ELSEVIER

Signal Processing 80 (2000) 1185–1215

SIGNAL PROCESSING

www.elsevier.nl/locate/sigpro

Applications of neural networks to digital communications – a survey

Mohamed Ibnkahla[1]

National Polytechnic Institute of Toulouse, EMSEEIHT, 2, Rue Camichel, 31071 Toulouse Cedex, France

Received 28 November 1997

Abstract

Neural networks (NNs) are able to give solutions to complex problems in digital communications due to their nonlinear processing, parallel distributed architecture, self-organization, capacity of learning and generalization, and efficient hardware implementation. The paper gives an overview of the applications of NNs to digital communications such as channel identification and equalization, coding and decoding, vector quantization, image processing, nonlinear filtering, spread spectrum applications, etc. The key issue in neural network approaches is to find an appropriate architecture that gives the best results. The paper shows, through several examples, how to choose the neural network structures and how to combine neural network algorithms with other techniques such as adaptive signal processing, fuzzy systems and genetic algorithms. Finally, the paper reviews the mathematical approaches used to understand the learning and convergence behavior of neural network algorithms.

Zusammenfassung

Neurale Netzwerke (NNs) können Lösungen für komplexe Probleme der digitalen Übertragungstechnik finden dank ihrer nichtlinearen Verarbeitung, der verteilten parallelen Architektur, Selbstorganisation, der Lern- und Verallgemeinerungsfähigkeiten und durch effiziente Hardwarerealisierungen. Die Arbeit gibt eine Übersicht über Anwendungen von NNs auf Probleme der digitalen Übertragungstechnik wie Kanalidentifikation und -entzerrung, Kodierung und Dekodierung, Vektorquantisierung, Bildverarbeitung, nichtlineare Filterung, Anwendung der Spreadspektrumtechnik usw. Das Hauptproblem bei Anwendungen neuraler Netzwerke ist die Suche einer entsprechenden Architektur, die die besten Ergebnisse liefert. Anhand einiger Beispiele zeigt die Arbeit, wie Strukturen neuraler Netzwerke ausgewählt und wie die Algorithmen mit anderen Methoden wie adaptiven Verfahren, Fuzzysystemen und genetischen Algorithmen kombiniert werden müssen. Am Ende der Arbeit werden mathematische Ansätze besprochen, die für das Verständnis des Lern- und Konvergenzverhaltens der Algorithmen in Neuralen Netzwerken benutzt werden.

Résumé

Les réseaux de neurones sont capables d'apporter des solutions à des problèmes complexes en communications numériques grâce à leur traitement non linéaire, leur architecture parallèlement distribuée, leur auto-organisation, leur capacité d'apprentissage et de généralisation et leur implantation efficace. Cet article dresse un panorama des applications des réseaux de neurones aux communications numériques comme l'identification, l'égalisation, le codage et le

[1]Present address: Department of Electrical and Computer Engineering, Walter Light Hall, Room #408, Queens University, Kingston, Ontario, K7L 3N6, Canada.

E-mail address: mohamed.ibnkahla@ece.queensu.ca (M. Ibnkahla).

0165-1684/00/$ - see front matter
PII: S0165-1684(99)00030-X

Nomenclature

Adaline	adaptive tapped delay line
ANN	artificial neural network
BER	bit error rate
CDMA	code division multiple access
CMA	constant modulus algorithm
DFE	decision feedback equalizer
EM	expectation maximization
em	entropy minimization
HNN	Hopfield neural network
LMS	least mean squares algorithm
LRE	least relative entropy
LTE	linear transversal equalizer
μ	learning rate
MLP	multi-layer perceptron
MLNN	multi-layer neural network
MSE	mean squared error
NN	neural network
OMD	output multi-stage detector
PCA	principal component analysis
PDF	probability distribution function
PSK	phase shift keying
QAM	quadrature amplitude modulation
RBF	radial basis function
RTRL	real-time recurrent learning
SER	symbol error rate
SNR	signal-to-noise ratio
SOM	self-organizing map
SSPA	solid-state power amplifier
S-UMTS	satellite UMTS
TDL	tapped delay line
TWTA	traveling wave tube amplifier
UMTS	universal mobile telecommunications systems
VQ	vector quantization

décodage, la quantification vectorielle, le traitement d'images, le filtrage non linéaire, les techniques d'étalement de spectre, etc. Le point clef pour une utilisation efficace des réseaux de neurones est de trouver une architecture adaptée au problème et qui donne les meilleurs résultats. Ce papier montre, à travers plusieurs exemples, comment choisir les structures neuronales et comment combiner les algorithmes neuronaux avec d'autres techniques comme le traitement adaptatif du signal, les systèmes flous et les algorithmes génétiques. Enfin, le papier décrit les approches mathématiques qui ont été utilisées afin de comprendre le comportement des algorithmes neuronaux pendant l'apprentissage et la convergence.

1. Introduction

Neural networks (NNs) are parallel distributed information processing systems that are capable of learning and self-organizing. They are composed of a large number of simple processing units (called neurons) which are interconnected to form a network which performs complex computational tasks. NNs have several properties which make them very attractive for digital communications applications. These properties are parallel distributed architecture, adaptive processing, self-organization, universal approximation, efficient hardware implementation, etc.

The successful applications of NNs to digital communications come also from their easy integration in large information processing chains. An efficient use of NNs arises in their combination with other techniques such as adaptive signal processing, coding and modulation techniques, fuzzy set theory, genetic algorithms, and evolutionary computation.

The paper reviews the applications of neural networks to digital communications. The paper is organized as follows: Section 2 presents the most popular neural network architectures such as multi-layer perceptrons (MLP), radial basis function (RBF) networks, and self-organizing maps (SOM). The section reviews also the mathematical approaches that are used to understand neurocomputing mechanisms. Sections 3 is devoted to the applications such as channel identification and

equalization, coding and decoding, vector quantization, image processing, etc. Finally the conclusion is given in Section 4.

2. Neural networks: basic concepts

This section describes three popular neural network architectures: Multi-layer perceptrons (MLP), radial basis function (RBF) networks and self-organizing maps (SOM). The section presents as well the recent results derived from information geometry for neural network learning. Other important neural network architectures will not be described in this paper. They can be found in the references, for example, adaptive resonance theory (ART) networks [39,40,72], recurrent neural networks [77,194], principal component analysis (PCA) algorithms [110,117,147], modular neural networks [96,97], Hopfield networks [86].

It is worth noting that several neural network algorithms have been combined with other techniques, such as genetic algorithms and evolutionary computation [19,53,84,109,114,115,154], fuzzy systems [33,41,50,72,102,113], wavelets [172,203], adaptive filtering [21,23,92,104,132], information geometry [11], etc.

2.1. Learning and generalization

Neural networks perform two major functions: learning and generalization [181]. Learning is the process of adapting the connection parameters in a neural network in order to minimize a loss function given an input vector. In general, the learning process is stopped according to a given criterion (e.g. when the obtained weights produce a sufficiently small error performance). However, in some cases (e.g. for time-varying channel equalization), the learning process is performed periodically. Generalization (or recall) is the process of accepting a new input vector and producing an output response. In this case, the (fixed) network weights are used in order to compute the output.

Neural network learning rules indicate how connection parameters are updated when an input learning example is presented to the network. There are two main learning processes:

Supervised learning: In supervised learning (Fig. 1), the network is trained to give the desired response $d(n)$ to a specific input signal $x(n)$. The network has to minimize an error criterion between the desired response and the network output $y(n)$

$$e(n) = d(n) - y(n).$$

Supervised learning is applied for applications such as function approximation, system identification, channel equalization, etc.

Unsupervised learning: In unsupervised learning, the desired response is not available and the response is based on the network ability to self-organize (Fig. 2). Only an input signal $x(n)$ is applied to the network input. The network then organizes itself internally so that each hidden neuron becomes sensitive to a specific set of the input data space. Unsupervised learning is used mainly for applications such as vector quantization, coding and decoding, blind equalization, etc.

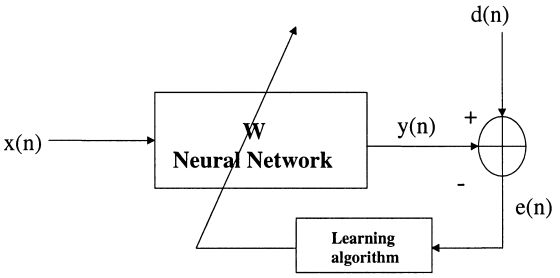

Fig. 1. Supervised learning.

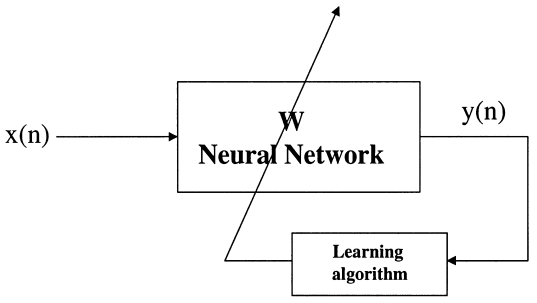

Fig. 2. Unsupervised learning.

2.2. Multi-layer neural networks

2.2.1. Structure

The multi-layer feed forward neural network (MLNN), called also multi-layer perceptron (MLP), is one of the most popular neural network architectures used in digital communications. Its basic unit, the neuron (Fig. 3), is composed of a linear combiner and an activation function. The neuron receives inputs from other processors. The linear combiner output is the weighted sum of the inputs plus a bias term. The activation function gives then the neuron output:

$$x = f(y),$$

where

$$y = \sum_{j=1}^{N} W_j x_j + b,$$

where x_j is the jth input value of the neuron, W_j the corresponding synaptic weight, and b the bias term. $\{W_j\}$ and $\{b\}$ form the free parameters of the neuron.

The activation function may be a linear or non-linear function (e.g. the identity function, a sigmoidal function, the sign function, etc.). The choice of the activation function depends on the application aimed at by the neural network [77].

A multi-layer neural net (see Fig. 4) is composed of neurons connected to each other. The input information is processed from the input layer to the output layer. The network inputs are the inputs of the first layer. The outputs of the neurons in one layer form the inputs to the next layer. The network outputs are the outputs of the output layer. The layer index is denoted by i. x_{ik} is the output of neuron k of layer i. W_{ijk} is the weight that links the output $x_{i-1\,j}$ to neuron k of layer i. $N(i)$ is the number of neurons in layer i. With these notations, the output x_{ik} of neuron (i,k) is given by

$$x_{ik} = f(y_{ik}),$$

where

$$y_{ik} = \sum_{j=1}^{N(i-1)} W_{ijk} x_{i-1\,j} + b_{ik}.$$

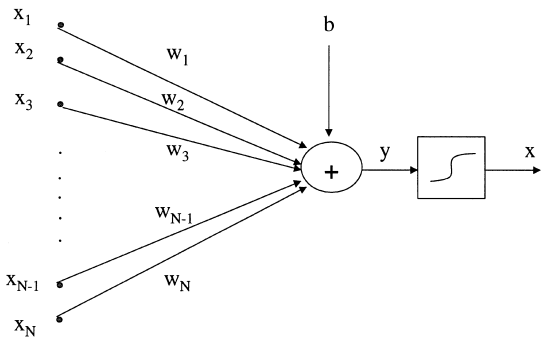

Fig. 3. The neuron.

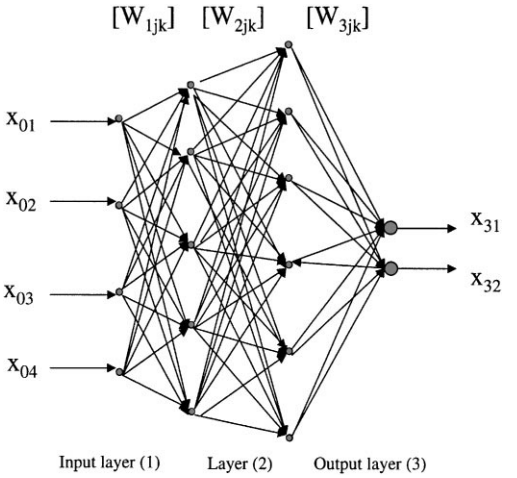

Fig. 4. A multi-layer neural network: The network has 3 layers, 4 input signals, 5 neurons in the first layer, 6 neurons in the second layer, 2 neurons in the output layer, and 2 output signals.

Universal approximation property of MLNN: It has been demonstrated that a two-layer feed forward perceptron with sigmoidal activation function and a scalar output can approximate arbitrarily well continuous functions, provided that an arbitrarily large number of neurons is available (see e.g. [57,87]). This property is called the universal approximation property.

2.2.2. The back-propagation (BP) algorithm

Most neural network applications do not have a priori knowledge of the network correct weights that allow to perform a desired mapping. Thus, a learning (training) procedure is necessary to

obtain these weights. For that purpose, the BP algorithm [126,162] is used. A set of input–output pairs $\{x(n), d(n)\}$ trains the network to implement the desired mapping. The BP algorithm is a supervised learning algorithm. It adjusts the MLNN weights so as to minimize any differentiable cost function, e.g. the squared error energy function (the error power between the network output and the desired output), $E(n) = \|d(n) - x_L(n)\|^2$, where $x_L(n)$ is the MLNN output vector at time n and $d(n)$ is the desired output. Note that other energy functions may also be employed (e.g. $E_p(n) = \|d(n) - x_L(n)\|^{2p}$).

The BP algorithm performs a gradient descent on the energy function in order to reach a minimum:

$$W_{ijk}(n+1) = W_{ijk}(n) - \mu \frac{\partial E(n)}{\partial W_{ijk}(n)}, \tag{1}$$

$$b_{ik}(n+1) = b_{ik}(n) - \mu \frac{\partial E(n)}{\partial b_{ik}(n)}.$$

These equations can be explicitly expressed as

$$W_{ijk}(n+1) = W_{ijk}(n) + \mu \delta_{ik} x_{i-1\,j},$$

$$b_{ik}(n+1) = b_{ik}(n) + \mu \delta_{ik}. \tag{2}$$

The so-called error term δ_{Lk} of the output layer is given by:

$$\delta_{Lk} = 2f'(y_{Lk})(d_k - x_{Lk}),$$

where f' denotes the derivative of the activation function $f'(x) = \mathrm{d}f(x)/\mathrm{d}x$.

