WVD may be categorized into (1) auto-terms representing the Wigner-Ville distributions of the individual components of x(t) by themselves, and (2) cross-terms representing the cross Wigner-Ville distributions between them.

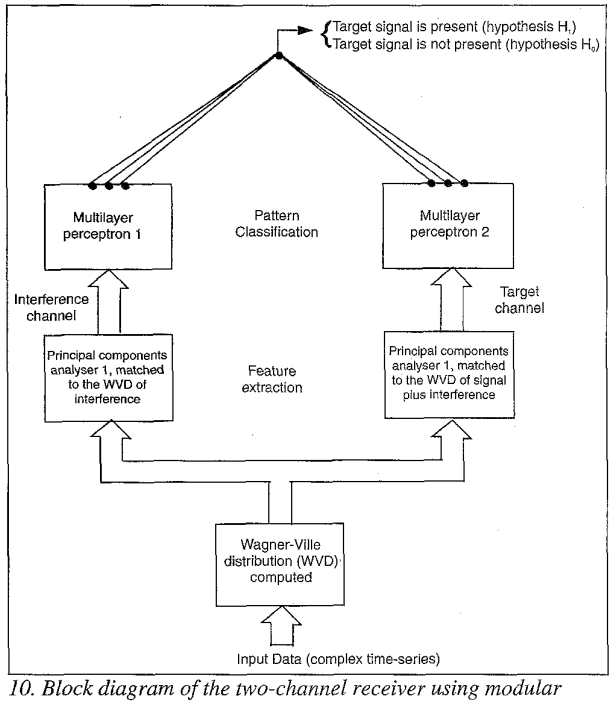
The latter two properties can be troublesome when the issue of interest is one of signal analysis. However, they are not viewed to be a serious factor when signal detection is the primary objective; in such an application the cross-Wigner-Ville distributions contribute to the information-preserving property of the WVD.

A signal processing tool that is well suited for data compression is principal components analysis (PCA) [106]. Basically, the PCA performs an eigendecomposition on a square matrix (in our case, the covariance matrix of the time series obtained by scanning the WVD image of the incoming radar signal on a column-by-column basis, with each column representing a time slice), orders the eigenvalues in descending order, and retains the eigenvectors associated with the largest eigenvalues. The compressed signal is represented by a linear combination of the eigenvectors retained by the PCA. Thus, the PCA is instrumental in extracting a finite set of features for the WVD image that is optimum (among linear techniques), in that the original WVD image (and therefore the original received signal) can be reconstructed from these features in a minimum mean-square error sense. In other words, information loss brought on by the extraction of features by the PCA is minimized and quantifiable, and so we are still operating in the realm of the information preservation rule. It is conceivable that for our application nonlinear devices known as principal curves/surfaces [114] in statistics can do better than PCA. A similar capability is provided by Kohonen's self-organizing feature map [115], which is of a low dimension and used to approximate a higher-dimensional scatter-plot of samples. For a comparison of statistical and SOFM approaches, see Mulier and Cherkassky [116].

The final operation in our modular learning detection strategy is that of pattern classification, the purpose of which is to distinguish between two different time-frequency images on the basis of features extracted by the PCA. One image pertains to the presence of clutter alone. The other image pertains to the combined presence of clutter and the target signal of interest. The pattern classification process is typi-



9. (a) WVD for a clearly visible growler; (b) WVD for a barely visible growler; (c) WVD for sea clutter.



10. Block diagram of the two-channel receiver using modular learning strategy.

cally nonlinear, making the task that much more challenging to implement.

Figure 10 shows a block diagram of a neural network-based implementation of the modular detection strategy described herein [110-111]. It consists of two channels, one termed the clutter or interference channel, and the other termed the target channel. Both channels are fed from a common input representing the WVD image of the received signal. Each channel consists of a PCA network followed by a multilayer perceptron for pattern classification. The PCA networks are trained in a self-organized fashion, using the generalized Hebbian algorithm (GHA) due to Sanger [117]. Table 4 presents a summary of this algorithm. The PCA network in the clutter channel is trained by presenting it with WVD images known to represent clutter only, under varying environmental conditions. Once the training is completed, the synaptic weights of that PCA network are fixed. The training procedure of the PCA network for the target channel follows a similar procedure, except for the fact that its training examples consist of WVD images known to contain target plus clutter, under varying conditions. The outputs of the PCA networks may be viewed as a specific number of dominant projections of the input WVD space on two subspaces, with one subspace representing clutter alone and the other subspace representing target plus clutter. Typically, these two subspaces are unknown and nonlinear; projections of the WVD space onto them are therefore best learned by way of real-life examples that are representative of the two scenarios. The end result is that the PCA network in one channel is adaptively matched to clutter alone, and the PCA network in the other channel is adaptively matched to target plus clutter, hence the designations of the two channels in Fig. 10 as clutter (interference) and target channels, respectively.

Each multilayer perceptron has two hidden layers and an output layer with three output nodes. The output nodes are linearly combined into a single decision-making node. Thus, the decision as to whether a target is present or not is deferred to the very output of the system, in accordance with the information preservation rule. Specifically, if a threshold set for a prescribed probability of false alarm is exceeded by the overall output of the receiver, a decision is made that a target is present; otherwise, a decision is made that the received radar signal consists of clutter alone. The synaptic weights of

Table 4: Generalized Hebbian Algorithm for PCA

In the context of a biological neural network, Hebb's postulate of learning [118] states that the ability of one neuron in the network to cause the firing of another neuron increases when that neuron consistently takes part in firing the other. In other words, when an input neuron and output neuron of a biological neural network fire at the same time, the synaptic connection between those two neurons is strengthened.

In an artificial neural network in which neuronal interactions are modeled simply as a linear combiner, the output of neuron j with N input nodes is defined by

$$y_j = \sum_{i=1}^{N} w_{ji} x_i$$

where x_i is the input of synapse i characterized by synaptic weight w_{ji}. Applying Hebb's postulate of learning to this model, the synaptic weight w_{ji} is increased when the values of both x_i and y_j are large. In a more general setting, we may say that the synaptic weight w_{ji} is modified in accordance with the correlation between the input signal x_i and output signal y_j.

Building on Hebb's postulate of learning, Oja [119] derived the following learning rule for the simplified model of a neuron described above:

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k)\, y_j(k)\, [x_i(k) - y_j(k)\, w_{ji}(k)],$$
$$i = 1, 2, \ldots, N$$

where k denotes the number of iterations, and $\eta(k)$ is the learning-rate parameter; ordinarily, $\eta(k)$ is assigned a constant value. An interesting property of Oja's rule is that the weight vector made up of the elements $w_{j1}, w_{j2}, \ldots, w_{jN}$ converges to the first principal component of the input data as k approaches infinity, such that in the limit we have

$$\lim_{k=0} \eta(k) = 0 \quad \textbf{and} \quad \sum_{k=0}^{\infty} \eta(k) = \infty$$

Sanger [117] formulated the generalized Hebbian algorithm by extending Oja's rule to a feedforward network consisting of N input nodes and M neurons. In particular, the adaptation rule for synaptic weight w_{ji} of neuron j that is fed from input i is as follows:

$$w_{ji}(k+1) = w_{ji}(k) + \eta(k) y_j(k) \left[x_i(k) - \sum_{l \le j} w_{li}(k) y_l(k) \right]$$

for $i = 1,2,\ldots,N$, **and** $j = 1,2,\ldots M$

Under this rule, the weight vectors $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M$ of neurons $1,2,\ldots,M$, respectively, converge to the largest M principal components of the input data as the number of iterations k approaches infinity, such that in the limit the learning-rate parameter $\eta(k)$ satisfies the above-mentioned conditions. The generalized Hebbian algorithm includes Oja's rule as a special case.

the two hidden layers and output layers of both perceptrons and those of the linear combiner are trained simultaneously in a supervised manner by presenting the whole network with WVD images that are known to represent the following situations that can arise:

- Clutter alone
- Strong target return plus clutter
- Barely visible target return plus clutter

The training of these layers is performed using the back-propagation algorithm. As a matter of interest, the first hidden layer of each multilayer perceptron uses the notions of receptive fields and weight sharing described in [12, 120]. By "receptive field," we mean that each neuron in the first hidden layer is connected only to a finite set of neurons that lie in its local neighborhood in the input layer. By "weight sharing," we mean that all the receptive fields of the layer share the same set of weights. The use of receptive fields and weight sharing is designed to reduce the number of synaptic connections and possibly improve the generalization performance of the multilayer perceptrons.

There are two basic questions that need to be addressed in the context of the modular learning strategy for signal detection described in Fig. 10. First, what is the rationale for using two channels? To answer this question, we first note that in the traditional approach to radar target detection in a clutter-dominated environment, for example, we may use a "best" mismatched filter for clutter discrimination [121]. In such an approach involving a single channel in the receiver, the requirement for best performance in additive noise is traded for an improvement in performance in clutter by purposely mismatching the filter. We may avoid the need for this trade-off in performance by using two nonlinear matched filters as depicted in Fig. 10, with each filter being adaptively matched to the received signal arising under one of the two hypotheses that are to be distinguished. In addition, the use of two different channels as described herein provides two independent assessments of the decision that should be taken, given the received signal. A simple and yet effective method of integrating the two channel outputs is through the use of linear combining [122], which is precisely what has been done in designing the modular learning strategy of Fig. 10.