The calculation of the error term δ_{ik} of the hidden unit (i,k), can be easily expressed as a function of the next layer error terms:

$$\delta_{ik} = f'(y_{ik}) \sum_{j=1}^{N(i+1)} W_{i+1\,kj} \delta_{i+1\,j}.$$

Thus, the weight update is performed by propagating the error terms from the output layer to the input layer.

2.2.3. The complex valued BP algorithm

In digital communications and signal processing, signals are often complex valued and need to be processed in complex vector spaces. Complex-valued neural networks (CNN) have been proposed in order to represent and process complex-valued signals.

Some authors have generalized the MLNN and the BP algorithm to the complex plane [52,124,173]. For example, Leung and Haykin [124] have proposed a CNN which is similar to the real-valued MLNN. Its weights are complex valued and the activation function is an extension of logistic function to the complex plane: $f(z) = 1/(1 + \exp(-z))$. This activation function is not bounded everywhere. In order to avoid this problem of singularity, Leung and Haykin proposed to scale the input data to some region in the complex plane. The network output is calculated the same way as in the real-valued case (see above), except that the variables are replaced here by complex-valued variables.

The complex BP (CBP) algorithm updates the weights as

$$W_{ijk}(n+1) = W_{ijk}(n) + \mu \delta_{ik} x^*_{i-1\,j}, \tag{3}$$

where $*$ denotes the complex conjugate, δ_{ik} is the error term which is computed by error back propagation.

For the output layer, the output error term is given by

$$\delta_{Lk} = 2f'(y^*_{Lk})(d_k - x_{Lk}),$$

where

$$y_{Lk} = \sum_{j=1}^{N(L-1)} W_{Ljk} x_{L-1\,j} + b_{Lk}.$$

For the hidden layers, δ_{ik} is computed by

$$\delta_{ik} = f'\left(\left(\sum_{j=1}^{N(i-1)} W_{ijk} x_{i-1\,j} + b_{ik}\right)^*\right) \sum_{j=1}^{N(i+1)} W^*_{i+1\,kj} \delta_{i+1\,j}.$$

2.3. RBF networks

2.3.1. Structure

The RBF network (see e.g. [77]) is a two-layer feedforward network (Fig. 5), whose input-layer activation function is

$$f(x) = \phi(\|x\|),$$

where ϕ is a continuous function from R^+ to R (a radial basis function). In contrast to MLNN, the

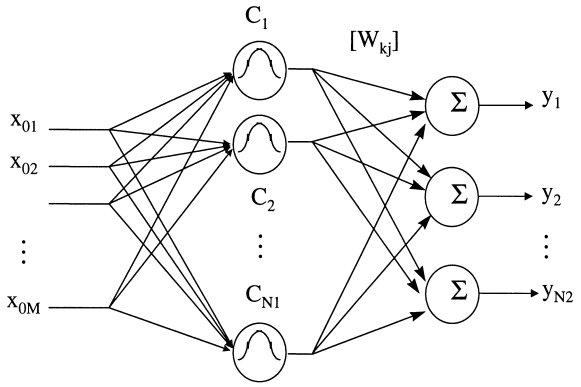

Fig. 5. A radial basis function (RBF) network (the input signal is M-dimensional, the first layer has N_1 neurons, the output layer has N_2 neurons).

activation function is a compact one, and no longuer a sigmoid. It may be, e.g. Gaussian shaped:

$$\phi(t) = \exp\left(-\frac{t^2}{2\sigma^2}\right).$$

The outputs of the first layer neurons are written as

$$x_{1k} = \phi_k(\|x - c_k\|),$$

where x is the input vector and c_k is the weight (or center) associated to neuron k. The network outputs are written as

$$y_j = \sum_{k=1}^{N_1} w_{kj}\phi_k(\|x - c_k\|),$$

where N_1 is the number of neurons in the first layer, and w_{kj}, $j = 1, \ldots, N_2$, are the weights associated with the output layer. The free parameters are therefore the centers $\{c_k\}$ and the weights $\{w_{kj}\}$.

Universal approximation property of RBF networks: It has been demonstrated that an RBF network can approximate continuous functions arbitrarily well, provided that an arbitrarily large number of neurons is available. Moreover, RBF networks have the property of *best approximation*. See [70,153] for an in-depth study of RBF (and other neural network architectures) approximation properties.

2.3.2. Learning rule

Several algorithms have been proposed to update RBF networks. The most widely used algorithm is composed of an unsupervised learning rule for centers update, and a supervised learning rule for the output weights update.

Centers update: The centers are updated according to the following steps:

1. Present a signal $x(n)$ to the network.
2. Compute the distances between the input vector and the centers:

   $$D_i(n) = \|x(n) - c_i(n)\|.$$
3. Determine the closest center to the input signal:

   $$I \text{ such that } D_I(n) = \min_i D_i(n).$$
4. Update the center c_I according to:

   $$c_I(n+1) = c_I(n) + \mu(x(n) - c_I(n)).$$

   where μ is a small positive constant.
5. Back to step 1.

Supervised learning for the weights (*output layer*): The LMS algorithm is used to update the output layer weights:

1. Present a pair of input-desired output signals $(x(n), d(n))$.
2. Compute the error between the network output and the desired output:

   $$e_j(n) = d_j(n) - y_j$$
3. Update the weights of the output layer according to:

   $$w_{kj}(n+1) = w_{kj}(n) + \alpha e_j(n)\phi_k(\|x - c_k\|),$$

   where α is a small positive constant.
4. *Back to step 1.*

2.4. Self-organizing maps

Let E be a p-dimensional vector space and w an element of E. A neuron defined in E is characterized by its weight vector w. A self-organizing map (SOM) defined on E is a grid A (which can be one, two, or n dimensional, with $n \in N$) of neurons w_k defined in E [110,111,157]. A two-dimensional SOM is represented in Fig. 6.

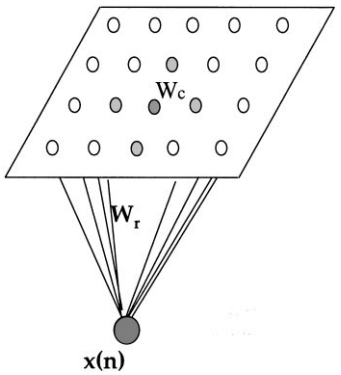

Fig. 6. A 2-Dimensional self-organizing map (a winning neuron c is represented with 4 neighbors).

In the following we present learning and generalization processes of self-organizing maps (see e.g. [110,111,157]).

Competitive learning rule: Let x be the input to the SOM at time n. The adaptation of a neuron w_k at time t is determined by computing the distance between x and w_k:

$$d(x, w_k) = \|x - w_k\|.$$

The basic learning rule for the SOM map (called also Kohonen learning rule) is performed according to the following steps:

0. Initialize the SOM: Neurons w_i are initialized with small random values.
1. Present an input signal $x(n)$.
2. For each neuron w_k of the map, compute the corresponding distance to the input signal (or adaptation) $d(x, w_k)$. Determine the so-called excitation center w_c (i.e. the neuron of grid A that has the minimal adaptation):
$$d(x, w_c) = \min_{k \in A} d(x, w_k).$$
3. Update the weights:
$$w_k^{\text{new}} = w_k^{\text{old}} + \lambda h_{kc}(x(n) - w_k^{\text{old}}). \quad (4)$$
4. Back to step 1 (present a new input sample $x(n+1)$) until an equilibrium of the map is reached (to within a given precision).

h_{kc} is the excitation response which defines the response of neuron k when c is the center. For example, h_{kc} may be a function which decreases as the distance $|k - c|$ increases (the decreasing rate of this function determines the neighborhood of the winning neuron).

Generalization (*vector quantization*): Generalization means the computation of the SOM output after the learning phase is completed. A new input signal x is then presented to the map. The corresponding output w (wich is called the excited neuron) is detemined according to the following mapping F:

$$F: \quad E \to A,$$
$$x \to w \text{ such that } d(x, w) = \min_{r \in A} d(x, w_r).$$

The mapping F performs then a vector quantization [71,110] of the input space into the map A.

2.5. Natural gradient learning

The preceding algorithms are based on the stochastic gradient method [8,162,192]. However, when the parameter space is not Euclidiean but has a Riemannian metric structure, the ordinary gradient does not give the steepest direction of the cost function. In such a case, the steepest descent is given by the natural gradient [7,10,11,15]. Amari [11] expresses the natural gradients explicitly in the case of the space of perceptrons for neural learning, the space of matrices for blind source separation, and the space of linear systems for blind source deconvolution. It was proved that the natural gradient on-line learning gives the Fisher efficient estimator when the cost function is differentiable, so that it is asymptotically equivalent to the optimal batch procedure.

2.5.1. Natural gradient [11]

Let $S = \{w\}$ be a parameter space on which a cost function $L(w)$ is defined. When S is a Euclidean space with an orthonormal coordinate system w, the squared length of a small incremental vector $\mathrm{d}w$ is given by

$$\|\mathrm{d}w\|^2 = \sum_{i=1}^{N} (\mathrm{d}w_i)^2,$$

where $\mathrm{d}w_i$ are the components of $\mathrm{d}w$. However, when the coordinate system is nonorthonormal or the space S is Riemannian, the squared length is

given by the quadratic form

$$\|\mathrm{d}w\|^2 = \sum_{i,j} g_{ij}(w)\,\mathrm{d}w_i\,\mathrm{d}w_j.$$

The matrix $G = (g_{ij})$ is called the Riemannian metric tensor and depends in general on w, it reduces to the unit matrix I in the Euclidean orthonormal case.

The steepest descent direction of a function $L(w)$ at w is defined by the vector $d(w)$ that minimizes $L(w + \mathrm{d}w)$ under the constraint $\|\mathrm{d}w\|^2 = \varepsilon^2$ for a sufficiently small constant ε.

It was demonstrated [11] that the steepest descent direction of $L(w)$ in a Riemannian space is given by

$$-\tilde{\Delta}L(w) = -G^{-1}(w)\Delta L(w),$$

where G^{-1} is the inverse of G, and $\Delta L(w)$ is the conventional gradient

$$\Delta L(w) = \left(\frac{\partial}{\partial w_1}L(w), \ldots, \frac{\partial}{\partial w_N}L(w)\right).$$

The natural gradient of L in the Riemannian space is defined by

$$\tilde{\Delta}L(w) = G^{-1}\Delta L(w).$$

When the space is Euclidean and the coordinate system is orthonormal, then

$$\tilde{\Delta}L(w) = \Delta L(w).$$

This suggests the gradient descent algorithm of the form

$$w(n+1) = w(n) - \mu\tilde{\Delta}L(w),$$

where μ is the algorithm learning rate.

2.5.2. Natural gradient learning [11]

We present here two important examples of learning algorithms that have been introduced in [11]. Let us consider an information source which generates a sequence of independent random variables $x(1), x(2), \ldots, x(n), \ldots$, subject to the same i.i.d. probability distribution $q(x)$ and processed by a processing tool, e.g. a filter, a neural network, etc., which is parametrized by a weight vector w. Let $l(x, w)$ be the loss function to be minimized. Then the cost function is defined by

$$L(w) = E(l(x, w)),$$

where E denotes the expectation with respect to x. The stochastic descent learning method is given by

$$w(n+1) = w(n) - \mu C(w(n))\Delta l(x(n), w(n)), \tag{5}$$

where C is a positive-definite matrix [11]. The natural gradient on-line method takes $C(w)$ equal to $G^{-1}(w)$ when the Riemannian structure is defined.

Statistical estimation: We assume a statistical model $p(x, w)$. The problem here is to obtain the probability distribution $p(x, \hat{w})$ which approximates the unknown density function $q(x)$. A typical loss function is

$$l(x, w) = -\log p(x, w).$$

The expected loss is then given by

$$L(w) = -E(\log p(x, w)) = E_q\left[\log\frac{q(x)}{p(x, w)}\right] + H_X,$$

where H_X is the entropy of $q(x)$ that does not depend on w. Thus, minimizing L is equivalent to minimizing the function

$$D(q(x)/p(x, w)) = \int q(x)\log\frac{q(x)}{p(x, w)}\,\mathrm{d}x.$$

Note that when $q(x)$ is written as $q(x) = p(x, w^*)$, this is equivalent to obtaining the maximum likelihood estimator $\hat{w}$. The Riemannian structure of the parameter space of a statistical model is defined by the Fisher information

$$g_{ij}(w) = E\left[\frac{\partial \log p(x, w)}{\partial w_i}\frac{\partial \log p(x, w)}{\partial w_j}\right].$$

The updating rule is obtained using Eq. (5).

Multi-layer neural network: As was discussed above, the output of a multi-layer neural network can be expressed as $f(x, w)$, where x is the input signal and w is a vector that represents the network weights. Let $q(x)$ be the unknown probability distribution of x. Let us consider a teacher network which, by receiving x, generates the corresponding output y subject to a conditional probability distribution $q(y/x)$. The task of the student network is to

obtain the optimal w^* which approximates the behavior of the teacher. A typical loss function $l(x, w)$ is given by

$$l(x, y, w) = \tfrac{1}{2}\|y - f(x, w)\|^2,$$

where y is the output given by the teacher. We consider a statistical model of neural networks such that its output y is given by a noisy version of $f(x, w)$,

$$y = f(x, w) + v,$$

where v is multi-variate Gaussian noise with zero mean and unit covariance matrix. The probability density of the input–output pair $z = (x, y)$ is then

$$p(z, w) = c \times q(x)\exp(-\tfrac{1}{2}\|y - f(x, w)\|^2),$$

where c is a constant, and the loss function is rewritten as

$$l(z, w) = \text{const.} + \log(q(x)) - \log(p(z, w)).$$

Thus the natural gradient on-line learning algorithm is written as

$$w(n + 1) = w(n) - \mu\tilde{\Delta}l(x(n), y(n), w(n)).$$

The Riemannian stucture is naturally given to the parameter space of multi-layer networks by the Fisher information matrix:

$$g_{ij}(w) = E\left[\frac{\partial \log p(x, y, w)}{\partial w_i}\frac{\partial \log p(x, y, w)}{\partial w_j}\right].$$

The calculation of G as well as its inversion is given in [11]. Yang and Amari [198] propose an algorithm which avoids to inverse the Fisher information matrix.

Amari [11] studied also the natural gradient in the space of matrices and blind source separation, and in the space of linear systems with memory and blind source deconvolution.

2.6. Understanding neurocomputing mechanisms

For an engineer, it is necessary to develop 'mathematical techniques to evaluate neural network's performance with real-world data' [121]. In the digital communications domain, if new algorithms are to be used in real communication systems, it is important that the algorithm designer understands their learning behavior and performance capabilities. The convergence properties and performance evaluation of neural network algorithms have been a large research area, and since the 1960s, several approaches have been proposed to understand the mechanisms of neural networks. This section reviews the research results on this topic.

Amari gives in [8] a survey on the mathematical approaches that are used to understand neurocomputing dynamics. His study included back-propagation, associative memory, and self-organization. The paper of White [189] reviews concepts and analytical results, based on statistical models, relevant to the analysis of learning procedures in feed forward neural networks. A collection of several papers on the mathematical approaches to neurocomputing can be found in [178].

Heskes and Kappen [83] presented a general study of on-line learning processes in artificial neural networks. They addressed several problems such as learning processes and their average behavior, learning in a changing environment, learning parameter adjustment, transition times between local minima, unfolding twists in a self-organizing map, and on-line learning and global optimization.

Grossberg [72] summarizes several contributions to understand how neural models function autonomously in a stable fashion despite unexpected changes in their environments. The content of these models consists of a small set of equations that describes processes such as activation of short-term memory traces and associative learning adaptive weights or long-term memory traces; a larger set of modules that organize processes such as cooperation, competition, pattern learning; and a still larger set of neural networks or architectures for achieving general-purpose solutions of modal problems such as vision and speech.

In his 1967 paper [6] Amari studied the speed and accuracy of learning in the general stochastic descent dynamics. A Bayesian statistical approach has been used in [14] to obtain four types of universal results on learning under the annealed approximation. A universal property of learning curves for deterministic dichotomy neural network classifiers was derived in [9].

A mechanical–statistical approach has been used by Levin et al. [125] to predict learning and

generalization of neural networks. The authors made a link between the mechanisms of a neural network and statistical information theory in the case of a classification problem. A review of information theoretic approaches to neural networks has been presented in [179].

The convergence properties of self-organizing maps have been studied in [157,158] using a Markovian model. A similar approach was used by Kohonen [110] for the one-dimensional Kohonen map. See also the work of Cottrel and Fort [55].

The influence of the number of layers on the performance and convergence behavior of the back-propagation algorithm was studied in [90]. The study was derived in the context of linear adaptive filtering [132]. The paper also shows the effects of the algorithm step size and the initial weight values upon the algorithm behavior and performance. A simple, but very interesting structure consisting of two inputs and one neuron was studied by Shynk and Roy [169]. They studied in particular the influence of the learning and momentum rates on the algorithm behavior near convergence. A statistical approach based on Price's theorem allowed Bershad et al. to analyze the mean weight behavior and the steady-state MSE of a single layer perceptron in a nonlinear system identification model [31,32]. A simplified two-layer neural network which models the amplitude conversion of a TWT amplifier was analyzed in [29,30]. The study shows the effects of the neural network structure, weights, initial conditions and algorithm step size upon the MSE of the neural network approximation. The performance analysis was based upon the derivation of recursions for the mean weight update which can be used to predict the weights and the MSE during learning. [92] presents an analysis which constitutes a theoretical basis for the usefulness of neural networks for modeling satellite channels and nonlinear amplifiers.

The above approaches study the behavior of a neural network by mathematical analysis or by computer simulation. However, it is important to study not only a specific network but also a whole family of networks to analyze the capabilities and limitations of the family of networks with the same architecture. When neural networks with a specific architecture are parametrized with N parameters (such as connection weights) summarized in a vector $\boldsymbol{\theta}$, then the set of such networks form an N-dimensional manifold. This is called the manifold of neural networks [15]. Geometry of a neural manifold is useful for understanding the total capability of a class of networks. When a network is of a stochastic nature, each network is accompanied with a probability distribution $p(x;\boldsymbol{\theta})$ or a conditional probability distribution $p(y/x;\boldsymbol{\theta})$ (x is the input, y is the output). Information geometry [10] connects these two sources of ideas. It originated from the information structure of a manifold of probability distributions and has been developed to be a new mathematical framework with new differential geometry notions. Amari [10] established a unified framework for statistical and geometrical ideas for studying neural networks including hidden units and unobservable variables. In particular, the relation between the statistical *EM* algorithm [97] (which is an iterative statistical technique of using the conditional expectation) and the geometrical *em* algorithm (which is based on information geometry) was elucidated. Several examples are given in [10] including stochastic multi-layer perceptron, mixtures of experts, and normal mixture model [96].

Information geometry has been applied to Boltzmann machines [15]. The set of all the Boltzmann machines with a fixed topology forms a geometrical manifold of high dimension, where modifiable synaptic weights of connections play the role of a coordinate system. [15] established a natural invariant Riemannian metric and a dual pair of affine connections on the Boltzmann neural network manifold. The paper addressed the following approximation problem: given an information source, find the Boltzmann machine that realizes, as faithfully as possible, the given probability distribution as its stationary distribution. When a parameter space has a certain underlying structure, the ordinary gradient of a function does not represent the steepest direction but the natural gradient does; information geometry has been used [11] for calculating the natural gradients in the parameter space of perceptrons, the space of matrices (for blind source separation) and the space of linear systems (for blind source deconvolution) [11]. This allowed

to analyze the dynamical behavior of natural gradient on-line learning which was proved to be Fisher efficient, implying that it has asymptotically the same performance as the optimal batch estimation of parameters. This suggests that the plateau phenomenon which appears in the BP algorithms disappears when the natural gradient is used [11]. Information geometry has also been applied in [12,13] for the analysis of learning algorithms for blind source separation (see [38,82,129,138]). [13] analyzed a general form of statistically efficient algorithms for blind source separation. It gives a necessary and sufficient condition for the separating solution to be a stable equilibrium of a general learning algorithm. The efficiency of the algorithms has been studied in [12].

3. Applications of neural networks to digital communications

3.1. Nonlinear channel modeling and identification

Channel modeling and identification is a very important task in digital communications (see e.g. [25]). It can be used for several purposes, such as receiver design, computer simulation and performance evaluation of communication channels, diagnosis, fault detection, etc. This section is devoted to neural network approaches to channel identification.

Neural networks have been widely used as powerful tools for modeling nonlinear dynamical systems (e.g. [143,144]). They provide an alternative to existing nonlinear identification techniques such as Volterra series [25,133] and nonlinear auto-regressive moving average (NARMA) models [34]. [98,172] give a comparative review of system identification approaches including Volterra series, wavelet networks, and neural networks.

The application of neural networks to system modeling and identification is motivated by the universal approximation property of neural networks. Indeed, several feed forward neural network architectures have been shown to be universal approximators, i.e. they can approximate arbitrarily well any continuous real-valued function on a bounded subset of R^n (see e.g. [57,70,87,153]). These results deal with memoryless (or static) mappings, i.e. the system output at time t depends only on its input at the same time.

Recently, multi-layer sigmoidal neural networks have been applied to modeling nonlinear memoryless channels such as traveling wave tube (TWT) amplifiers [94]. TWT amplifiers [25,164] are nonlinear devices that exhibit a nonlinear phase conversion and a nonlinear amplitude conversion. Neural network models have been shown to outperform classical TWT analytical models [28,164,180] (Fig. 7).

The advantages of TWT neural network models are

(1) they allow to approximate complicated nonlinearities which could not be well modeled by classical models,

(2) in addition to their small number of parameters (about 15 weights), they can be implemented in parallel, which reduces the computation time,

(3) they are adaptive, therefore a change in the TWT characteristics can be tracked by the neural net model, and

(4) they ensure good mathematical properties (e.g. asymptotic behavior and continuity).

Moreover, neural networks allow efficient approximation of the inverse TWT transfer function [91]. This makes them good alternatives to classical predistortions techniques such as polynomial [135], Volterra [100,101,122], and adaptive analog approaches [95].

Most communication channels are *dynamic* systems (i.e. with memory). Even though the universal approximation property does not apply for approximating dynamic systems, several authors have applied neural networks for dynamic systems modeling, see [89] which presents a survey of the applications of neural networks to dynamic modeling.

Most of the work on dynamic system identification has been empirical in nature, and the choice of the neural network identification structure depended closely on the specificities of the application.

For example, Narenda et al. [143,144] have used several MLP structures (trained with static and dynamic back-propagation methods) and nonlinear autoregressive moving average (NARMA) representations for identification and control of

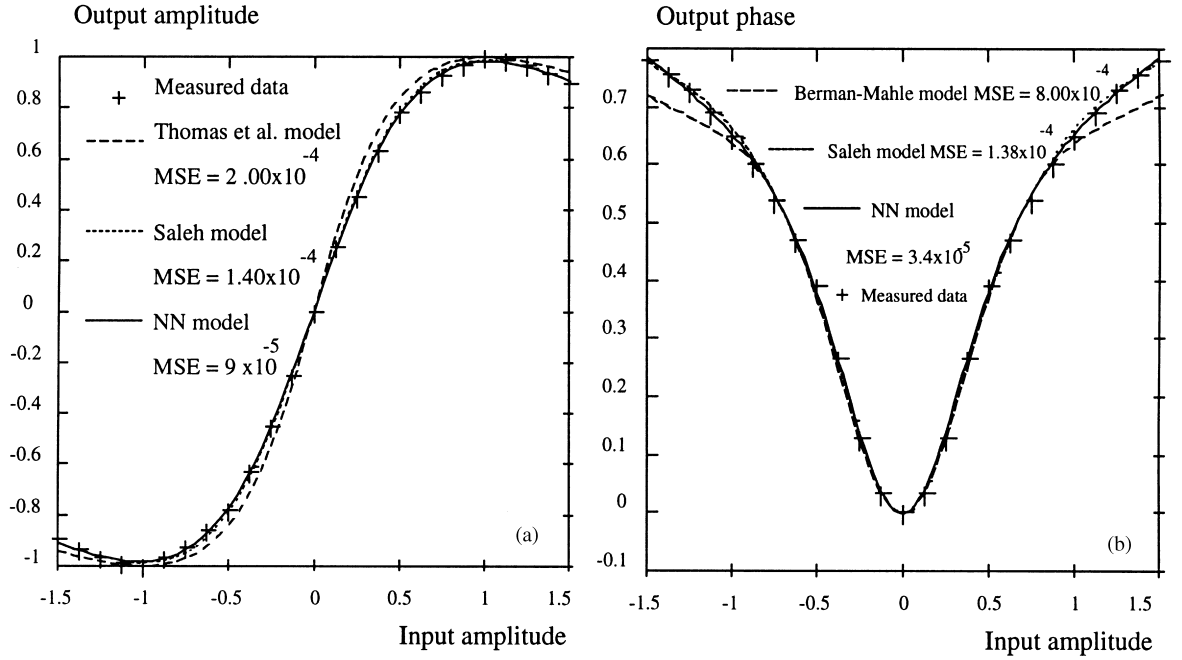

Fig. 7. Neural network modeling of the AM/AM (a) and the AM/PM (b) conversions of a TWT amplifier, and comparisons with classical TWT models.

dynamical systems with applications to robotics, autonomous navigation, aeronautics, etc.

Many neural network structures have been proposed in the literature for time-series modeling. These models include recurrent networks [54] and FIR networks [201]. The basic concepts of back-propagation through time are presented in [188]. See also other applications in [23,190].

Solid-state power amplifiers (SSPA) are examples of nonlinear dynamic systems that are used in digital communications. They are nonlinear amplifiers with memory. An SSPA is composed of adaptation circuits and a transistor. For example, the metal–semiconductor field effect transistor (MESFET) has an equivalent circuit model which involves several nonlinear phenomena (e.g. sources, capacitors [61,205]). A complex-valued neural network was used [92] to model a 2.18 − GHz SSPA. The model consisted of a linear complex filter followed by a memoryless nonlinear neural net followed by a linear complex filter. The approach allowed to successfully model the SSPA transfer function. Fig. 8 shows the generalization performance of the model when the SSPA operates at saturation. The generalization error is $\mathrm{MSE} = 2.96 \times 10^{-4}$, the corresponding signal-to-error ratio is 36.11 dB.

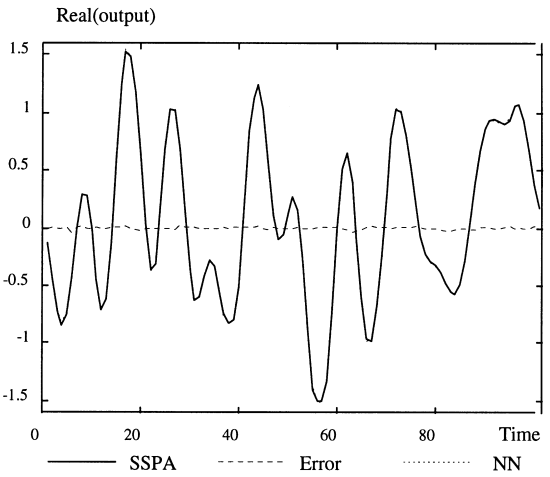

Fig. 8. Neural network modeling of SSPA amplifier.

Ibnkahla et al. [92] used a neural network approach for satellite channels identification (Fig. 9). This approach is alternative to classical identification techniques (such as Volterra series approaches, see e.g. [24–26]). Two structurally different neural network models were applied: a black box model and a block-oriented model (the learning process being performed by using the channel input–output signals). The first network is a three-layer perceptron which is a black box model because it yields a global modeling of the channel and cannot characterize each component separately. The block-oriented NN structure (Fig. 10) is similar to the channel structure. The satellite channel is composed of an uplink linear filter, a nonlinear memoryless amplifier and a downlink linear filter. The net is composed of a linear tapped delay line (TDL)

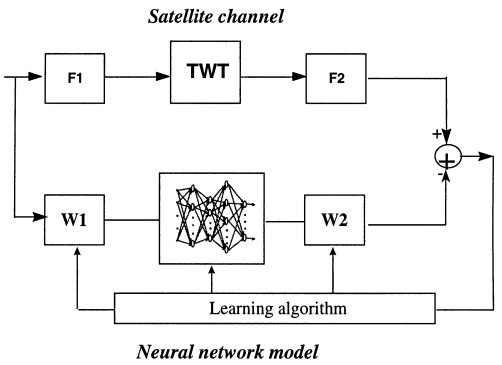

Fig. 9. Satellite channel identification structure.