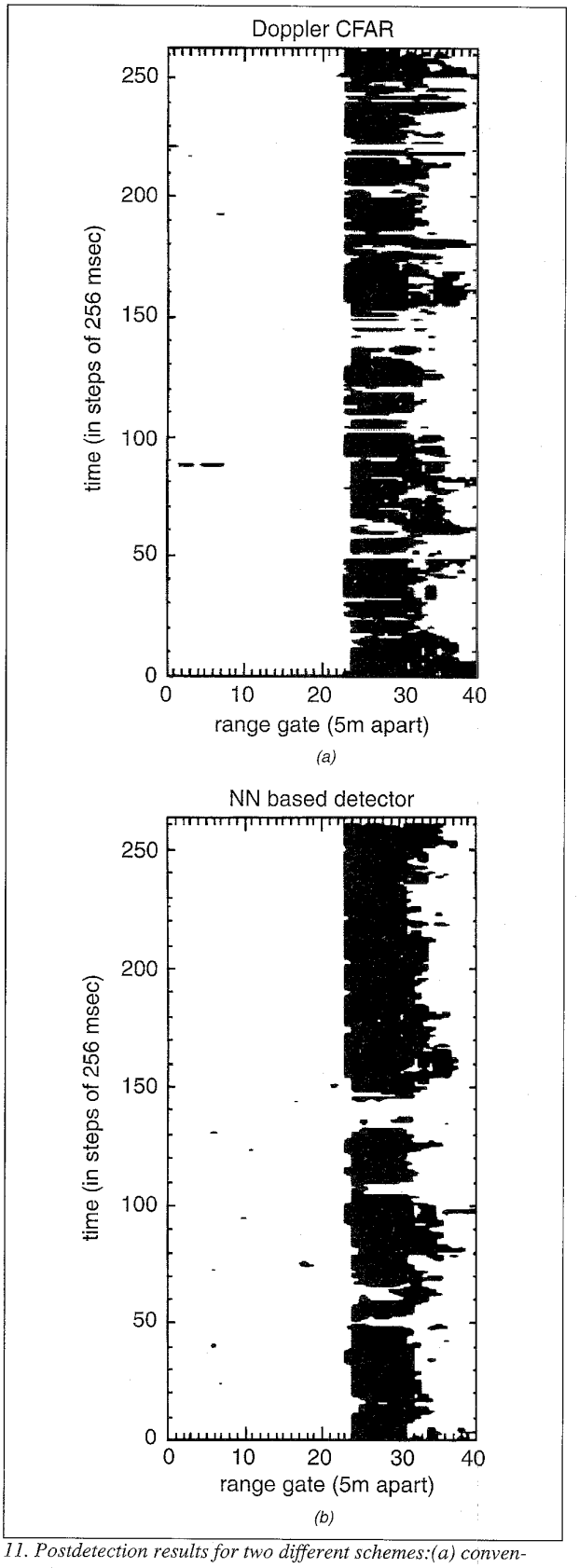
This strategy therefore makes it possible to learn how to classify the two sets of features that have been learned by the PCA networks in the target and clutter channels. An alternative way to state this is that it exhibits a "learning to learn" capability.

The second question is: why does each channel have three output nodes? The nonlinear input-output mapping produced by the modular learning strategy of Fig. 10 depends, among other things, on the number of output nodes per channel. The use of two output nodes per channel severely limits the capacity of each multilayer perceptron to classify the received signal. On the other hand, the use of three output nodes per channel makes it possible to provide a finer classification of the received signal by saying

- the received signal contains a strong target signal
- the received signal contains a weak target signal, or
- the received signal contains clutter alone

This, in turn, has the beneficial effect of reducing the overlap between the two primary classes of interest: target is not present (null hypothesis), and target is present (the other hypothesis). Consequently, the receiver with three output nodes per channel has the potential of outperforming the receiver with two output nodes per channel. Indeed, experimental results presented in [110, 111] bear out the validity of this statement.

Figure 11 presents a comparison of the detection results obtained for the modular receiver of Fig. 10 with those of a conventional Doppler CFAR (constant false alarm rate) processor for a false alarm rate set at 10^{-3}. Here, black denotes the presence of a target signal, and white denotes clutter. With the target constantly being in the range of the radar (i.e., for all time), we should ideally see a continuous black strip (representing the target) in a light background (representing the clutter). In light of this observation, a significant improvement in the modular learning strategy is found by filling in the periods of "silence" that are observed in the detection performance of the conventional Doppler CFAR processor. This "silence" is caused by the partial obscuration of the target (a small piece of ice in the experiment described herein) by an ocean wave in front of it, or the dipping of the target in a wave trough. The performance displayed in Fig. 11 is quite remarkable, since the modular learning system is able to perform satisfactorily even in a situation when the target returns are weak; in other words, a barely visible target has been made "visible in signal processing terms". The other observation made from Fig. 11 is the occasional blanking of a signal from the target (as seen in the middle of the plot); in such cases, there is no way any method would be able to detect the target since, insofar as the radar is concerned, the target is simply not there to be seen. In [111], experimental results are presented demonstrating that the modular learning system of Fig. 10 has a robust performance with respect to wide variations in sea state.



11. Postdetection results for two different schemes:(a) conventional Doppler CFAR processor; (b) modular learning scheme of Fig. 10.

In summation, it may be difficult, in the traditional approach to radar receiver design, to make provisions for real-life situations in a manner similar to that attainable with the modular learning strategy described in Fig. 10. Unfortunately, however, the highly complex composition of this structure defies a detailed mathematical analysis. Moreover, to design it, one would need a sufficiently large training set that is truly representative of the operational environment. Once the training process is completed and all the synaptic weights of the PCA networks and multilayer perceptron classifiers are computed and thereafter fixed, and the receiver is ready for normal operation, signal propagation through the network is very rapid. This is basically due to the fact that both the PCA networks and the multilayer perceptron classifiers consist of feedforward structures.

Case Study III: Mixture of Principal Components for Image Compression and Segmentation

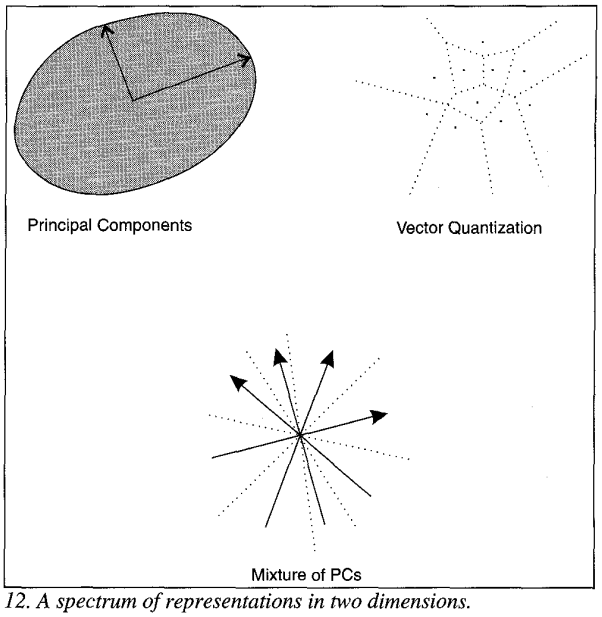
Our third and final case study pertains to the use of neural networks for image compression and segmentation. The study of image compression methods has been an active area of research since the inception of digital imaging. Successful image compression schemes must satisfy two conflicting requirements:

- During the coding phase of image compression, data are transformed from their native format, typically an array of gray level or trichromatic pixels, into a new format that requires less bandwidth or storage.

- The transformation must preserve the essential information content of the original image, so that the difference between the original and decoded images is not perceptually discernible. The significance of this difference must be clearly evaluated within the context of the end use of the image. For example, medical images must not lose their diagnostic value after compression.

A major problem with many image processing applications is their implicit assumption of stationarity. The fallacy of this assumption is the reason why many conventional image processing techniques perform poorly in the vicinity of edges. Here, the image statistics tend to be radically different from the global statistics of the image. Conventional image compression methods, such as the Karhunen-Loève transform (KLT) [106], are designed according to a globally optimal mean-square error criterion. However, the aforementioned nonstationarity of edge regions makes this criterion far from ideal. Therefore we may say that if an image compression method can be made to adapt to local nonstationarities in the image, then its performance would be superior to that of the KLT.

To account for variations in the local statistics of an image, a transformation must have the capability to adapt locally. In [123-124], a new family of adaptive transform coding methods, called a mixture of principal components (MPC), is described. Specifically, it combines desirable attributes of both principal components analysis (PCA) [106] and vector quantization (VQ) [125]. Within a class, an input vector is represented by a continuous, linear combination of M basis vectors of the subspace in a manner analogous to the PCA representation. But, because of the partitioning of the data into a discrete number of regions or classes, the MPC effects a nonlinear mapping of the input data in a manner analogous to VQ. The relations between these three methods of representation are illustrated in Fig. 12 for a two-dimensional example:



12. A spectrum of representations in two dimensions.

1. The PCA approach forms a complete, continuous representation of input data using, in this example, a linear combination of two basis vectors, as indicated in Fig. 12(a).

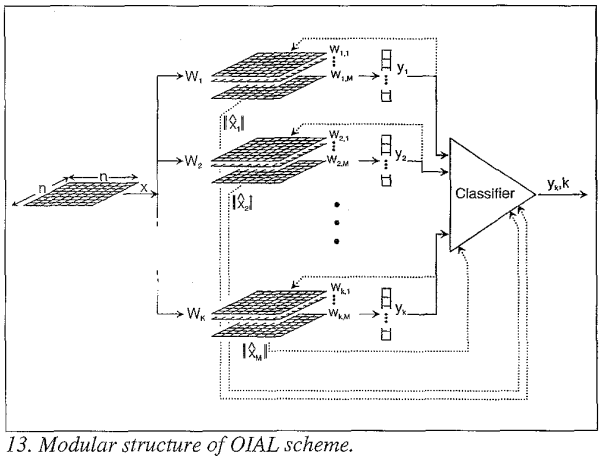
2. With VQ, the input data are represented in a purely discrete manner by partitioning the input space, in this example, into 10 distinct regions and representing each region by a Voronoi center, as indicated in Fig. 12(b).