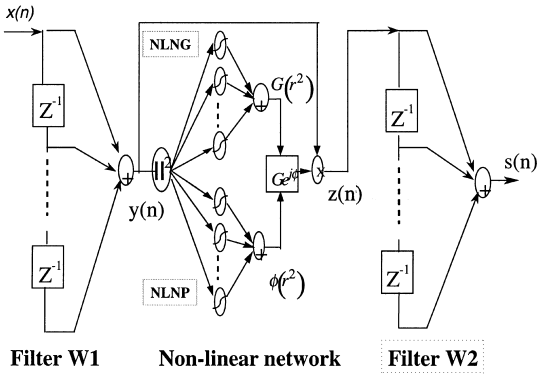

Fig. 10. Neural network structure (block oriented model) used for the identification of the satellite channel.

filter, a nonlinear network, and a second TDL filter. It was shown that the block-oriented structure outperforms the black box one (see Figs. 12 and 11). Moreover, the block-oriented structure allows to characterize each component of the channel. The structure was also applied to fault detection in digital satellite transmissions [92]. Ibnkahla et al. [93] introduced a neural network approach for modeling and characterization of nonlinear time-varying fading channels. The model was composed of three parts: a linear adaptive filter, a memoryless nonlinear neural network, and a time-varying filter. The NN approach allowed to characterize the channel (time-invariant) linear filters, the channel nonlinearity (i.e. the TWT amplifier), and the multi-path propagation channel (i.e. the number of reflected paths and their characteristics). Another approach based on polynomial networks has been proposed by Ghogho et al. [69]. The idea was to model the memoryless nonlinearity by a finite-order polynomial, and to show that the time-varying nonlinear system is a time-varying Volterra system. A least-squares (LS) estimate of the time-varying Volterra kernels was then derived.

The problem of channel characteristics changes caused by external factors has been addressed in [160]. The authors proposed an adaptive structure to compensate for the deviations that may occur in the communication channel. The proposed approach was to model the system deviation, train a simple feed forward neural network using this information, and then using a specific parametrization form an appropriate adaptive filter for system correction. See also [130,151] for diagnosis and fault detection in other engineering applications.

In all of these applications to dynamic system modeling and identification, a theory is still to be established to justify such applications. In particular, the following issues are needed [137]:

(i) define a class of operators (see e.g. [137,155]) analogous to multi-layer neural networks,

(ii) determine the approximation capability of the proposed class of operators, and

(iii) provide a realization theory for implementing the proposed set of operators.

A considerable work has been done in the recent years addressing these issues. For example, Modha and Hecht-Nielsen [137] introduced multilayer

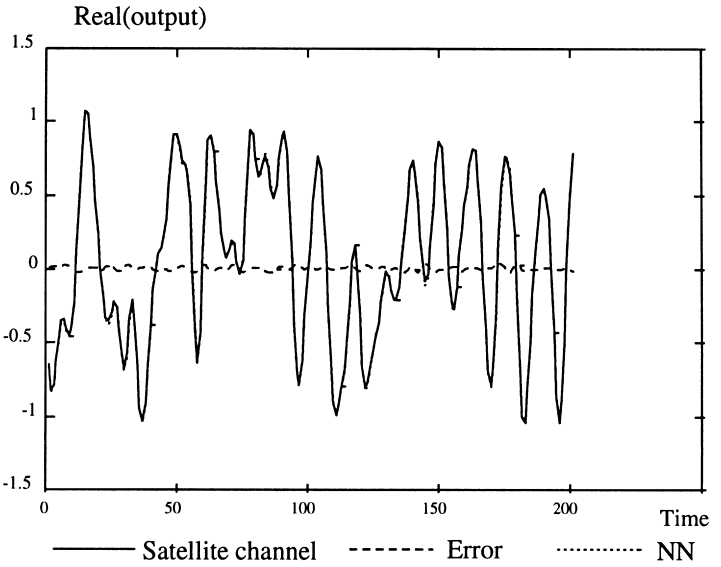

Fig. 11. Neural network modeling of a satellite channel: Comparison between the channel output and the model output for the same input (real part). The filter-nonlinearity-filter structure has been used.

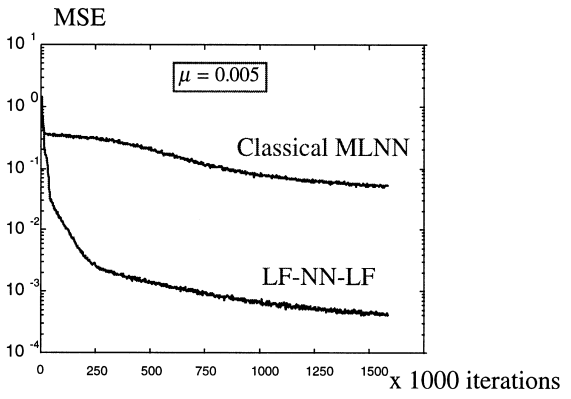

Fig. 12. NN Modeling of a satellite channel: Learning curves of a black-box model (a 3-layer perceptron) and a block-oriented model (a linear filter followed by a nonlinear memoryless network followed by a linear filter): The block-oriented model has a faster convergence.

functionals which are a parametric family of real-valued mappings on arbitrarily normed linear spaces. Multi-layer functionals were then employed to define multi-layer operators: a parametric family of operators on certain spaces of real-valued functions on the real line. It was established that multi-layer operators can provide an input–output representation, in a certain sense, for any time-invariant system characterized by a continuous functional.

A general structure that is capable of approximating input–output maps of nonlinear discrete-time systems was recently introduced in [174]. The structure is composed of two stages, a dynamical stage followed by a memoryless nonlinear stage. A necessary and sufficient condition was given for a large set of structures of this form to be capable of approximating a wide class of nonlinear discrete-time systems. In particular, the authors show that a structure with a "complete memory" dynamical stage and a sufficiently powerful memoryless stage is able to approximate arbitrarily well a wide class of continuous, causal, approximately finite-memory, time-invariant mappings (i.e. systems with memory but with time-invariant coefficients) between discrete time spaces. Several examples of complete memory structures were presented, which provides a template for designing numerous NN architectures for dynamic system approximation (see also [165]).

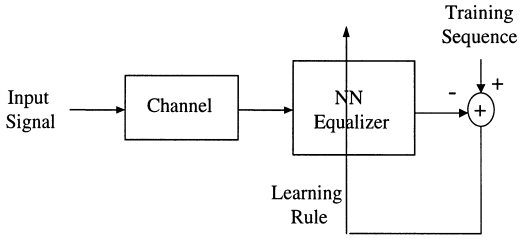

Fig. 13. Block diagram of (supervised) adaptive equalization.

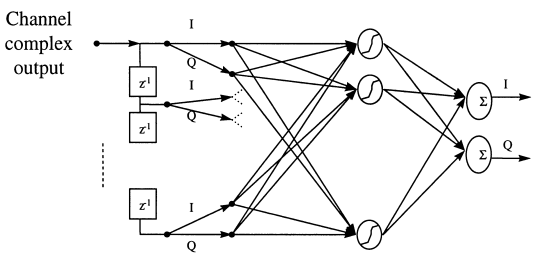

Fig. 14. Example of an MLP equalizer (the input and output signals are complex-valued).

3.2. Channel equalization

Adaptive channel equalization (Fig. 13) is a major issue in digital communication systems. Adaptive receivers are used to reduce channel disturbances such as noise, intersymbol interference (ISI), co-channel and adjacent channel interference, nonlinear distortions, fading, time-varying characteristics, etc. Neural network-based equalizers have been proposed as alternatives to classical equalizers and provided significant performance improvement in a variety of communication channels. This section reviews neural network approaches to channel equalization including real-valued networks, complex-valued MLP and RBF networks, polynomial perceptrons, cellular networks, recurrent neural networks, information geometry-based approaches, self-organizing maps, etc.

Numerous real-valued neural network-based adaptive equalizers have been introduced in the literature to overcome intersymbol interference (ISI), noise, and nonlinear distortions of digital communication channels (see e.g. [42,44,47,51, 108,127]). These equalizers employing various neural network structures, such as multi-layer perceptrons, radial basis function networks and modular networks, have been shown to equalize nonlinear channels successfully and outperform classical linear equalizers. For example, Chen et al. [43] showed that MLP equalizers (see e.g. Fig. 14) are able to generate complex nonlinear separation curves and thus equalize highly nonlinear channels. [47] presented a programmable VLSI neural network processor for communication receiver. A powerful channel equalizer was implemented with this processor chip configured as a four-layer perceptron. [88] introduced a functional-link neural-network-based decision feedback equalizer (DFE) in order to overcome inter-symbol interference, additive noise, and co-channel interference. The structure was shown to provide significantly superior BER performance compared to the conventional DFE and LTE equalizers, RBF equalizers, and MLP structures. Deviciana and Zakhor [58] proposed MLP-based receivers for continuous phase modulation channels. The paper gave an analytical study of the performance (i.e. probability of error) of NN equalizers. In particular, the study showed that for large SNR, the ratio between the output and input noise variance can be analytically predicted and depends on the number of saturated nodes, the product of the input and output weights, and the temperature parameter of the nonlinearity. Kirkland and Taylor [108] present a comparative study on the use of neural network architectures for channel equalization. They showed their advantages over conventional equalization schemes.

Complex-valued neural networks have been used for channel equalization with two-dimensional signaling, for instance quadrature amplitude modulation (QAM) and phase shift keying (PSK) modulation channels. Lo and Hafez [127] have studied the performance of conventional and complex perceptron-based equalizers of linear channels in the presence of intersymbol interference, additive noise, and co-channel interference. They concluded that the performance of the neural network equalizer matches that of the conventional equalizer under all noise and interference conditions. A complex neural model was applied in [195] for M-QAM data communications in the presence of co-channel interference.

A complex-valued version of radial basis function networks (RBF) has been introduced by Chen et al. [45,46]. They have shown that this structure is able to generate complicated nonlinear decision surfaces and to approximate an arbitrary nonlinear function in complex multi-dimensional space. They applied the complex RBF network to adaptive realization of a Bayesian solution for 4-QAM digital communication channel equalization. It was shown that the optimal Bayesian equalizer is structurally equivalent to the complex RBF network. This intimate connection was exploited to develop fast training algorithms for implementing a Bayesian equalizer based on RBF networks.

A lattrice polynomial perceptron (LPP) has been applied to 64QAM cellular mobile communication channels [196]. The basic idea of the equalizer is to obtain the Gram–Schmidt orthogonal decomposition of the input signals of the polynomial perceptron and then compute the desired response signal as the linear combination of the transformed signals through nonlinear activation function. The LPP was applied to a frequency-selective slow fading channel with adjacent-channel interferences. The LPP was shown to outperform other equalizers such as the conventional DFE.

A cellular neural network [49] has been designed for maximum likelihood sequence estimation (MLSE) of signals in the presence of ISI and noise [168]. The performance of the neural network MLSE receiver can be improved by a parallel hardware annealing technique. Sheu et al. [168] addressed the hardware architecture, network models, and neuron models in terms of BER performance, and compared the results to the Viterbi algorithm. Simulation results showed that the cellular neural network is very efficient in realizing the MLSE receiver. Other hardware implementation issues are studied in [20].

Recurrent neural networks (see e.g. [194]) are highly nonlinear dynamical systems that exhibit a rich and complex dynamical behavior. This property makes them suitable for highly nonlinear channel equalization. Kechriotis et al. [106] used the real-time recurrent learning (RTRL) algorithm [194] which was extended to the complex plane. The algorithm was shown to have comparable performance with the traditional linear TDL equalizers for the case of linear channels. It outperformed them when the channel transfer function has either spectral nulls or severe nonlinear distortions. Moreover, recurrent neural networks were shown to outperform multi-layer perceptron equalizers (which have larger computational complexity) for linear and nonlinear channel equalization cases. The RTRL algorithm was also extended to the case of blind equalization, and showed better performance than the constant modulus algorithm (CMA) in both linear and nonlinear cases.

Another approach to recurrent neural network learning has been proposed in [149]. The method exploits the principle of discriminative learning [99] which minimizes an error functional that is a direct measure of the classification error. The authors gave an alternative to gradient-based learning algorithms (such as backpropagation in feedforward networks) which are characterized by low convergence rate and may not be suitable in applications where fast convergence is required. They applied least-squares (LS) methods (which are commonly used in the signal processing domain) to fully recurrent neural networks. The proposed approach was able to provide higher convergence speed with respect to gradient-based methods. The method was shown to outperform the RTRL algorithm for the equalization of linear and nonlinear channels.

Another alternative to the MSE criterion-based algorithms has been introduced by Adali et al. [2,4]. In their approach, the conditional probability distribution function (PDF) of the transmitted signal given the received signal is parametrized by a general neural network structure. The PDF parameters are estimated by minimization of the accumulated relative entropy cost function. This least relative entropy (LRE) equalizer has been shown to perform high complex decision boundaries, and to track abrupt changes in a nonlinear channel response whereas the mean square error-based multi-layer perceptron equalizer cannot.

The self-organizing map has been combined with conventional equalizers such as linear transversal equalizer (LTE) and decision feedback equalizer (DFE). In these methods, the SOM is either in cascade or in parallel with the conventional equalizer [112]. The m_i vectors of the SOM

define the adaptive decision levels in the detector part. If we denote by $y(n)$ the output of the DFE equalizer part, then the output error, $e(n) = y(n) - m_i(n)$, controls the adaptation of the DFE. Thus, the purpose of the DFE is to overcome the dynamic linear distortions, while the SOM adaptively compensates for the nonlinear distortions. This structure was applied to a nonlinear two-path channel, showing that the SOM-based equalizer outperformed conventional equalizers for different nonlinearity types and SNR levels.

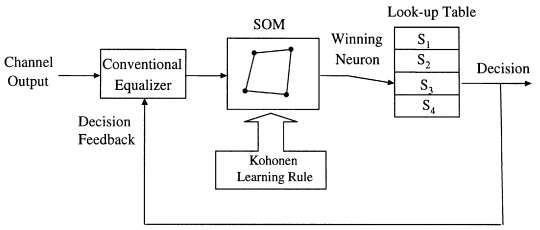

Fig. 15. Example of an SOM equalizer for 4-QAM signals (in the decision directed mode).

Neural networks have been recently applied to satellite UMTS channel equalization (NEWTEST ACTS European Project) [73]. The project applied several NN architectures and several combinations of NN with classical techniques (e.g. the linear transversal equalizer (LTE) combined with SOM, LTE combined with RBF, etc., see Figs. 15–17). Several architectures have been implemented by the project for real-time trials. The most important result of the project is that NN approaches outperform classical equalizers when complicated modulation schemes are used (e.g. M-QAM modulations, $M > 4$) [35,36].

3.3. Coding, decoding and error correcting codes

Coding, decoding and error correcting codes are very promising fields of NN applications. In these

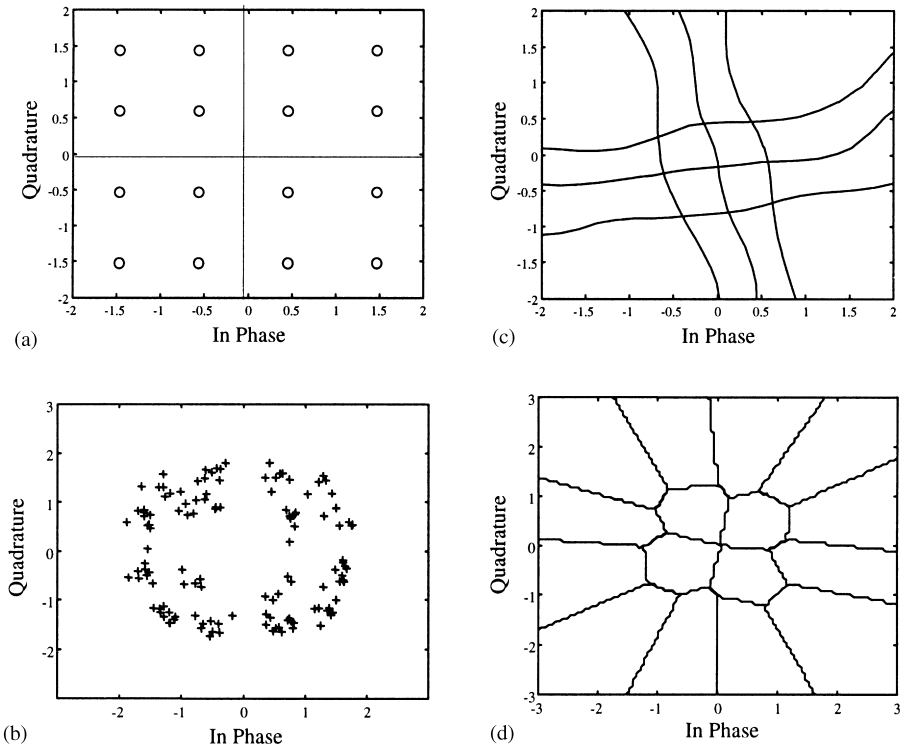

Fig. 16. Example of decision boundaries: (a) A 16-QAM signal has been transmitted through a non linear satellite channel, (b) represents the output constellation of an LTE equalizer, (c) represents the decision boundaries of an LTE combined to an MLP equalizer, (d) represents the decision boundaries of an LTE combined to an SOM equalizer.

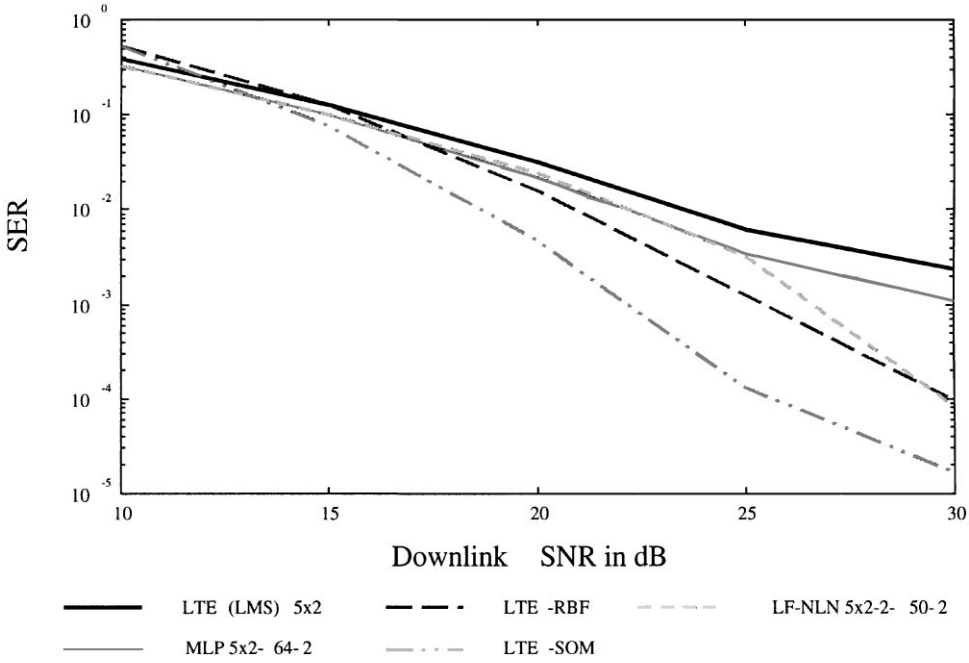

Fig. 17. Equalization of a satellite mobile channel: Symbol error rate (SER) vs. downlink SNR for 16-QAM signal with 25 dB uplink SNR, −5dB backoff and 150 km/h mobile speed.

domains, parallel calculations have to be performed and high computational speeds are required. The adaptability and flexibility of neural networks allow them to efficiently operate in complex situations, where some of the simplifying assumptions of standard coding and decoding techniques are not fulfilled.

In [37] the authors establish an analogy between the concept of error-correcting codes and the concept of neural networks. Performing maximum likelihood decoding in a linear block-error correcting code is shown to be equivalent to finding a global maximum of the energy function of a certain NN. Given a linear block code, an NN can be constructed in such a way that every local maximum of the energy function corresponds to a code word and every code word corresponds to a local maximum. The equivalence between the two concepts led the authors to obtain very interesting results. For example they have found a simple proof for the minimum distance of Reed–Muller codes by considering the equivalent problem in polynomials. Another result is that solvable cases in one problem can be used to identify non solvable cases in the equivalent problem. In particular they showed how to use the Hamming code to solve certain 0–1 nonlinear problems that are not solvable by known techniques.

An interesting application can be found in [148] where the authors constructed a neural network for decoding channels in additive white gaussian noise for the BCH(7,4) code (Fig. 18). They showed that the use of NNs for error correction is advantageous in the case of complex situations where some simplifying assumptions of standard decoding techniques are not fulfilled. The NN approach was applied to the case of colored noise and showed remarkable performance compared to classical decoding techniques.

Neural networks have been applied as block-parallel decoders for convolutional codes in [163]. The NN structure is based on hetero-associative associations and modified Hamming network concepts. These NN decoders allow noise-free

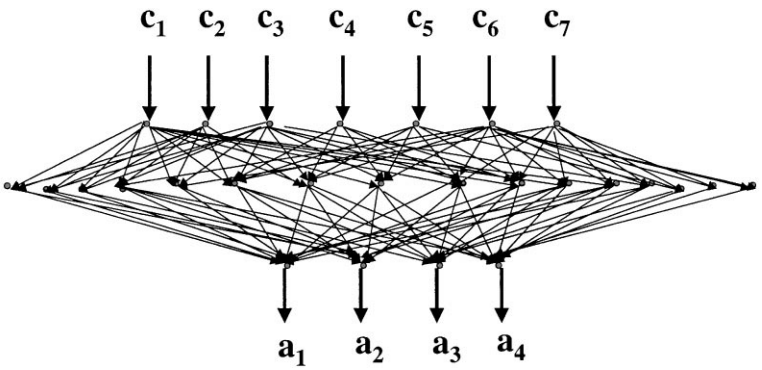

Fig. 18. Error correcting neural network for the BCH (7,4) code: seven values of each word are fed to the seven input nodes, there are 2^4 intermediate neurons each of them corresponding to a codeword, the four output neurons represent the k information bits of the block considered.

data to be decoded without adding processing noise. The study shows that NNs can reduce the computational complexity of the Viterbi decoding by means of a block-parallel implementation, producing similar error correcting capabilities.

An artificial neural network Viterbi decoder is presented in [185]. It matches the performance of the ideal Viterbi decoder. The analog neuron implementation of the NN decoder allows it to be significantly faster than comparable digital-only designs. The connection weights of the ANN decoder is either $+1$ or -1, which eliminates weights considerations in the implementation. The NN Viterbi decoder is parallel, simple and suited for VLSI implementation, which can yield a very high-speed Viterbi decoder.

3.4. Spread spectrum applications

In the recent years, several researchers have been interested in the design of neural network receivers for code division multiple access (CDMA) signals. To demodulate the received CDMA signal, we need to overcome the inherent channel noise, which is in general assumed to be Gaussian, and the multiple access interference (MAI). The conventional receiver can recover the information by passing the information through a bank of filters which are matched to the users signature waveforms. Since the conventional receiver ignores the presence of interfering signals, it is reliable only when there is a small number of simultaneous transmissions. One of the major limitations of the conventional receiver is that its performance decreases severely when the relative received power of the interfering signals becomes large (the so-called near-far problem). Verdû [183] has shown that optimum near-far resistant multi-user receiver can be obtained. The optimal detector has, however, its computational complexity that increases exponentially with the number of users. Near-optimal receivers have also been proposed in the literature (e.g. [128,182]). These receivers require the knowledge of the spreading codes of all users. This section reviews the spread spectrum applications of neural networks and shows their contribution compared to conventional techniques.

Several authors have proposed neural networks as multi-user receivers in CDMA communications (see Figs. 19 and 20). These approaches were motivated by

(i) neural receivers are adaptive; this makes them able to track a changing communication environment or to adapt to unknown user parameters, such as unknown spreading codes,

(ii) neural networks perform nonlinear decision regions,

(iii) the cyclostationarity of the multiple access interference (MAI) suggests that an adaptive NN receiver should be able to learn how to remove the MAI,

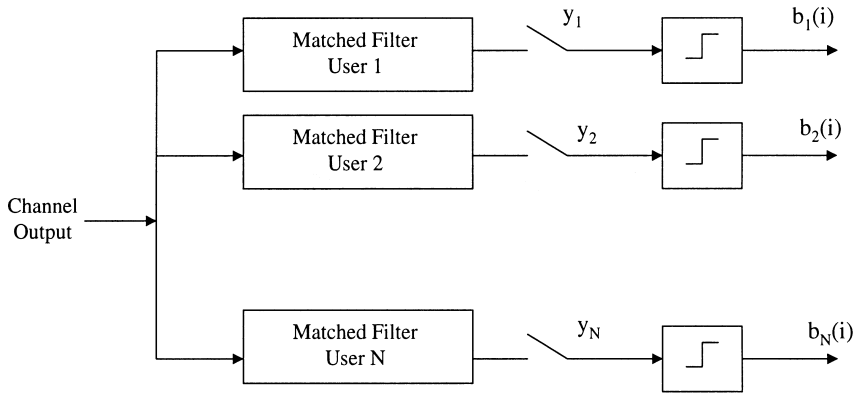

Fig. 19. Conventional CDMA receiver.

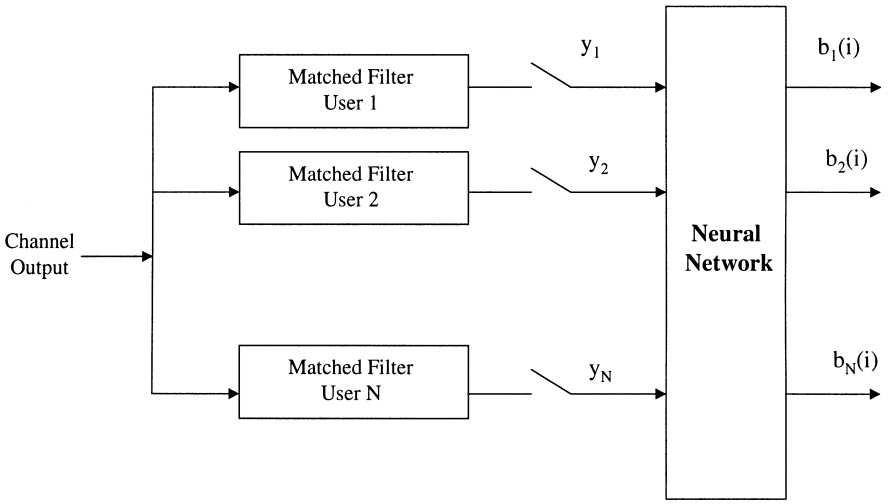

Fig. 20. Block diagram of a CDMA receiver using a neural network.

(iv) the NN receivers have efficient computational complexity and are near-far resistant,

(v) NN performances are close to the optimal ones.

A two-layer perceptron CDMA receiver has been proposed in [1]. The NN weights were updated using the BP algorithm and a modified version of the BP algorithm. The performance of the NN receivers were compared to the conventional and optimum receivers in synchronous and asynchronous systems. The NN outperformed the conventional receiver and had a performance which is comparable with that of the optimal receiver. The structure suffers, however, its high complexity, since the number of nodes increases exponentially as the number of users increases.

This work has been extended by Mitra and Poor [136] by providing comparisons of the perceptron performance to that of a linear adaptive receiver as well as providing a convergence analysis of the NN algorithms in the multi-user context. Through several simulation examples, the adaptive receivers have shown similar performance. Note that the adaptive receivers do not require the knowledge of the spreading codes of the interfering users.

Another approach based on Hopfield [86] neural networks (HNN) was introduced by Kechriotis and Manolakos in [104]. The proposed HNN detector was proved to be a powerful generalization of the multi-stage detector (OMD) (which is a suboptimal detector [182]). The objective function of the optimal multi-user detector problem was translated into an HNN energy function. The HNN structure was evaluated through several simulations and showed better performance than the multi-stage detector and approached the optimal detector at a much lower computational cost. The NN receiver can be implemented in analog VLSI, which makes it well suited for high-speed communication applications.