3. The MPC lies between these two extremes of data representation, as indicated in Fig. 12(c):

· In a manner similar to the VQ, the input space is partitioned, in this example, into 4 distinct regions.

· Within each region, the input data are represented, in this example, by a single basis vector. Thus, like PCA, the data are given a continuous representation.

For higher-dimensional input spaces, the number of basis vectors used in MPC may be two or more, in which case we find that planes, hyperplanes, or higher-dimensional subspaces are formed within the input space.



13. Modular structure of OIAL scheme.

Figure 13 shows a network structure for implementing one particular form of the MPC. The system is modular, consisting of a number of modules corresponding to different classes of input data. Each module consists of a linear transformation, whose basis vectors are computed using an initial training period. The appropriate class for a given input vector is determined by the subspace classifier. The system utilizes a learning algorithm referred to as the optimallv integrated adaptive learning (OIAL) algorithm. The algorithm is self-organized, combining both Hebbian learning and competitive learning; it produces an adaptive linear transformation that tries to minimize the mean-squared error between the input data and the decoded data, beyond that attainable with the KLT. As such, the learning algorithm is well suited to the task of image compression. A summary of OIAL is presented in Table 5.

Figure 14a shows the magnetic resonance image (MRI) used for training. The image in Fig. 14b is the adjacent section from the same study (patient), which was used for testing. Each image consists of 256 x 256 pixels, with the dynamic range of 8 bits or 256 gray levels. The training image was divided into blocks of 8 x 8 pixels for an input dimension of $N = 64$. The blocks were overlapped at two pixel intervals for a total number of 15,625 training samples. During training, the samples were presented in random order. For comparison, the KLT was calculated based on the same training data.

The test image was divided into 8 x 8 non-overlapping blocks. These blocks were transformed by the previously computed system into a set of coefficients, quantized, and then transformed back into image blocks. The coefficients were quantized in a similar manner to that of the JPEG standard. The first coefficient was coded via first-order DPCM using a uniform quantizer. The remaining coefficients were coded via PCM using a uniform quantizer. For a given coding rate, the same quantization interval was used for all

Table 5: Optimally Integrated Adaptive Learning Algorithm

Let an image be subdivided into blocks of n-by-n pixels. These blocks can be considered as N dimensional vectors x with $N = n^2$. The general form of the class of optimally integrated adaptive learning (OIAL) is as follows [123]:

1. Initialize K transformation matrices $\{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_K\}$, where $\mathbf{W}_i$ is an M-by-N dimensional matrix with $M \leq N$; the lth row of $\mathbf{W}_i$ is denoted by $w_i^{(l)}$, $i = 1,2,..., M$.

2. For each training input vector $\mathbf{x}$:

(a) classify the vector based on the subspace classifier

$$\mathbf{x} \in C_i \quad \text{if} \quad \|\mathbf{P}_i \mathbf{x}\| = \max_{j=1}^{K} \|\mathbf{P}_j \mathbf{x}\|$$

where $\mathbf{P}_i = \mathbf{W}_i^{\mathrm{T}} \mathbf{W}_i$, and

(b) update the transformation matrix $\mathbf{W}_i$ according to:

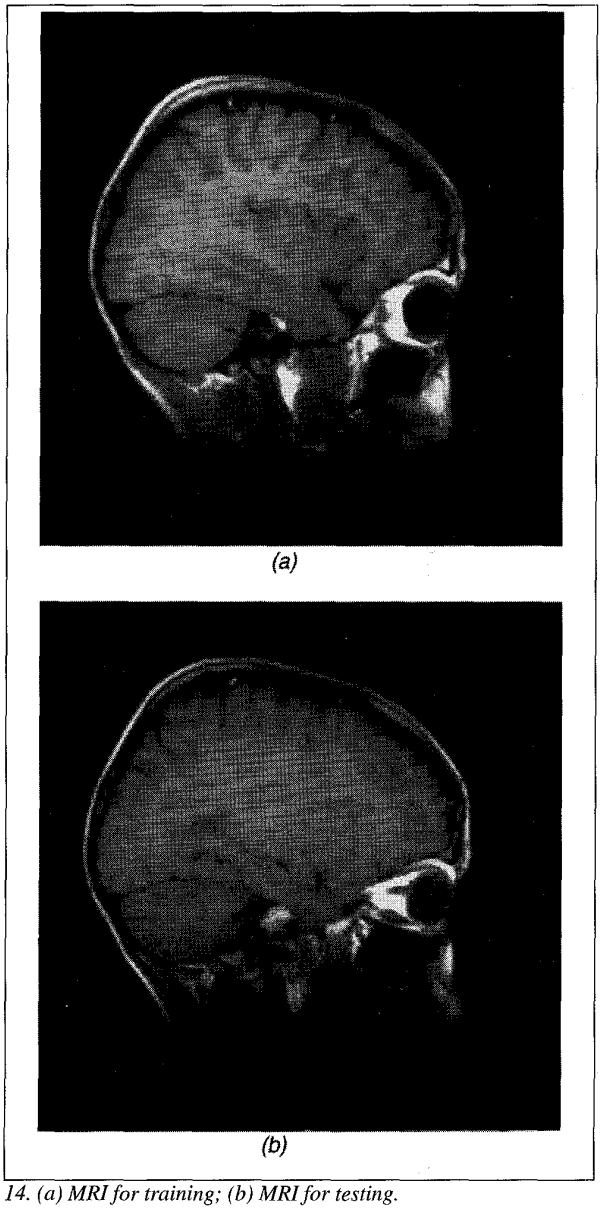
$\mathbf{W}_i(k+1) = \mathbf{W}_i(k) + \eta Z(\mathbf{w}, \mathbf{W}_i(k))$ where η is the learning rate parameter, and $Z(\mathbf{x}, \mathbf{W}_i)$ is a learning rule that converges to the M principal components of $\{\mathbf{x} \mid \mathbf{x} \in C_i\}$

3. Repeat for each training vector until the transformations converge.

In the first step, care must be taken in the choice of the initial set of transformation matrices. For good performance they should be representative of the distribution space of the training data in which case the OIAL algorithm acts as a fine tuner. If some of the $\mathbf{W}_i$'s were to be initialized to values corresponding to regions outside of the distribution space, then the resulting partition would be suboptimal. There are a number of methods to reduce the possibility of this occurring as described here [123]:

- Arbitrarily partition the training set into K classes and estimate the corresponding transformations using either iterative learning rules or batch eigendecomposition.
- Use a single fixed-basis transformation such as the discrete cosine transform (DCT) and add a small amount of random variation to each class to produce a set of unique transformations.
- Use an estimate of the global principal components of the data with a small amount of random variation added to each class.

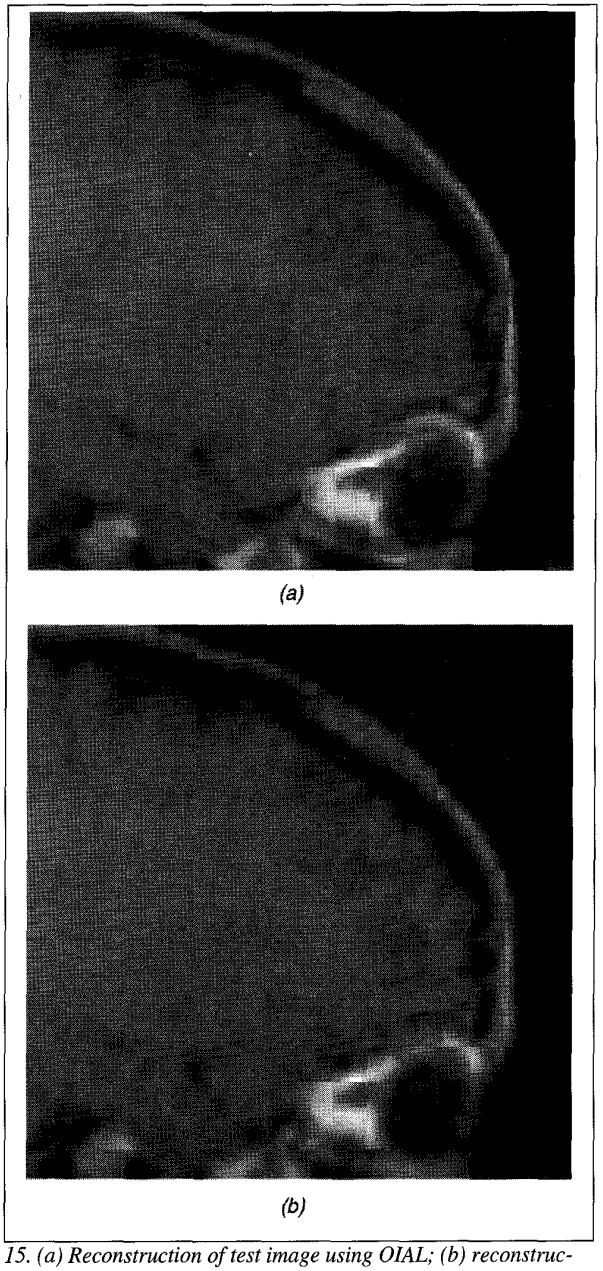
It is this latter approach that is used in the experimental results reported in [123]. With this form of initialization, OIAL acts as a "fine tuning" filter adapting to local nonstationarities, particularly those that exist around edges in the image.



14. (a) MRI for training; (b) MRI for testing.

the coefficients. The quantized data were then Huffman-coded with a codebook optimized for Laplacian distributions. The number of bits assigned for the class information was simply $\log_2 K$ bits per block. Different bit rates were used in the quantization step. For the KLT, an identical coding scheme was used except, of course, that no additional bits per block were required to code the class assignment.

For the new approach, the overall mean-squared error was reduced and the perceptual image quality was improved, when compared to the KLT. More image details were preserved and fewer artifacts were introduced. In particular, Fig. 15a shows the details of the new coding scheme with 128 classes, 4 coefficients per block, at 0.25 bpp, while Figs. 15b shows the corresponding details of KLT coding at the same bit rate of 0.25 bpp. When examining the detailed structure of these two images, it is clear that the OIAL image preserves more features than the KLT image. In the upper forehead region near the skull, the dark line of the outer table of the skull between the white line of the skin and the white line of the diploe (i.e., hard porous tissue between the walls of the cranial bones) is visible in the OIAL, but completely obscured in the KLT. The same is true of the detail in the top portion of the orbit. Not only does the KLT lose information, it also introduces texture variations that are not present in the original image nor in the OIAL image. This texture interferes with the visibility of the sulci in the outer portion of the brain.



15. (a) Reconstruction of test image using OIAL; (b) reconstruction of test image using KLT.

16. Original Lena image.

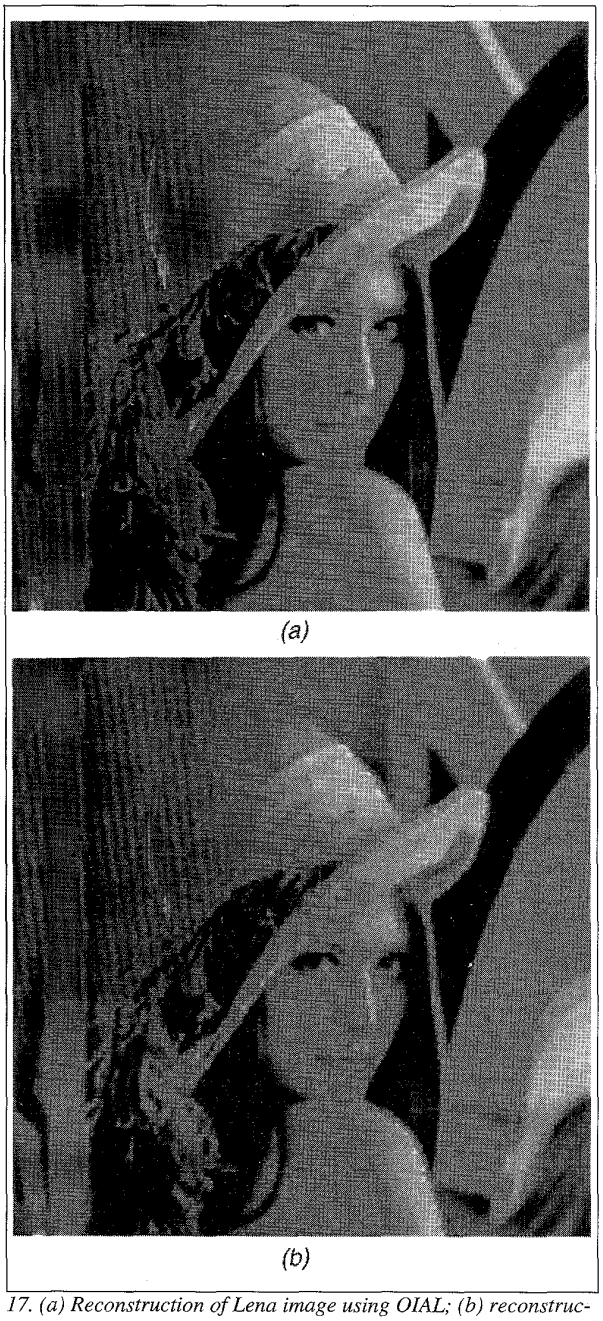
The image compression results just presented are a good indication that the OIAL generalizes within the pertinent class of image. While the "within class condition" may seem restrictive at first, in practice this need not be so. Moreover, while we do not claim that there exists a single network configuration that would perform as well as a general-purpose image compression scheme across a wide variety of images, it is interesting, nevertheless, to see how well a system trained on one class of images generalizes outside that image class. Figure 16 shows the Lena image that is obviously quite different from the image used for training, as shown in Fig. 14a. Figure 17a shows the resulting compressed image using the same network (4 coefficients and 128 classes) and bit rate (0.25 bpp) as that used for the image shown in Fig. 14b. The mean-squared error from this image was 54.9, referred to the original image. For comparison, the image was compressed using the KLT of itself and quantized to the same number of bits (4 coefficients with 0.25 bpp). The resulting image is shown in Fig. 17b, and has a mean-squared error of 71.0, also referred to the original image. These two images clearly show that the OIAL system trained on a magnetic resonance image of a head performs better than the KLT optimized for the specific image being coded.

For many applications, it may be advantageous to have similarity between adjacent classes. The self-organizing feature map (SOFM) introduced by Kohonen [115] makes for such a provision in a simple and yet effective fashion. (The use of the SOFM algorithm as the basis of subspace classifiers is also discussed by Kohonen [126]; the resulting structure is referred to as the "adaptive subspace self-organizing algorithm", which is quite different from the integration of the OIAL and SOFM algorithms.)

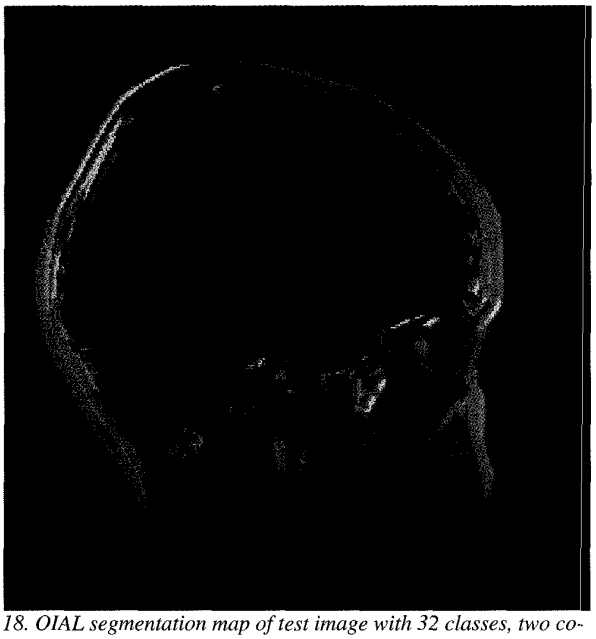
During training, each training vector is used not only to update the winning class, but also classes that are adjacent to it. By integrating the SOFM algorithm into the composition of the OIAL algorithm [123], two useful properties are achieved:

- The OIAL algorithm acquires the ability to perform image segmentation.
- The problem of choosing the initial set of transformation matrices is removed from the algorithm.

The result of this integration is a topological ordering of classes, during training with like classes being close together



17. (a) Reconstruction of Lena image using OIAL; (b) reconstruction of Lenna image using KLT.



18. OIAL segmentation map of test image with 32 classes, two coefficients per class. Color indicates class membership; intensity is weighted by the magnitude of the second coefficient for each block.

in a manner analogous to the ordering of directionally sensitive columns in the visual cortex [127]. This is illustrated in Fig. 18, where each basis block acts as a feature detector. The features corresponding to the basis vectors are either lines or edges of a specific orientation. When comparing adjacent classes, the angles of the features are similar. Moreover, the angles change in a somewhat regular manner as the class number progresses. One other important property resulting from the combined use of OIAL and SOFM algorithms is that the segmentation is independent of variations in illumination, as it is natural in the human visual system [128]. This property may prove to be of significant practical value in image analysis (e.g., computer aided tomography preprocessing, and object/background discrimination).

The OIAL algorithm summarized in Table 5 is one way of implementing the MPC method. In [124], another algorithm, called the multi-class maximum entropy coder (McMEC), is described for implementing the MPC method. The McMEC algorithm uses only one basis vector per module, while the OIAL algorithm has M basis vectors. As a consequence, the McMEC algorithm is required to use a much larger number of modules than the OIAL algorithm.

One of the most demanding application areas of image compression is compressing medical images, where the implications of any sort of distortion are grave indeed. Dony, *et al.*, [129] have investigated the application of the McMEC algorithm to the compression of clinical chest radiographs acquired digitally using the Fuji computed radiography (CR) system. Comparative evaluations with the KLT were also included in the study. Four degrees of compression were used: 10:1, 20:1, 30:1, and 40:1. Seven radiologists evaluated the images. The original and four compressed versions of each image were shown simultaneously to each radiologist, who was asked to rate each one on a five-point scale for image quality and visibility of pathology. Only for the 40:1 versions were there any unacceptable ratings of the McMEC compressed images; even then, these images received a substantial number of top ratings. Many times, the radiologists commented on how little difference there was, if any, between the images. Occasionally, a radiologist would pick the 40:1 compressed image as the best. When compared to the KLT, the McMEC versions ranked better than or as good as the KLT versions 17 times out of 18. In addition, four out of nine times the 40:1 McMEC version was ranked as good or better than the 30:1 KLT version, which is quite remarkable.

Currently, the performance of the MPC approach of image compression is being explored for synthetic aperture radar (SAR) images. In preliminary results, the OIAL approach has reduced the reconstruction error by 3 dB over the KLT for the same compression ratio. This margin of improvement appears to be valid up to compression ratios of 50:1. One explanation for this marked reduction in distortion is the fact that SAR image formation process is highly nonlinear and includes a high degree of "speckle" noise. As a result, the nonlinear nature of the MPC approach matches the signal characteristics better than the linear KLT.