The HNN structure was combined with a signal processing detector to form a hybrid signal processing-NN receiver [105]. This structure takes advantage of both digital signal processing and analog neural network implementations at a reasonable computational and hardware cost. The hybrid detector consists of a digital signal processing stage which processes the outputs of the conventional detector in order to reduce the OMD optimization problem, followed by an HNN that may be used to solve a remaining optimization problem of manageable size that has the same structure as the OMD. It was shown through several simulations that the hybrid detector exhibits superior performance when compared to the multi-stage detector at a much lower computational cost.

The problem of power prediction in mobile communication systems has been addressed in [68]. A neural-network-based predictor was proposed for received power level prediction in direct sequence (DS) CDMA systems. The predictor was composed of an Adaline (i.e. Adaptive Tapped Delay Line [191]) followed by a multi-layer perceptron. An information criterion-based model selection principle, the predictive minimum description length method [159], was applied to select optimal neural network structures (i.e. the optimal number of input and hidden nodes). The hybrid neural predictor structure is first optimized off-line. The optimized structure is then used with on-line adaptation for predictive filtering of the noisy power signal, the statistical characteristics of which may change drastically with time. The neural network approach was applied to very noisy Rayleigh fading channels with $1.8-\mathrm{GHz}$ carrier frequency. Simulation results showed that the optimal NN predictor can provide smoothed in-phase and quadrature signals with SNR gains of about 12 and 7 dB at the urban mobile speeds of 5 and 50 km/h, respectively. The NN predictor offers higher noise attenuation and wider prediction bandwidth than the linear predictor [68].

3.5. Vector quantization and image processing

Image compression is an important tool for several applications in telecommunications such as satellite remote sensing, multimedia communications, broadcast television, UMTS communications, Internet, etc. Vector quantization is a very efficient compression method for image coding systems [71]. Neural networks are interesting alternatives to classical image processing and compression techniques due to their quality image reproduction, computational effort, and adaptation. Their massively parallel structure, high degree of interconnections, capability of learning and self-organization allow them to solve several problems in processing image data. In contrast, image data processing approaches based on serial paradigm of information processing, are more suited for sequential or scalar information [60]. This section reviews neural network approaches to vector quantization and image processing, including MLP networks, PCA algorithms, SOMs, fuzzy SOMs, etc.

The nonlinear processing performed by neural networks allows to design nonlinear predictors for predictive coding. For example, a multi-layer network has been shown to outperform linear predictors in terms of predictive gain [59]. The MLP consisted of 3-unit input layer, 30-unit hidden layer, and one output unit (Fig. 21). The experimental results showed about 4 dB improvement SNR over the optimal linear predictive coding system. Higher-order models, based on discrete-time Volterra expansions, were also applied to nonlinear prediction [131]. The system is equivalent to a generalized autoregressive (AR) model. The motivation for using a nonlinear expansion is that AR image models do not adequately account for sharply defined structures such as edges in images. Thus,

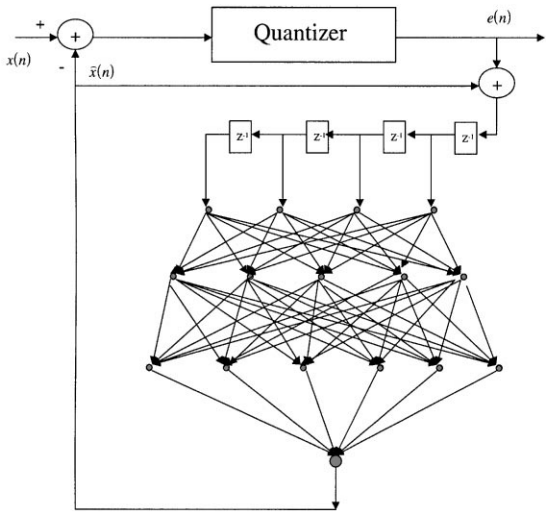

Fig. 21. Example of predictive coding using a multi-layer neural network.

higher-order nonlinear terms are required. Significant improvements over optimal linear predictors were shown for 1-D and 2-D predictors. Pipelined recurrent neural networks (PRNN) were used for nonlinear and nonstationary signal prediction [77,79]. Haykin and Li applied PRNN to speech signals, the prediction gain was improved by 3 dB compared to a linear FIR filter trained with the LMS algorithm [79].

Several algorithms have been employed for image principal component extraction in block transform coding. Compared to standard eigen decomposition techniques, principal component analysis (PCA) algorithms have the advantage of being more computationaly efficient and require small storage overhead, in addition to their adaptability over long-term variations of the image statistics. Numerous PCA algorithms have been proposed in the literature (see e.g. [77,147]). For example, Oja's learning rule [147] has been shown to converge to the largest principal component [77,147]. Sanger [166] extended Oja's rule to compute the learning of the first M principal components. These algorithms use only feed forward connections. Other algorithms use feed forward and feedback connections, they are also called "decorrelating algorithms" because the feedback connections decorrelate the successive outputs. Kung and Diamatras [117,118] applied the adaptive principal components extraction (APEX). The advantages of PCA algorithms over standard eigenvalue decomposition algorithms are that (i) these iterative techniques can be more computationaly efficient, (ii) because of their iterative nature, they can be allowed to adapt to slowly varying changes, (iii) no extra overhead is needed to store the data or its higher-order statistics, and (iv) if an extra basis is needed, its computation would be more computationaly efficient.

Kohonen learning rule has been extensively applied to vector quantization. Classical VQ techniques are rather sensitive to codeword errors. For example, if a code word y is transmitted through a noisy channel, it may be changed to some other codeword y'. Thus, if x is the sample vector corresponding to y and x' is the sample vector corresponding to y', in a nonordered codebook, x and x' may look completely different, and therefore the corresponding codeword will be decoded with a considerable error. One of the advantages of the SOM is its capability to form ordered topological feature maps, i.e. after training, the SOM weight vectors are spatially ordered in the array, i.e. the neighboring vectors in the map are more similar than the more remote ones. Encoding means conversion of the new input vector into the discrete array coordinates of the winner in the SOM. For the decoding side, the coordinates of an identical SOM array are selected. In the case of a noisy channel, if a transmitted code y is transformed into y', then the ordering of the map guarantees that if y' is similar to y, then the decoded sample vector x' is similar to the transmitted sample vector x [112]. The SOM algorithms provide a codebook in which the distortion effects are automatically taken into account. This has been demonstrated in [112] where an SOM-based error-tolerant image compression was compared to a nonordered codebook (i.e. a randomly ordered codebook). The results showed that, in the image with random order in the codewords, the errors are rather severe, e.g., in the middle of dark areas there are clear blocks, and dark blocks are inserted in clear areas. In the image with error-tolerant coding, the errors are smoother, e.g., in the dark areas the erroneous blocks are "almost" dark. The subjectively experienced

qualities of the images differ significantly, although the same number of codeword errors was present in both the images. In [145] the self-organizing map was shown to outperform the LBG algorithm and allowed faster convergence. Leung and Chan [123] proposed an SOM-based approach to properly create the mapping from the codebook of the quantizer to the channel signal set of the communication system. The neighborhood structure of the SOM and the neighborhood of the channel signal set were used to construct the mapping. This approach was shown through simulations to be robust with respect to channel noise.

The SOM algorithm has been improved by using a frequency sensitive competitive learning algorithm (FSCL) [5]. The idea is to increase the probability of underutilized neurons being selected as winners. This is accomplished by introducing a distortion measure that ensures all codewords in the codebook to be updated with a similar probability. The FSCL algorithm yielded MSE and SNR performances which are comparable to the locally optimal LBG algorithm [5,67]. The codebooks obtained by the FSCL algorithm had a sufficient entropy so that Huffman coding of the VQ indices would not provide significant additional compression. Moreover, the FSCL algorithm allows to reduce underutilization problems [5,67]. Vector quantization has been applied to the problem of displaying natural images with a reduced set of colors (colormap) and to the interframe coding of image sequences [120]. A competitive learning algorithm with an efficient initialization strategy (CL-TS) that solves the problem of underutilized nodes was proposed. The CL-TS algorithm outperformed the FSCL, SOM, and LBG algorithms.

Combined fuzzy-SOM algorithms have been proposed in [102] for learning vector quantization (FALVQ). The authors established a relationship between the form of the membership function used for formulating the LVQ problem and the competition between the prototypes during the learning process. This was achieved by introducing two interference functions which depend on the membership function and affect the update of the winning and nonwinning prototypes. The algorithms were applied to design high-quality vector quantizers for nontrivial tasks (see Refs. [33,102] for other fuzzy LVQ approaches).

3.6. Nonlinear signal processing

The most frequently used signal processing tools have been linear in nature from the origin. Linear filters have the advantage of being easily analyzed and implemented. By minimizing the MSE criterion, it is usually possible to find the optimal linear filter. Moreover, linear filters are optimal among all types of filters when the noise is additive white Gaussian. However, in many signal processing and communication applications, the channel through which the signal is transmitted is not additive and Gaussian, it is not stationary and may have unknown statistics. Thus, nonlinear filters have been proposed as an alternative to linear filters. The nonlinear processing of neural networks and their adaptive behavior made them well suited for nonlinear filtering.

Yin et al. [200] defined a new class of nonlinear filters called neural filters based on the threshold decomposition and neural networks. It was shown that the neural filters include all filters defined either by continuous functions, such as FIR filters, or by boolean functions, such as generalized stack filters. Two adaptive algorithms were derived for finding optimal neural network filters under the MSE criterion and mean absolute error (MAE) criterion, respectively. These algorithms were applied to 1-D and 2-D signal processing showing that they can efficiently remove various kinds of noise such as Gaussian noise and impulsive noise.

Zhang et al. [204] addressed the structure and properties of generalized adaptive neural filters (GANFs). GANFs encompass a large class of nonlinear sliding-window filters which include stack filters. The authors demonstrated that the MAE of the optimal GANF is upper-bounded by that of the optimal stack filter. Thus, the optimal GANF is expected to perform better in noise suppression than stack filters. The proposed adaptive algorithms for configuring GANF are effective and robust in both one-dimensional signal processing and image processing.

An interesting work has been performed by Fung et al. [66] to derive neural network transfer

functions. The authors derived explicit expressions for the frequency response or generalized transfer functions of both feed forward and recurrent neural networks in terms of the network weights. The derivation of the algorithm was established on the basis of the Taylor series expansion of the activation functions used in a particular neural network. This lead to a representation which is equivalent to the nonlinear recursive polynomial model and enabled the transfer function to be based on the harmonic expansion method. The application examples presented in the paper showed that the frequency response appear to be highly sensitive to the network topology and training, and that the time-domain properties fail to reveal deficiencies in the trained network structure. This suggests for example that selecting networks based on the MSE performance will be very difficult if the objective is to construct a model which characterizes the dynamics of the underlying system rather than to provide good time-domain predictions.

The optimum nonlinear filtering problem has been recently addressed in [81]. Two fundamentally different approaches to this problem have been considered. The first is the stochastic partial differential equation (SPDE) approach which provides a mathematical basis as well as a numerical solution for solving the optimum nonlinear filtering problem. The second approach is based on the use of a neural filter, which can be implemented using a recursive multi-layer perceptron or radial basis function network. The design of the neural network requires the availability of training data that is representative of the domain of interest. The advantage of the NN approach is its computationaly feasible solution to the nonlinear filtering problem without having to formulate a stochastic model for the problem. Its drawback is that the designer does not know how far he is from the optimal solution (because the learning process being trapped in a local minimum of the error surface, or a filter of adequate architecture being not used). From the critiques of the two approaches, the authors conclude that much can be gained by working with both approaches side by side. In particular, a numerical solution to the SPDE approach produces the optimum nonlinear filtering performance attainable, which, in turn, can be used to fine tune a computationaly feasible neural filter or to seek its correct complexity.

An interesting approach that introduced analog neural networks to the framework of the adaptive filtering domain was proposed by Baykal and Constantinides in [21]. Baykal and Constantinides [21] reveal that there is a significant potential behind analog neural network implementations of digital signal processing applications. In particular, the paper reformulated the underdetermined-order recursive least-squares (URLS) algorithm [22] via a neural network architecture. The response of the neural network is seen to be identical to that of the algorithmic approach. This network avoids the drawbacks of digital computation such as error propagation and matrix inversion, which is ill-conditioned in most cases. It is numerically attractive because it performs an implicit matrix inversion. The neural network offers also the flexibility of easy alteration of the prediction order of the URLS algorithm which may be crucial in some applications. The neural network can be easily integrated into a digital system by appropriate digital-to-analog and analog-to-digital converters. Other examples of reformulation of conventional optimization and signal processing applications can be found e.g. in [48,58].

3.7. Other neural network applications to digital communications

The previous sections presented some typical applications of neural networks in digital communications, this section presents other applications that have not been developed in the paper.

Neural networks have been widely used for adaptive traffic control in ATM networks (e.g. [75,177]) and traffic management in telephone networks (e.g. [16]). One of the major applications of neural networks is statistical pattern recognition (e.g. computer vision [119], texture analysis and classification [33,113], and speech recognition [110]). Several authors have applied neural networks to mobile and satellite communications (e.g. for frequency assignment in satellite communication systems [64], cellular mobile communication planning using self-organizing maps [63], and

time-slot assignment in time-division multiplexing hierarchical switching systems [65]). Neural networks have been also used for fault detection in communication systems [85], analog and digital modulation recognition [142], and for equalization of data storage channels [140,141].

4. Conclusion

The paper gives an overview of the applications of neural networks to digital communications. The paper aims at covering the most representative applications of neural networks such as channel modeling and identification, equalization, coding and decoding, image processing, etc. The abundant literature that has grown exponentially in the recent years shows that the communications community gives great interest to neural network-based communication systems. Neural networks will certainly be one of the key technologies for the communication domain in the 21st century, and accordingly a special effort may be expected to be paid to real-time hardware implementation issues.

5. Further reading

The following references are also of interest to the reader: [3,17,18,27,56,62,74,76,80,103,107,116, 134,139,146,150,152,156,161,167,170,171,175,176, 184,186,187,193,197,199,202].

References

[1] B. Aazhang, B. Paris, G. Orsak, Neural networks for multiuser detection in code-division multiple-access communications, IEEE Trans. Commun. 40 (7) (July 1992) 1212–1222.

[2] T. Adali, X. Liu, M.K. Sonmez, Conditional learning with neural networks and its application to channel equalization, IEEE Trans. Signal Process. 45 (4) (April 1997).