Information-Theoretic Models for Unsupervised Learning

In the previous section we discussed three different signal processing applications of neural networks that require the use of unsupervised learning, exemplified by the nonlinear predictive model in Case Study I, and linear PCA networks in Case Studies II and III. In this section, we discuss another powerful approach to unsupervised learning, which is rooted in information theory. The approach builds on the so-called principle of maximum information preservation, also referred to as Infomax for short, which is due to Linsker [12, 130-133]. It may be stated as follows:

"The transformation of a vector $\mathbf{x}$ in the input layer of a neural network to a vector $\mathbf{y}$ in the output layer of the network should be so chosen that activities of the neurons in the output layer jointly maximize information about the activities in the input layer. The parameter to be maximized is the mutual information between the input vector $\mathbf{x}$ and the output vector $\mathbf{y}$ in the presence of processing noise."

This principle may be viewed as the neural network counterpart to the concept of channel capacity, which defines the Shannon limit on the rate of information transmission through a communication channel.

Becker and Hinton [12, 134-137] have extended the idea of maximizing mutual information to unsupervised processing of the image of a natural scene. Specifically, for a given image, the mutual information between the outputs of two neural network modules is maximized, with adjacent and nonoverlapping patches of the image providing the inputs.

Table 6: Entropy Maximization as a Basis for Unsupervised Learning

Let the vector $\mathbf{x}$ denote a received signal. Let $g(\cdot)$ denote a bounded nonlinear function that operates on $\mathbf{x}$ to produce the output signal vector

$$\mathbf{y} = g(\mathbf{x})$$

The components of both $\mathbf{x}$ and $\mathbf{y}$ are continuous random variables. Let y_i and y_j denote any two components of $\mathbf{y}$. The joint differential entropy of y_i and y_j is defined by [146]:

$$h(y_i,y_j) = h(y_i) + h(y_j) - I(y_i;y_j)$$

where $h(y_i)$ is the differential entropy of y_i and likewise for $h(y_j)$; and $I(y_i;y_j)$ is the mutual information between y_i and y_j. The maximization of $h(y_i,y_j)$ consists of maximizing the individual entropies $h(y_i)$ and $h(y_j)$ while minimizing $I(y_i;y_j)$. When $I(y_i;y_j)$ is zero, the two random variables y_i and y_j are statistically independent. The independent component analysis (ICA) described in [142,147] for blind signal separation and the whitening approach described in [144] for blind deconvolution are examples of minimizing the mutual information $I(y_i;y_j)$ for all pairs of y_i and y_j contained in the output signal vector $\mathbf{y}$. The principle of minimizing $I(y_i;y_j)$ for all y_i and y_j is also known as the principle of redundancy reduction [148].

It can be shown that when the non-linear function $g(\mathbf{x})$ is exactly the cumulative density function (cdf) of the sources, then the goal of maximizing the entropies, $h(y_i)$, of the individual outputs, cannot compete with that of minimizing the mutual information, $I(y_i;y_j)$, between them. In fact, in this case both optimizations have the same solutions. According to Bell and Sejnowski [139,140], even when this condition is relaxed and the function $g(\mathbf{x})$ is other than a pure match for the source distribution, minimum mutual information may still be achieved when the source distributions have a higher kurtosis than the function given by the gradient, $g'(\mathbf{x})$, of the nonlinear function. For many natural (super-Gaussian) signals, this makes the logistic or hyperbolic nonlinear function a useful choice.

In the blind signal separation problem, the components of the input vector $\mathbf{x}$ are represented by N channel outputs. For this problem, Bell and Sejnowski [139,140] start with the nonlinear model:

$$\mathbf{y} = g(\mathbf{x}) = \tanh(\mathbf{W}\mathbf{x} + \mathbf{w}_o)$$

where the bias vector w_o and separating matrix $\mathbf{W}$ are to be computed. Maximization of the joint differential entropy $h(y)$ leads to the following simple rules for the adaptive computation of $\mathbf{W}$ and $\mathbf{w}_o$ [139,140]:

$$\Delta\mathbf{w}_o = -2\eta\mathbf{y}$$

$$\Delta\mathbf{W} = \eta\left(\mathbf{I}\mathbf{W}^{\mathrm{T}}\mathbf{I}^{-1}\right)2\mathbf{y}\mathbf{x}^{\mathrm{T}}$$

where η is the learning-rate parameter assumed to be the same for both rules. The $\Delta\mathbf{w}_o$ rule makes it possible to centre the hyperbolic tangent function on the received signal vector, $\mathbf{x}$. The opposing forces represented by the first term and the second (anti-Hebbian) term in the $\Delta\mathbf{W}$ rule are exactly balanced to achieve the maximum joint entropy for each component of $\mathbf{y}$.

In the blind deconvolution problem, the components of the vector $\mathbf{x}$ are represented by a set of adjacent samples of the received signal. For this problem, Bell and Sejnowski [139,140] start with the nonlinear model

$$\mathbf{y} = g(\mathbf{W}\mathbf{x}) + \mathbf{w}_o$$

where $\mathbf{x}$ and $\mathbf{y}$ are vector representations of the received signal and the output of the blind deconvolution network, respectively. Again using a hyperbolic tangent function for the nonlinear function $g(\cdot)$, Bell and Sejnowski derive the following simple rules for adjusting the weights of a tapped-delay-line representing the deconvolution network:

$$\Delta w_L = \eta\sum_{i=1}^{M}\left(\frac{1}{w_L} - 2x_i y_i\right)$$

$$\Delta w_{L-j} = \eta\sum_{i=j}^{M}\left(-2x_{i-j}y_i\right)$$

where η is the learning-rate parameter, w_L is the leading tap weight, and the w_{L-j} with $j > 0$, are the tap weights linking x_{i-j} to y_i. The rule for changing the bias weight vector $\mathbf{w}_o$ is the same as before.

This latter principle is referred to in the literature as $I_{\max}$. A polarimetric radar application involving navigation along a confined waterway, which builds on a variant of $I_{\max}$, is described in [137-138].

Both of these unsupervised learning procedures, Infomax and $I_{\max}$, rely on the use of noisy models for their operation, which makes their application all the more realistic. Infomax is well suited for the development of unsupervised learning models and feature maps. $I_{\max}$, on the other hand, is well suited for image processing with emphasis on the discovery of properties of a noisy sensory input that exhibit some form of coherence across space and time.

Bell and Sejnowski [139, 140] have built on these information-theoretic models for unsupervised learning by developing their own algorithms to tackle the difficult signal processing problems of blind signal separation and blind deconvolution. The paper by Herault and Jutten [141] is the first neural net paper on blind signal separation using Heb-

bian learning. For a list of references on blind signal separation, see Comon [142]. Note, however, toward the end of 1995, close to 50 papers have been published on this subject since Comon's paper. Also, for a review of traditional signal processing methods used in blind deconvolution, see [143-144]. It appears that the first application of Infomax to the blind deconvolution problem in the context of blind equalization of a communication channel was described in [145].

A typical blind signal separation may be represented by a set of sources corresponding to a number of people engaged in a conversation with music in the background. The signals, $s_1(t)$, $s_2(t)$,...,$s_N(t)$ produced by these different sources are mixed together by an N-by-N matrix, **A**. The sources of these signals and the mixing matrix **A** are all unknown. All that is available for processing is a corresponding set of N received signals $x_1(t)$, $x_2(t)$,...,$x_N(t)$, which are linear superpositions of the original signals $s_1(t)$, $s_2(t)$,...,$s_N(t)$. The problem is to reconstruct these original signals by finding a separating matrix, **W**, that is a permutation and rescaling of the inverse of the unknown matrix, **A**. The problem described herein is sometimes referred to as the "cocktail-party problem."

A similar and equally difficult problem is blind deconvolution, where a source signal $s(t)$ is operated on by a linear filter of impulse response, $h(t)$, to produce a received signal, $x(t)$. The original source signal, $s(t)$, and the impulse response, $h(t)$, are both unknown. All that is given is a statistical model of the source responsible for generating the signal, $s(t)$. Given the received signal, $x(t)$, the problem is to reconstruct the original signal, $s(t)$, with little or no distortion. Areas of application of blind deconvolution include the following [144,145]:

1. Cancellation of reverberation due to the barrel effect encountered in hands-free telephone operation.

2. Seismic deconvolution, where the problem is complicated by a lack of precise knowledge of the short-duration pulse used for excitation.

3. Image restoration, where difficulties arise due to unknown blurring effects caused by photographic and/or electronic imperfections.

4. Blind equalization of a communication channel where it is not feasible to send a training sequence of long enough duration.

Going back to the important contribution by Bell and Sejnowski [139, 140], the essence of their approach may be summarized as follows. In a neural network whose individual neurons are characterized by sigmoidal activation functions, maximization of the information transfer across the network tends to reduce the redundancy between the neurons in the output layer of the network. It is the latter property that enables the network to perform signal separation or deconvolution in an unsupervised manner. It is assumed that the original signal consists of independent symbols. Table 6 presents a summary of the underlying principle in the Bell-Sejnowski procedure for their blind signal separation and blind deconvolution algorithms. In [139-140], experimental results are presented that demonstrate the capability of this new approach for solving blind signal separation and blind deconvolution problems. Although these demonstrations are indeed impressive, the unsupervised learning algorithms developed by Bell and Sejnowski in their present forms are of limited use in certain fundamental respects, as summarized here:

- The neural network models considered are of the single-layer type, with the result that the optimal mappings discovered by the algorithms are constrained to be linear; the use of multilayer models may lead to the development of more powerful input-output mappings.

- For the blind signal separation problem, for N inputs it is assumed that there are an equal number of outputs available for processing. There is no corresponding theory for the more general case when the number of inputs is not equal to the number of outputs.

- In a realistic environment pertaining to the blind signal separation problem, there are unavoidable propagation delays associated with the individual signal paths before they are mixed together. Typically, these propagation delays are unknown. Some adaptive mechanism would therefore have to be incorporated into the blind signal separation algorithm to take account of this practical issue.

- In the blind deconvolution problem, it is assumed that the original signal consists of statistically independent symbols. Although this assumption can be justified in certain situations (e.g., channel equalization), it would be useful to relax the assumption of statistical independence.

These four important research issues, the first three of which are highlighted in [139], certainly deserve more attention.

Perhaps the most important point to emphasize here is that the use of information theoretic models for unsupervised learning, as in the work of Bell and Sejnowski, is a move away from the mean-square error criterion that has permeated so much of the traditional approach to the design of neural networks.

Concluding Remarks

Neural networks, often referred to as an emerging technology, have grown very rapidly on many fronts during the past 10 years. Their theory and design principles have benefited enormously from contributions made by workers in many diverse fields. As such, they represent a significant addition to the "kit of tools" available to system designers. Their ability to learn in a supervised or unsupervised manner, depending on the way in which they are applied, makes them well suited for solving difficult signal processing tasks. In

particular, they can naturally account for the commonly encountered properties of real-life data, including nonlinearity, nonstationatity, and non-Gaussianity.

Perhaps the biggest virtue of neural networks is that they learn about their environment by way of examples, and thereby construct an input-output mapping that brings to mind the notion of nonparametric statistical inference. In so doing, the solutions that they compute may not be guaranteed to be optimum, but they are usually found to be good engineering solutions. Most importantly, from a signal processing perspective, they have the potential (by themselves or in combination with other technologies) to outperform their traditional counterparts, as demonstrated by the three case studies presented in this article.

Acknowledgments

I am truly grateful to Dr. Vladimir Cherkassky, University of Minnesota, Minneapolis, and Dr. William J. Williams, University of Michigan, Ann Arbor, for their many critical inputs and useful suggestions that have impacted the finalizing of this article in a significant way. I also wish to thank Dr. Henry D.I. Aberbanel, University of California, San Diego, Dr. Anthony Bell, Salk Institute, California, Dr. Fay Boudreaux-Bartels, University of Rhode Island, Dr. Bernie Mulgrew, University of Edinburgh, Scotland, and Dr. Kari Torkkola, Motorola, Utah, for their helpful inputs on selected parts of the article. I am indebted to my own colleagues, Dr. Sue Becker, Department of Psychology, Dr. Nanda Kambhatla, Communications Research Laboratory, Dr. James P. Reilly, Department of Electrical and Computer Engineering, and Dr. Patrick Yip, Department of Mathematics and Statistics, my former research colleagues Dr. Tarun Bhattacharya, Raytheon Canada, Dr. Robert D. Dony, Sir Wilfrid Laurier University, and Dr. Graeme Jones, Raytheon Canada, and my current graduate students Hugh Pasika and Paul Yee for reading the entire manuscript and offering numerous suggestions for improving it. I wish to acknowledge the many helpful inputs on chaos that I received on the Internet from Dr. Martin Casdagli, University of Michigan, Ann Arbor, Dr. Daniel Kaplan, McGill University, Quebec, Dr. Mathew Kennel, Oak Ridge National Laboratory, Tennessee, Dr. Lou Pecora, Naval Research Laboratory, Washington, DC, Dr. Florin Takens, University of Gronigen, The Netherlands, Dr. James Theiler, Los Alamos National Laboratory, New Mexico, Dr. Howell Tong, University of Kent, England, and Dr. James Yorke, University of Maryland. Last, but by no means least, I wish to thank Dr. Jose Principe, University of Florida, Gainsville. Dr. Andrew Webb, Defence Research Agency, Great Malvern, England, three anonymous reviewers and Dr. Don Hush, Associate Editor of the *SP Magazine* for their careful reviews of an earlier version of the article.

Simon Haykin is Professor of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada.

References

1. North, D.O., "An analysis of the factors which determine signal-noise discrimination in pulsed carrier systems," *Proc. IEEE*, vol. 51, pp. 1016-1027, 1963.

2. Van Vleck, J.H., and D. Middleton, "A theoretical comparison of visual, aural, and meter reception of pulsed signals in the presence of noise," *J. Appl. Phys.*, vol. 17, pp. 940-971, 1946

3. Wiener, N, *Extrapolation, Intepolation, and Smoothing of Stationary Time Series with Engineering Applications,* Wiley, 1949. (This book was originally issued as a classified National Defense Research Council Report, February, 1942.)

4. McCulloch, W.S., and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin Math. Biophys.*, vol. 5, pp. 115-133, 1943.

5. Rosenblatt, F., "The perceptron: A probabilistic model for information storage and organization in the brain," *Psych. Rev.*, vol. 65, pp. 386-408, 1958.

6. Widrow, B., and M.E. Hoff, Jr., "Adaptive switching circuits," *IRE WESCON Convention Record*, pp. 96-104, 1960.

7. Hopfield, J.J., "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554-2558, 1982.

8. Rumelhart, D.E., and J.L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 1, Cambridge, MA: MIT Press. 1986.

9. Chiu, C.-T., K. Mehrota, C. K. Mohan, and S. Ranka, "Robustness of feedforward neural networks." In P. K. Simpson, editor, *Neural Networks: Theory, Technology, and Applications*, pp. 348-353, 1996, IEEE Press, New York.

10. Haykin, S., *Adaptive Filter Theory*, Third Edition, Prentice-Hall, 1996.

11. Widrow, B., and S. Stearns, *Adaptive Signal Processing*, Prentice-Hall, 1985.

12. Haykin, S., *Neural Networks: A Comprehensive Foundation*, New York: Macmillan College Publishing Company, 1994.

13. Lippmann, R.P., "An introduction to computing with neural nets," *IEEE ASSP Magazine*, vol. 4, pp. 4-22, 1987.

14. Hush, D.R., and B.G. Horne, "Progress in supervised neural networks: What's new since Lippmann?," *IEEE Signal Processing Magazine*, vol. 10, pp. 8-39, 1993.

15. Rao, C.R., *Linear Statistical Inference and its Applications*, Second Edition, Wiley, New York. 1973.

16. Barron, A.R., and R.L. Barron, "Statistical learning networks: A unifying view," In *Proceedings of the 1988 Symposium on the Interface: Statistics and Computing Science* (Ed., E.J. Weaman), pp. 192-203, American Statistical Association, Washington, DC, 1988.

17. White, H., "Learning in artificial neural networks: A statistical perspective," *Neural Computation*, vol. 1, pp. 425-464, 1989.

18. Ripley, B.D., "Neural networks and related methods for classification," *J. Royal Statistical Society Series B*, vol. 56, pp. 409-456, 1995.

19. Ripley, B.D., "Flexible nonlinear methods for classification," *World Congress on Neural Networks*, vol. 2, pp. 927-934, Washington, DC., 1995.

20. Murtagh, F., "Neural networks and related massively parallel methods for statistics: A short review," *International Statistical Review*, vol. 62, pp. 275-288, 1994.

21. Cheng, B., and D.M. Titterington, "Neural Networks: A review from a statistical perspective," *Statistical Science*, vol. 9, pp. 2-54, 1994.

22. Cherkassky, V., J.H. Friedman, and H. Wechsler, editors, *From Statistics to Neural Networks: Theory and Pattern Recognition Applications*, Springer-Verlag, 1994.

23. Helstrom, C.W., *Statistical Theory of Signal Detection*, Second Edition, Pergamon Press, Oxford. 1968.

24. Poor, H.V., *An Introduction into Signal Detection and Estimation*, Springer-Verlag, New York. 1988.

25. Cherkassky, V., D. Gerhing, and F. Mulier, "Pragmatic comparison of statistical and neural network methods for function estimation," *World Congress on Neural Networks*, vol. 2, pp. 917-926, Washington, DC., 1995.

26. Kearns, M., "A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split." In M. C. Mozer, editor, *Advances in Neural Information Processing Systems*, MIT, Press, 1996.

27. Friedman, J.H., and W. Stuetzle, "Projection pursuit regression," *J. Amer. Statist. Soc.*, vol. 76, pp. 817-823, 1981.

28. Friedman, J.H., "Exploratory projection pursuit," *J. Amer. Statist. Soc.*, vol. 82, pp. 249-266, 1987.

29. Wahba, G., "Spline Models for Observational Data," *SIAM*, Philadelphia, 1990.

30. Friedman, J.H., "Multivariate adaptive regression splines (with discussion)," *Annals of Statistics*, vol. 19, pp. 1-141, 1991.

31.Friedman, J.H., "An overview of predictive learning and function approximation". In V. Cherkassky, J.H. Friedman, and H. Wechsler (editors), *From Statistics to Neural Networks*, Springer-Verlag, 1994..

32. Cherkassky, V., Private communication, 1995.

33. Cybenko, G., "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, and Systems,* vol. 2, pp. 303-314, 1989.

34. Hornik, K., M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.

35. Park, J., and I.W. Sandberg, "Universal approximation using radial-basis function networks," *Neural Computation*, vol. 3, pp. 246-257, 1991.

36. Barron, A.R., "Universal approximation bounds for superposition of sigmoid functions," *IEEE Trans. Information Theory*, vol. 39, pp. 930-945, 1993.