[3] T. Adali, M.K. Sonmez, X. Liu, Partial likelihood for real-time signal processing, Proceedings of ICASSP'96, 1996, pp. 3561–3564.

[4] T. Adali, M.K. Sonmez, K. Patel, On the dynamics of the LRE algorithm: a distribution learning approach to adaptive equalization, Proceedings of IEEE ICASSP'95, Detroit, MI, April 1995, pp. 929–932.

[5] S.C. Ahalt et al., Competitive learning algorithms for vector quantization, Neural Networks 3 (1990) 277–290.

[6] S. Amari, Theory of adaptive pattern classifiers, IEEE Trans. Electron. Comput. EC-16 (June 1967) 299–307.

[7] S. Amari, Differential Geometry Methods in Statistics, Springer Lecture Notes in Statistics, Vol. 28, Springer, Berlin, 1985.

[8] S. Amari, Mathematical foundations of neurocomputing, Proc. IEEE 78 (9) (September 1990) 1443–1463.

[9] S. Amari, A universal theorem for learning curves, Neural Networks 6 (1993) 161–166.

[10] S. Amari, Information geometry of the EM and em algorithms for neural networks, Neural Networks 8 (9) (1995) 1379–1408.

[11] S. Amari, Natural gradient works efficiently in learning, Neural Computation 10 (1998) 251–276.

[12] S. Amari, J.F. Cardoso, Blind source separation-semiparametric statistical approach, IEEE Trans. Signal Process. 45 (11) (November 1997) 2692–2700.

[13] S. Amari, T. Chen, A. Cichocki, Stability of learning algorithms for blind source separation, Neural Networks 8 (1997) 1345–1351.

[14] S. Amari, N. Fujita, S. Shinomoto, Four types of learning curves, Neural Computation 4 (1992) 605–618.

[15] S. Amari, K. Kurata, H. Nagaoka, Information geometry of boltzmann machines, IEEE Trans. Neural Networks 3 (2) (March 1992).

[16] B. Ambrose, R. Goodman, Neural networks applied to traffic management in telephone networks, IEEE Proc. 84 (10) (October 1996) 1421–1429.

[17] N. Ansari, Y. Chen, Dynamic digital satellite communication network management by self-organization, International Joint Conference on Neural Networks, IJCNN'90, Vol. 2, 1990, pp. 567–570.

[18] R. Arnott, Diversity combining for digital mobile radio using radial basis function networks, Signal Processing 63 (1) (1997) 1–16.

[19] T. Back, H.P. Schwefel, An overview of evolutionary algorithms for parameter optimization, Evol. Comput. 1 (1) (1993).

[20] S.H. Bang, B.J. Sheu, J. Choi, Programmable VLSI neural network processors for equalization of digital communication channels, Proceedings of International Workshop on applications of Neural Networks to Telecommunications, LEA Publishers, 1993, pp. 1–12.

[21] B. Baykal, A. Constantinides, A neural network approach to the underdetermined-order recursive least-squares adaptive filtering, Neural Networks 10 (8) (1997) 1523–1531.

[22] B. Baykal, A. Constantinides, Underdetermined-order recursive least-squares adaptive filtering, the concept and algorithms, IEEE Trans. Signal Process. 45 (1997) 346–362.

[23] F. Beaufays, B. Widrow, Two-layer structures for fast adaptive filtering, Proceedings of WCNN'94, San Diego, CA, June 1994, pp. III-87-93.

[24] S. Benedetto, E. Biglieri, Non linear equalization of digital satellite channels, IEEE J. Selected Areas Commun. 1 (1) (January 1983).

[25] S. Benedetto, E. Biglieri, V. Castellani, Digital Transmission Theory, Prentice-Hall International, Englewood Cliffs, NJ, 1987.
[26] S. Benedetto, E. Biglieri, R. Daffara, Modeling and performance evaluation of nonlinear satellite links – A Volterra series approach, IEEE Trans. Aerospace Electron. Systems 15 (4) (July 1979).
[27] N. Benvenuto, M. Marchesi, F. Piazza, A. Uncini, Non linear satellite radio links equalized using blind neural networks, IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'91, 1991, pp. 1521–1524.
[28] A. Berman, C. Mahle, Non linear phase shift in traveling-wave tubes as applied to multiple access communications satellites, IEEE Trans. Commun. 18 (1) (February 1970).
[29] N.J. Bershad, M. Ibnkahla, G. Blowens, J. Cools, A. Soubrane, N. Ponson, Second order fluctuation analysis of a two-layer neural network used for modeling nonlinear memoryless channels, IEEE Trans. Signal Process. (May 1999) 1297–1303.
[30] N.J. Bershad, M. Ibnkahla, F. Castanié, Statistical analysis of a two-layer perceptron back propagation algorithm used for modeling non-linear memoryless channels: the single neuron case, IEEE Trans. Signal Process. 45 (3) (March 1997) 747–756.
[31] N. Bershad, J.J. Shynk, P. Feintush, Statistical analysis of the single-layer backpropagation algorithm: Part I, IEEE Trans. Signal Process. 41 (2) (February 1993) 573–582.
[32] N. Bershad, J.J. Shynk, P. Feintush, Statistical analysis of the single-layer backpropagation algorithm: Part II, IEEE Trans. Signal Process. 41 (2) (February 1993) 583–590.
[33] J.C. Bezdek, A review of probabilistic, fuzzy, and neural models for pattern recognition, J. Intell. Fuzzy Systems 1 (1) (1993) 1–25.
[34] S.A. Billings, S. Chen, M.J. Korenberg, Identification of MIMO non linear systems using a forward regression orthogonal estimator, Int. J. Control 49 (1989) 2157–2189.
[35] S. Bouchired, Equalization of time-varying non-linear channels using neural networks: application to the satellite mobile channel, Ph.D. Thesis, INP, Toulouse, France, 1999.
[36] S. Bouchired, M. Ibnkahla, D. Roviras, F. Castanié, Equalization of satellite UMTS channels using neural network devices, Proceedings of IEEE ICASSP'99, Phoenix, May 1999.
[37] J. Bruck, M. Blaum, Neural networks, error-correcting codes, and polynomials over the binary n-cube, IEEE Trans. Inform. Theory 35 (5) (September 1989).
[38] J.F. Cardoso, B. Laheld, Equivariant adaptive source separation, IEEE Trans. Signal Process. 44 (1996) 3017–3030.
[39] G. Carpenter, Neural networks models for pattern recognition and associative memory, Neural Networks 2 (1989) 243–257.
[40] G. Carpenter, S. Grossberg, The ART of adaptive pattern recognition by a self-organizing neural network, IEEE Comput. (March 1988) 77–88.
[41] L.W. Chan, The fuzzy polynomial rules, Proceedings of WCNN'94, San Diego, CA, June 1994, pp. 1–805–810.
[42] S. Chen, C. Cowan, P. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 (2) (March 1991).
[43] S. Chen, G. Gibson, C. Cowan, P. Grant, Adaptive equalization of finite non-linear channels using multilayer perceptron, Signal Processing 20 (1990) 107–119.
[44] S. Chen, G. Gibson, C. Cowan, P. Grant, Reconstruction of binary signals using an adaptive radial-basis-function equalizer, Signal Processing 22 (1991) 77–93.
[45] S. Chen, S. McLaughlin, B. Mulgrew, Complex-valued radial basis function networks, Part I: Network architecture and learning algorithms, Signal Processing 35 (1) (January 1994) 19–31.
[46] S. Chen, S. McLaughlin, B. Mulgrew, Complex-valued radial basis function networks, Part II: Application to digital communication channel equalization, Signal Processing 36 (2) (March 1994) 175–188.
[47] J. Choi, S. Bang, B. Sheu, A programmable analog VLSI neural network processor for communication receivers, IEEE Trans. Neural Networks 4 (3) (May 1993) 484–495.
[48] A. Chicocki, R. Unbehauen, Simplified neural networks for solving linear least-squares and total least-squares problems in real time, IEEE Trans. Neural Networks 5 (1994) 910–923.
[49] L. Chua, L. Yang, Cellular neural networks: theory, IEEE Trans. Circuits Systems 35 (October 1988) 1257–1272.
[50] F.L. Chung, T. Lee, Fuzzy competitive learning, Neural Networks 7 (3) (1994) 539–551.
[51] J. Cid-sueiro, A. Figueiras-Vidal, Digital equalization using modular neural networks: an overview, in: E. Biglieri, M. Luise (Eds.), Signal Processing for Digital Communications, Springer, London, 1996.
[52] T. Clarke, Generalization of neural networks to the complex plane, International Joint Conference on Neural Networks, IJCNN'90, Vol. 2, 1990, pp. 435–440.
[53] F. Comelbas, R. Roca, Using genetic algorithms to design constant weight codes, Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications, LEA Publishers, 1993, pp. 119–124.
[54] J. Connor, R. Martin, L. Atlas, Recurrent neural networks and robust time series prediction, IEEE Trans. Neural Networks 5 (2) (1994).
[55] M. Cottrell, J.C. Fort, A statistic model of retinotopy: a self organizing process, Biol. Cybernet. 53 (1986) 405–411.
[56] A. Culhane, M. Peckerar, C. Marrian, A neural net approach to discrete Hartley and Fourier transforms, IEEE Trans. Circuits Systems 36 (1989) 695–703.
[57] G. Cybenko, Approximation by superposition of a sigmoidal function, Math. Control Signals Systems 2 (1989) 303–314.

[58] G. De Veciana, A. Zakhor, Neural net-based continuous phase modulation receivers, IEEE Trans. Commun. 40 (8) (August 1992).
[59] S. Dianat, N. Nasrabadi, S. Venkataraman, A non linear predictor for differential pulse-code encoder (DPCM) using artificial neural networks, Proceedings of ICASSP'91, Toronto, Canada, May 1991, pp. 2793–2796.
[60] R.D. Dony, S. Haykin, Neural network approaches to image compression, IEEE Proc. 83 (2) (February 1995).
[61] C. Duvanaud et al., High efficiency class F GaAs FET amplifiers operating with very low bias voltages for use in mobile telephones at 1.75 GHz, IEEE Microwave Guided Wave Lett. 3 (August 1993) 268–270.
[62] A. Esposito, S. Rampone, R. Tagliaferri, A neural network for error correcting decoding of binary linear codes, Neural Networks 7 (1) (1994) 195–202.
[63] T. Fritsch, Cellular mobile communication design using self-organizing feature maps, in: B. Yuhas, N. Ansari (Eds.), Neural Networks in Telecommunications, Kluwer Academic Publishers, Dordrecht, 1994, pp. 143–172.
[64] N. Funabiki, S. Nishikawa, A gradual neural network approach for frequency assignment in satellite communication systems, IEEE Trans. Neural Networks 8 (6) (November 1997) 1359–1370.
[65] N. Funabiki, Y. Takefuji, A parallel algorithm for time-slot assignment problems in TDM hierarchical switching systems, IEEE Trans. Commun. 42 (10) (October 1994) 2890–2898.
[66] C. Fung, S. Billings, H. Zhang, Generalized transfer functions of neural networks, Mech. Systems Signal Process. 11 (6) (1997) 843–868.
[67] A. Galanopoulos, J. Fowler, S.C. Ahalt, Vector quantization using artificial neural network models, in: E. Biglieri, M. Luise (Eds.), Signal Processing for Digital Communications, Springer, London, 1996, pp. 347–357.
[68] X.M. Gao, X.Z. Gao, J. Tanskanen, S. Ovaska, Power prediction in mobile communication systems using an optimal neural network structure, IEEE Trans. Neural Networks 8 (6) (November 1997).
[69] M. Ghogho, S. Meddeb, J. Bakkoury, Identification of time-varying non linear channels using polynomial filters, Proceedings of IEEE Workshop on Non Linear Signal and Image Processing, Michigan, USA, 1997.
[70] F. Girosi, T. Poggio, Networks and the best approximation property, Biol. Cybernet. 63 (1990) 169–176.
[71] R. Gray, Vector quantization, IEEE ASSP Mag. (April 1984) 4–29.
[72] S. Grossberg, Self-organizing neural networks for stable control of autonomous behavior in a changing world, in: J.G. Taylor (Ed.), Mathematical Approaches to Neural Networks, North-Holland, Amsterdam, 1993, pp. 139–197.
[73] A. Guntsch, M. Ibnkahla, G. Losquadro, M. Mazzella, D. Roviras, A. Timm, EU's R&D activities on third generation mobile satellite systems (S-UMTS), IEEE Commun. Mag. (February 1998).
[74] A. Gutierrez, W.E. Ryan, Performance of adaptive Volterra equalizers on non linear satellite channels, Proceedings of ICC'96, Seattle, USA, June 1995.
[75] I. Habib, Applications of neurocomputing in traffic management of ATM networks, IEEE Proc. 84 (10) (October 1996) 1430–1441.
[76] S. Haykin, Adaptive Filter Theory, Prentice-Hall, Englewood Cliffs, NJ, 1986.
[77] S. Haykin, Neural Networks: a Comprehensive Foundation, IEEE Press, New York, 1994.
[78] S. Haykin, L. Li, 16kb/s adaptive differential pulse code modulation of speech, in Proceedings of International Workshop on Applications of Neural Networks to Telecommunications, LEA Publishers, 1993, pp. 132–138.
[79] S. Haykin, L. Li, Non linear adaptive prediction of non-stationary signals, IEEE Trans. Signal Process. 45 (1995) 526–535.
[80] S. Haykin, X. Li, Detection of signals in chaos, IEEE Proc. 83 (1) (January 1995) 95–122.
[81] S. Haykin, P. Yee, E. Derbez, Optimum non linear filtering, IEEE Trans. Signal Process. 45 (11) (November 1997).
[82] J. Hérault, C. Jutten, Réseaux de Neurones et Traitement du Signal, Hermès, Paris, 1994.
[83] T. Heskes, B. Kappen, On-line learning processes in artificial neural networks, in: J.G. Taylor (Ed.), Mathematical Approaches to Neural Networks, North-Holland, Amsterdam, 1993, pp. 199–233.
[84] J.H. Holland, Adaptation in Natural And Artificial Systems, MIT Press, Cambridge, MA, 1993.
[85] A. Holst, A. Lansner, Diagnosis of technical equipment using a bayesian neural network, Proceedings of International Workshop on the Applications on Neural Networks to Telecommunications, LEA Publishers, 1993, pp. 147–153.
[86] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sci. 79 (April 1982) 2554–2558.
[87] K. Hornik et al., Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989) 359–366.
[88] A. Hussain, J. Soraghan, T. Durrani, A new adaptive functional-link neural network-based DFE for overcoming co-channel interference, IEEE Trans. Commun. 45 (11) (November 1997) 1358–1362.
[89] J. Hwang, S. Kung, M. Niranjan, J. Principe, The past, present, and future of neural networks for signal processing, IEEE Signal Process. Mag. (November 1997) 28–48.
[90] M. Ibnkahla, On the influence of the number of layers on the performance and convergence behavior of the back propagation algorithm, Proceedings of ICASSP'97, Munich, Germany, April 1997.
[91] M. Ibnkahla, Neural network analysis and compensation of non linearities in digital satellite transmissions, Internal Report, ECE QUK, December 1999.
[92] M. Ibnkahla, N.J. Bershad, J. Sombrin, F. Castanié, Neural network modeling and identification of non-linear