37. Kurkova, V., P.C. Kainen, and V. Kreinovich, "Dimension-independent rates of approximation by neural networks and variation with respect to half-spaces". In *World Congress on Neural Networks,* vol. I, pp. 54-57, Washington, DC, 1995.

38. Vapnik, V.N., and A.Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theoretical Probability and its Applications,* vol. 17, pp. 264280, 1971.

39. Vapnik, V.N., *The Nature of Statistical Learning* Theory, Springer-Verlag, 1995.

40. Natrajan, B.K., *Machine Learning: A Theoretical Approach*, Morgan Kaufmann, 1991.

41. Hlawatsch, F., and G.F. Boudreaux-Bartels, "Linear and quadratic time-frequency signal representations," *IEEE Signal Processing Magazine,* vol. 9, pp. 21-61, 1992.

42. Rioul, O., and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8. No. 10, 1991.

43. Brotherton, T., T. Pollard, and D. Jones, "Applications of time-frequency and time-scale representations to fault detection and classification," *Proceedings of the IEEE-SP International Symposium on Time-frequency and Time-scale Analysis,* pp. 95-97, Victoria, BC. 1992.

44. Pati, Y.C., and P.S. Krishnaprasad, "Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations," *IEEE Transactions on Neural Networks,* vol. 4, pp. 73-85, 1993.

45. Manjunath, B.S., and R. Chelleppa, "A unified approach to boundary perception: Edges, textures, and illusory contours," *IEEE Transactions on Neural Networks,* vol. 4, pp. 96-108, 1993.

46. Kreinovich, V., O. Sirisaengtaksin, and S. Cabrera, "Wavelet neural networks are asymptotically optimal approximators for functions of one variable," *Proceedings of IEEE International Conference on Neural Networks,* vol. 1, pp. 299-304, Orlando, Florida, 1994.

47. Szu, H., J. Garcia, and J. DeWitte, "Telemedicine: Recognition from 2D views and 1D projections," *World Congress on Neural Networks,* vol. 2, pp. 828-837, Washington, DC, 1995.

48. Lo, S.-C.B. et al., "Wavelet-based convolution neural network for pattern recognition," *World Congress on Neural Networks,* vol. 2, pp. 838-843, Washington, DC, 1995.

49. Szu, H., B.A. Telfer, J. Anandkumar, and M. Zaghlul, "Remote ECG diagnosis using wavelet transform and artificial neural networks," *World Congress on Neural Networks,* vol. 2, pp. 844-848, Washington, DC, 1995.

50. Qian, W., L. Li, B. Zheng, and L.P. Clarke, "Digital mamography: Wavelet-based mixed feature ANN for automatic detection of microcalcifications," *World Congress on Neural Networks,* vol. 2, pp.849-853, Washington, DC, 1995.

51. Rao, S. S., and R. S. Pappu, "Hierarchical wavelet neural networks." In P. K. Simpson, editor, *Neural Networks: Theory, Technology, and Applications,* pp. 83-90, IEEE Press, 1996.

52. Viterbi, A.J., "Wireless digital communication: A view based on three lessons learned," *IEEE Communications Magazine,* vol. 29, No. 9, pp. 33-36, 1991.

53. Poggio T., and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481-1497, 1980.

54. Bishop, C.M., "Curvature-driven smoothing in backpropagation neural networks," CLM-P880, AEA Technology, Cullham Laboratory, Abingdon, VIC.

55. Sackinger, C., et al., "Application of the ANNA neural network chip to high-speed character recognition," *IEEE Trans. Neural Networks*, vol. 3, pp. 498-505, 1992.

56. Hammerstrom, D., "A VLSI architecture for high-performance, low-cost, on-chip learning," *IJCNN*, vol. 2, pp. 537-544, San Diego, CA., 1990.

57. Soderstrom, T., and B. Svensson, "Using and designing massively parallel computers for artificial neural networks," *J. Parallel and Distributed Computing*, vol.14, pp.260-285, 1992.

58. Hinton, G.E., and T.J. Sejnowski, "Learning and relearning in Boltzmann machines". In *Parallel Distributed Processing* edited by D.E. Rumelhart and J.L. McClelland, MIT Press, 1986.

59. Wejchert, J., and G. Tesauro, "Visualizing processes in neural networks," *IBM J. Res. and Dev.*, vol. 35, pp. 244-253, 1991.

60. Goldstein, H., "Sea echo in propagation of short radio waves." In D.E. Kerr, editor, MIT Radiation Laboratory Series, vol. 13, Section 6.6, McGraw-Hill, New York, 1951.

61. Jakeman, E., and P.N. Pusey, "A model for non-Rayleigh sea echo," *IEEE Transactions on Antennas and Propagation*, vol. AP-24, pp.806-814, 1976.

62. Skolnik, M. editor. *Radar Handbook,* Second Edition, McGraw-Hill, New York, 19...

63. Ward, K.D., C.J. Baker, and S. Watts, "Maritime surveillance radar, Part I: Radar scattering from the ocean surface," *IEE Proceedings,* vol. 137, Pt. F, pp. 51-62, 1990.

64. Leung, H., and S. Haykin, "Is there a radar clutter attractor?" *Applied Physics Letters,* vol. 56, pp. 592-595, 1990.

65. Li, X., and S. Haykin, "Chaotic characterization of sea clutter," *l'Onide Electrique,* Special Issue on Radar, SEE, France, pp. 60-65, March 1994.

66. Haykin; S., and X. Li. "Detection of signals in chaos," *Proceedings of the IEEE,* vol. 83, pp. 95-122. 1995.

67. Schuster, H.G., *Deterministic Chaos: An Introduction,* VCH, Weinheim, Germany, 1980.

68. Ruelle, D., *Chaotic Evolution and Strange Attractors,* Cambridge University Press, 1989.

69. Ott, E., *Chaos in Dynamical Systems,* Cambridge University Press, 1993.

70. Newhouse, S., "Understanding chaotic dynamics". In J. Chandra, editor, *Chaos in Nonlinear Dynamical Systems,* SIAM, 1984.

71. Wolf, A., J.B. Swift, H.L. Swinney, and J.A. Vastano, "Determining Liapunov exponents from a time series," *Physica* 16D, pp.285-317, 1985.

72. Brown, R., P. Bryant, and H. Aberbanal, "Computing the Liapunov exponents of a dynamical system from observed time series," *Physical Review A,* vol.43, pp.2787-2806, 1991.

73. Schouten, J.C., F. Takens, and C.M. van den Bleek, "Maximum-likelihood estimation of the entropy of an attractor," *Physical Review E*, vol.49, pp.126-129, 1994.

74. Farmer, J.D., E. Ott, and J. A. Yorke, "The dimension of chaotic attractors," *Physica D, vol 7,* pp. 153-180, 1983.

75. Grassberger, P., and I. Procaccia, "On the characterization of strange attractors," *Phys. Rev. Letters,* vol. 50, p.346, 1983.

76. Pineda, F.J., and J.C. Sommerer, "A fast algorithm for estimating the generalized dimension and choosing time delays." In *Time Series Prediction: Forecasting the Future and Understanding the Past,* edited by A.S. Weigend and N.A. Gershenfeld, pp.367-385, AddisonWesley, 1994.

77. Schouten, J.C., F. Takens, and C.M. van den Bleek, "Estimation of the dimension of a noisy attractor," *Physical Review E*, vol.50, pp. 1851-1861, 1994.

78. Kaplan, J.L., and J.A. Yorke, in *Functional Differential Equations and Approximation of Fixed Points,* edited by H.-O. Peitgen and H.O. Walther, Springer, Verlag, 1979.

79. Packard, N., J.P. Crutchfield, J.D. Farmer, and R.S. Shaw, "Geogmetry from a time series," *Phys. Rev. Letters,* vol. 45, p.712, 1980.

80. Takens, F., "On the numerical determination of the dimension of an attractor". In D. Rand and L.-S. Yong, editors, *Dynamical Systems and Turbulence, Warwick 1980. Lecture Notes in Mathematics,* vol. 898, pp. 366-381, Springer-Verlag, 1981.

81. Sauer, T., J.A. Yorke, and M. Casdagli, "Embodology," *Journal of Statistical Physics*, vol. 65, pp. 579-616, 1991.

82. Whitney, H., "Differential manifolds," *Ann. Math.*, vol. 37, pp. 645-680, 1936.

83. Fraser, A.M., "Information and entropy in strange attractors," *IEEE Trans. Information Theory,* vol.35, pp.245-262, 1989.

84. Aberbanel, H., and M.B. Kennel, "Local false nearest neighbors and dynamical dimensions from observed chaotic data," *Physical Review E,* vol.47, pp.3057-3068, 1993.

85. Abarbanel, H.D.I., R. Brown, J.J. Sidorowich, and L.S. Tsimring, "The analysis of observed chaotic data in physical systems," *Reviews of Modern Physics,* vol. 65, pp. 1331-1392, 1993.

86. Tong, H., "A personal overview of nonlinear time series analysis from a chaos perspective," *15th Nordic Conference on Mathematical Statistics,* Lund, Sweden, August 1994.

87. Ruelle, D., *Chance and Chaos*, Princeton University Press, p.67, 1991.

88. Li, T., and J.A. Yorke, "Period three implies chaos," *Ann. Math. Monthly,* vol. 82, p.985, 1975.

89. Lorenz, E.N., "Deterministic nonperiodic flow," *J. Atmos. Sci.,* vol. 20, pp. 130-141, 1963.

90. Lorenz, E.N., "On the prevalence of periodicity in simple systems." In *Global Analysis,* edited by Mgrmele and J. Marsden, pp.53-75, Springer-Verlag, 1979.

91. Ruelle, D., and F. Takens, "On the nature of turbulence," *Commun. Math. Phys.*, vol.20, pp.167-192, 1971.

92. Aberbanel, H.D.I., R. Katz, J. Cembrole, T. Galeb, and T. Frison, "Nonlinear analysis of high Reynold number flows over a buoyant asymmetric body," *Phys. Rev. E,* vol.49, pp. 40034018, 1994.

93. Casdagli, M., "Nonlinear prediction of chaotic time series," *Physica,* vol.35D, pp.335-356, 1989.

94. Haykin, S., A. Ukrainec, B. Currie, X. Li and M. Audette, "A neural network-based noncoherent radar processor for a chaotic ocean environment," *ANNIE'95,* St. Louis.

95. Suga, N., "Bisonar and neural computation in bats," *Scientific American,* vol. 262(6), pp. 6068-, 1990.

96. Gabor, D., "Theory of Communication," *J. IEE,* vol.93, pp.429-457, 1946.

97. Haykin, S., *Communication Systems,* Third Edition, Vlley, 1994.

98. Cohen, L., "Time-frequency distributions - A review," *Proceedings of the IEEE,* vol. 77, pp. 941-981, 1989.

99. Cohen, L., *Time-frequency Analysis,* Prentice-Hall, Englewood Cliffs, NJ, 1995.

100. Boashash, B., "Time-frequency signal analysis". In S. Haykin, editor, *Advances in Spectrum Analysis and Array Processing,* vol. I, pp-418-517, Prentice-Hall, Englewood Cliffs, NJ. 1991.

101. Wigner, E.P., "On the quantum correction for thermodynamic equilibrium," *Phys. Rev.,* vol. 40, pp. 749-759, 1932.

102.Ville, J., "Theorie et applications de la notion de signal analytique," *Cables et Transmissions,* vol.2A, pp.61-74, 1948.

103.Cohen, L., "Generalized phase-space distribution functions," *J. Math. Phys.*, vol. 7, pp.781786. 1966.

104. Hlawalsch, F., "Regularity and unitarity of bilinear time-frequency signal representations," *IEEE Transactions on Information Theory,* vol. 38, pp. 82-94, 1992.

105. Hlawatsch, F., "Bilinear time-frequency representations of signals: The shift-scale invariant class," *IEEE Transactions on Signal Processing,* vol. 62, pp. 357-366, 1994.

106. Fukunaga, J., *Statistical Pattern Recognition,* Second Edition, New York: Academic Press, 1990.

107. Flandrin, P., "A time-frequency formulation of optimum detection," *IEEE Trans. Signal Process.*, vol.36, pp.1377-1384, 1988.

108. Williams, W.J., H.P. Zaveri, and J.C. Sackallares, "Time-frequency analysis of electrophysiology signals in epilepsy," *IEEE Engineering in Medicine and Biology*, pp.133-143, March/April 1995.

109. Zaveri, H.P., W.J. Williams, L.D. Iasemedis, and J.C. Sackallares, "Time-frequency representation of electrocorticograms in temporal like epilepsy," *IEEE Trans. Biomedical Engr.*, vol.39, pp. 502-509, 1992.

110. Haykin, S., and T. Bhattacharya, "Wigner-Ville distribution: An important functional block for radar target detection in clutter," in *ASILOMAR Conference on Signals, Systems, and Computers,* Pacific Grove, CA., 1994.

111. Haykin, S., and Bhattacharya, T.K., "Modular learning strategy for signal detection in a nonstationary environment," *IEEE Transactions on Signal Processing,* under review.

112. White, L.B., and Boashash, B., "Time-frequency coherence - a theoretical basis for cross spectral analysis of nonstationary signals," in *Proc. IASTED Int. Symp. Signal Processing and its Applications,* pp.18-23, Brisbane, Australia, 1987.

113. Abeyskera, S.S., and B. Boashash, "Methods of signal classification using the images produced by the Wigner-Ville distribution," *Pattern Recognition Letters,* vol. 12, pp. 717-729, 1991.

114. Hastie, T., and W. Stuetzle, "Principal curves," *J. American Stat. Assoc.*, vol. 84, pp. 502516. 1989.

115. Kohonen, T., "The self-organizing map," *Proceedings of the IEEE,* vol. 78, pp. 1464-1480, 1990.

116. Mulier, F., and V. Cherkassky, "Self-organization as an iterative kernel smoothing process," *Neural Computation,* vol. 7, pp. 1141-1153, 1995.

117. Sanger, T.D., "Optimal unsupervised learning in a single-layer feedforward neural network," *Neural Networks,* vol. 1, pp. 459-473, 1989.

118. Hebb, D.O., *The Organization of Behavior,* Wiley, New York, 1949.

119. Oja, E., "A simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology,* vol. 15, pp. 267-273, 1982.

120. LeCun, Y. et al., "Handwritten digit recognition with a back-propagation network". In D.S. Touretsky, editor, *Advances in Neural Information Processing Systems,* vol. 2, pp. 396-404, Morgan Kaufmann, San Meteo, CA.

121. Stutt, C.A., and L.J. Spafford, "A 'best' mismatched filter response for radar clutter discrimination," *IEEE Transactions on Information Theory,* vol. IT-14, pp. 280-287, 1968.

122. Perrone, M.P., and L.N. Cooper, "Learning from what's been learned: Supervised learning in multi-neural network systems," *World Congress on Neural Networks,* vol. 3, pp. 354-357, Portland. OR. 1993.

123. Dony, R., and S. Haykin, "Optimally Adaptive Transform Coding," *IEEE Trans. Image Processing*, October 1995.

124. Dony R., and S. Haykin, "Multi-class maximum entropy coder," *IEEE International Conf. on Systems, Man, and Cybernetics,* Vancouver, BC, Canada, Oct. 1995.

125. Gray, R.M., "Vector quantization," *IEEE ASSP Magazine*, vol. 1, pp. 4-29, 1984.

126. Kohonen, T., *Self-Organizing Maps,* Springer-Verlag, 1995.

127. Hubel, D.H., and T.N. Wiesel, "Brain mechanisms of vision," *Scientific American,* pp. 130146, September 1979.

128. Cornsweet, T.N., "Visual Perception," Academic Press, New York, 1970.

129. Dony, R.D., S. Haykin, C. Coblenz, and C. Nahmias, "Compression of digital chest radiographs using a mixture of principal components neural networks," *Radiological Society of North America*, Chicago, IL., November 26-December 1, 1995.

130. Linsker, R., "Self organization in a perceptual network," *Computer*, vol. 21, pp. 105-117, 1988.

131. Linsker, R., "An application of the principle of maximum information preservation to linear systems". In D.S. Touretzky, editor, *Advances in Neural Information Processing Systems*, vol. 1, pp. 186-194, Morgan Kaufmann, San Mateo, CA, 1989.

132. Linsker, R., "How to generate ordered maps by maximizing the mutual information between input and output signals," *Neural Computation*, vol. 1, pp. 402-411, 1989.

133. Linsker, R., "Self organization in a perceptual system: How network models and information theory may shed light on neural organization". In S.J. Hanson and C.R. Olson, editors, *Connectionist Modeling and Brain Function: The Developing Interface*, pp. 351-392, MIT Press, Cambridge, MA, 1990.

134. Becker, S., "An Information-theoretic Unsupervised Learning Algorithm for Neural Networks," Ph.D. Thesis, University of Toronto, Ontario, Canada.

135. Becker, S., and G.E. Hinton, "A self-organizing neural network that discovers surfaces in random-dot stereograms," *Nature* (London), vol. 355, pp. 161-163, 1992.

136. Becker, S., and G.E. Hinton, "Learning Mixture Models of Spatial Coherence," *Neural Computation,* vol. 5, pp. 267-277, 1993.

137. Ukrainec, A., and S. Haykin, "A Modular Neural Network for Enhancement of Cross-polar Radar Targets," *Neural Networks*, vol 9, 1996.

138. Ukrainec, A., and S. Haykin, "Mutual Information-based Learning and its Application to Radar," in *Fuzzy Logic and Neural Networks Handbook*, to be published by McGraw-Hill Inc., C.H. Chen, Editor, 1996.

139. Bell, A.J., and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, 1995.

140. Bell, A.J., and T.J. Sejnowski, "Blind separation and blind deconvolution: An information-theoretic approach," *Proc. ICASSP*, vol. 5, pp. 3415-3418, Detroit, Michigan, 1995.

141. Herault, J., and C. Jutten, "Space or time adaptive signal processing by neural network models," In *Neural Networks for Computing* edited by J.S. Denker, AIP Conference Proceedings ISI, American Institute for Physics, 1986.

142. Comon, P., "Independent component analysis, a new concept?," *Signal Processing,* vol. 26, pp. 287-314, 1994.

143. Haykin, S. editor, *Blind Deconvolution,* Englewood Cliffs, NJ: Prentice-Hall, 1994.

144. Duhammel, P., "Tutorial: Blind Equalization," *The 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing,* Detroit, Michigan, 1995.

145. Haykin, S., "Blind equalization formulated as a self-organized learning process," *26th Annual Asilomar Conference on Signals, Systems, and Computers,* vol. 1, pp. 346-350, Pacific Grove, California, 1992.

146. Cover, T.M., and J.A. Thomas, *Elements of Information Theory,* Wiley, New York, 1991.

147. Comon, P., C. Jutten and J. Herault, "Blind separation of sources, part II: problem statement," *Signal Processing,* vol. 24, pp. 11-21, 1992.

148. Barlow H.B., Unsupervised learning." *Neural Computation,* vol. 1, pp. 295-311, 1989.