channels with memory: Algorithms, applications, and analytic models, IEEE Trans. Signal Process. 46 (5) (May 1998) 1208–1220.
[93] M. Ibnkahla, M. Ghogho, F. Castanié, Neural network modeling and identification of non linear rapidly fading channels, Proceedings of COST # 254, Toulouse, France, July 1997.
[94] M. Ibnkahla, J. Sombrin, F. Castanié, N.J. Bershad, Neural networks for modeling non-linear memoryless communication channels, IEEE Trans. Commun. 45 (7) (July 1997) 768–771.
[95] IEICE Trans. Electron. (Special Issue) Low Distortion Technol. Microwave Dev. Circuits E-82-C(5) (May 1999).
[96] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Computation 3 (1) (1991) 79–87.
[97] M.I. Jordan, R.A. Jacobs, Hierarchical mixture of experts and the EM algorithm, Neural Computation 6 (1994) 181–214.
[98] A. Juditsky, H. Hjalmarsson, A. Benveniste, B. Deylon, L. Ljung, J. Sjoberg, Q. Zhang, Nonlinear black-box modeling in system identification: mathematical foundations, Automatica 31 (12) (1995) 1725–1750.
[99] B. Juang, S. Katagiri, Discriminative learning for minimum error classification, IEEE Trans. Signal Process. 40 (December 1992).
[100] G. Karam, H. Sari, Analysis of predistortion, equalization and ISI cancellation techniques in digital radio systems with nonlinear transmit amplifiers, IEEE Trans. Commun. 37 (12) (December 1989).
[101] G. Karam, H. Sari, A data predistortio technique with memory for QAM radio systems, IEEE Trans. Commun. 39 (2) (February 1991) 336–344.
[102] N. Karayianinis, P. Pai, Fuzzy algorithms for learning vector quantization, IEEE Trans. Neural Networks 7 (5) (September 1996) 1196–1211.
[103] J. Karhunen, A. Hyvarinen, R. Vigario, J. Hurri, E.E. Oja, Applications of neural blind separation to signal and image processing, Proceedings of ICASSP'97, Munich, Germany, April 1997, pp. 131–134.
[104] G. Kechriotis, E. Manolakos, Hopfiel neural network implementation of the optimal CDMA multiuser detector, IEEE Trans. Neural Networks 7 (1) (January 1996) 131–141.
[105] G. Kechriotis, E. Manolakos, A hybrid digital signal processing-neural network CDMA multiuser detection scheme, IEEE Trans. Circuits Systems-II 43 (2) (February 1996) 96–104.
[106] G. Kechriotis, E. Zervas, E. Manolakos, Using recurrent neural networks for adaptive communication channel equalization, IEEE Trans. Neural Networks 5 (2) (March 1994) 267–278.
[107] R. Kil, Y. Oh, Vector quantization based on genetic algorithm, Proceedings of WCNN'95, Washington, DC, 1995 pp. I-778–I782.
[108] W.R. Kirkland, D.P. Taylor, Neural network channel equalization, in: B. Yuhas, N. Ansari (Eds.), Neural Networks in Telecommunications, Kluwer Academic Publishers, Dordrecht, 1994, pp. 143–172.
[109] S. Kirkpatrick, C.D. Gelatt Jr., M.P. Vecchi, Optimization by simulated annealing, Science 220 (4598) (May 1983) 671–680.
[110] T. Kohonen, Self-Organization And Associative Memory, 3rd Edition, Springer, Berlin, 1989.
[111] T. Kohonen, The self-organizing map, IEEE Proc. 78 (9) (September 1990) 1464–1480.
[112] T. Kohonen, E. Oja, O. Simula, A. Visa, J. Kangas, Engineering applications of the self-organizing map, IEEE Proc. (October 1996) 1358–1384.
[113] B. Kosko, Neural Networks For Signal Processing, Prentice-Hall Editions, Englewood Cliffs, NJ, 1992.
[114] J.R. Koza, Genetic Programming, On The Programming of Computers By Means of Natural Selection, MIT Press, Cambridge, MA, 1993.
[115] J.R. Koza, Genetic Programming II, Automatic Discovery of Reusable Programs, MIT Press, Cambridge, MA, 1994.
[116] C. Kuan, K. Hornik, Convergence of learning algorithms with constant learning rates, IEEE Trans. Neural Networks 2 (5) (September 1991).
[117] S.Y. Kung, Digital Neural Networks, PTR Prentice-Hall, Englewood Cliffs, NJ, 1993.
[118] S.Y. Kung, K. Diamantaras, J. Taur, Adaptive principal component extraction (APEX) and applications, IEEE Trans. Signal Process. 42 (May 1994) 1202–1217.
[119] J. Lampinen, E. Oja, Distortion tolerant pattern recognition based on self-organizing feature extraction, IEEE Trans. Neural Networks 6 (1995) 539–547.
[120] R. Lancina, S. Tubaro, Adaptive vector quantization for picture coding using neural networks, IEEE Trans. Commun. 43 (2/3/4) (February/March/April 1995) 534–544.
[121] C.G. Lau (Ed.), Neural Networks: Theoretical Foundations and Analysis, IEEE Press, New York, 1992.
[122] G. Lazzarin, S. Pupolin, A. Sarti, Nonlinearity compensation in digital radio systems, IEEE Trans. Commun. 42 (2/3/4) (February/March/April 1994).
[123] C. Leung, L. Chan, Transmission of vector quantized data over a noisy channel, IEEE Trans. Neural Networks 8 (3) (May 1997) 582–589.
[124] H. Leung, S. Haykin, The complex back propagation algorithm, IEEE Trans. Signal Process. 39 (9) (September 1991) 2101–2104.
[125] E. Levin et al., A statistical approach to learning and generalization in layered neural networks, Proc. IEEE 78 (10) (October 1990) 1568–1574.
[126] R.P. Lippmann, An intoduction to computing with neural nets, IEEE ASSP Mag. (April 1987) 4–22.
[127] N. Lo, H. Hafez, Neural network channel equalization, Proceedings of International Joint Conference on Neural Networks, IJCNN'92, Vol. 2, 1992, pp. 981–986.
[128] R. Lupas, S. Verdù, Linear multiuser detectors for synchronous CDMA channels, IEEE Trans. Inform. Theory 35 (1) (January 1989) 123–136.

[129] O. Macchi, E. Moreau, Self-adaptive source separation – Part I: Convergence analysis of a direct linear network controlled by the Hérault-Jutten algorithm, IEEE Trans. Signal Process. 45 (April 1997) 918–926.
[130] Y. Maki, K. Loparo, A neural network approach to fault detection and diagnosis in industrial processes, IEEE Trans. Control Systems Technol. 5 (6) (November 1997).
[131] C.N. Manikopoulos, Neural network approach to DPCM system design for image coding, IEE Proc. – I 139 (October 1992) 501–507.
[132] S. Marcos et al., A unified framework for gradient algorithms used for filter adaptation and neural network training, Int. J. Circuit Theory Appl. 20 (1992) 159–200.
[133] V. Marmarelis, X. Zhao, Volterra models and three-layer perceptrons, IEEE Trans. Neural Networks 8 (6) (November 1997) 1421–1433.
[134] T. Martinetz, K. Schulten, Topology representing networks, Neural Networks 7 (3) (1994) 507–522.
[135] T. Matsuoka et al., Compensation of non linear distortion during transmission based on adaptive predistortion method, IEICE Trans. Electron. E-80-C (6) (June 1997).
[136] U. Mitra, V. Poor, Adaptive receiver algorithms for near-far resistant CDMA, IEEE Trans. Commun. 43 (2/3/4) (February/March/April 1995) 1713–1724.
[137] D. Modha, R. Hecht-Nielsen, Multilayer functionals, in: J.G. Taylor (Ed.), Mathematical Approaches to Neural Networks, North-Holland, Amsterdam, 1993, pp. 235–2260.
[138] E. Moreau, O. Macchi, Self-adaptive source separation – Part II: comparison of the direct, feedback, and mixed linear network, IEEE Trans. Signal Process. 46 (January 1998) 39–50.
[139] B. Mulgrew, Applying radial basis functions, IEEE Signal Process. Mag. 13 (2) (March 1996) 50–65.
[140] S. Nair, J. Moon, Data storage channel equalization using neural networks, IEEE Trans. Neural Networks 8 (5) (September 1997) 1037–1048.
[141] S. Nair, J. Moon, A theoretical study of linear and nonlinear equalization of nonlinear magnetic storage channels, IEEE Trans. Neural Networks 8 (5) (September 1997) 1106–1118.
[142] A. Nandi, E. Azzouz, Modulation recognition using artificial neural networks, Signal Processing 56 (1997) 165–175.
[143] K.S. Narenda, Neural networks for control, IEEE Proc. 84 (10) (October 1996) 1385–1406.
[144] K.S. Narenda, K. Parthasarathy, Identification and control of nonlinear systems using neural networks, IEEE Trans. Neural Networks 1 (4) (1990) 4–27.
[145] N.M. Nasrabadi, R.A. King, Image coding using vector quantization: a review, IEEE Trans. Commun. 36 (August 1988) 957–971.
[146] T. Nitta, An extension of the back-propagation algorithm to complex numbers, Neural Networks 10 (8) (1997) 1391–1415.
[147] E. Oja, Subspace Methods of Pattern Recognition, Research Studies Press, Letchworth, UK, 1983.
[148] I. Ortuno, M. Ortuno, J. Delgado, Error correcting neural networks for channels with gaussian noise, Proceedings of International Joint Conference on Neural Networks, IJCNN'92, Vol. 4, 1992, pp. 295–300.
[149] R. Parisi, E. Di Claudio, G. Orlandi, B. Rao, Fast adaptive digital equalization by recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (November 1997) 2731–2739.
[150] R. Parker, M. Tummala, Identification of Volterra systems with a polynomial neural network, ICASSP'92, Vol. 4, 1992, pp. 661–564.
[151] B. Paya, I. Esat, Artificial neural network based fault diagnosis of rotating machinery using wavelet transforms as preprocessor, Mech. Systems Signal Process. 11 (5) (1997) 751–765.
[152] M. Peng, C. Nikias, J. Proakis, Adaptive equalization with neural networks: new multi-layer perceptron structures and their evaluation, Proceedings of International Joint Conference on Neural Networks, Vol. 2, 1992, pp. 301–304.
[153] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (9) (September 1990) 1481–1497.
[154] S. Puechmorel, Réseaux de Neurones et Optimisation Globale en Analyse Temps-Fréquence, Ph. D. Thesis, National Polytechnics Institue of Toulouse, France, 1994.
[155] S. Puechmorel, M. Ibnkahla, Operator-valued neural networks, Proceedings of World Congress on Neural Networks, WCNN'95, Washington, DC, July 1995.
[156] S.U. Qureshi, Adaptive equalization, IEEE Proc. 73 (9) (September 1985).
[157] H. Ritter, Martinetz, K. Schulten, Neural Computation and Self-organizing Maps, an Introduction, Addison-Wesley Publishing Company, New York, 1992.
[158] H. Ritter, K. Schulten, Convergence properties of Kohonen's topology conserving maps: fluctuations, stability, and dimension selection, Biol. Cybernet. 60 (1988) 59–71.
[159] J. Rissanen, Universal coding, information, prediction, and estimation using predictive MDL principle, IEEE Trans. Inform. Theory 30 (July 1984) 629–636.
[160] T. Robinson, M. Bodruzzaman, Adaptive filtering using Youla parametrization and neural networks, Proceedings of WCNN'95, Washington, DC, June 1995, pp. I-729–I734.
[161] D. Ruck, S. Rogers, M. Kabrisky, M. Oxley, B. Suter, The multi-layer perceptron as an approximation to a Bayes optimal discriminant function, IEEE Trans. Neural Networks 1 (4) (December 1990) 296–298.
[162] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge, MA, 1986, pp. 318–362.
[163] V. Sagar, G. Jacyna, H. Szu, Block-parallel decoding of convolutional codes using neural network decoders, Neurocomputing 6 (1994) 455–471.

[164] A. Saleh, Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers, IEEE Trans. Commun. Com-29 (11) (November 1981).
[165] I. Sandberg, L. Xiu, Network approximation of input-output maps and functionals, J. Circuits Systems Signal Process. 15 (6) (1996) 711–725.
[166] T. Sanger, Analysis of the two-dimensional receptive fields learned by the generalized hebbian algorithm in response to random input, Biol. Cybernet. 63 (1990) 221–228.
[167] A. Sarajedini, R. Hecht-Nielsen, The best of both words: casasent networks integrate multi-layer perceptrons and radial basis functions, International Joint Conference on Neural Networks, IJCNN'92, Vol. 3, 1992, pp. 905–910.
[168] B. Sheu, S. Bang, R. Chang, A cellular neural network with optimized performance for wireless communication receivers, Proceedings of WCNN'95, Washington, DC, June 1995, pp. II-660-II-664.
[169] J.J. Shynk, Performance surfaces of a single-layer perceptron, IEEE Trans. Neural Networks 1 (3) (September 1990).
[170] J.J. Shynk, N.J. Bershad, Steady state analysis of a single-layer perceptron based on a system identification model with bias terms, IEEE Trans. Circuits Systems 38 (9) (September 1991).
[171] J.J. Shynk, S. Roy, Convergence properties and stationary points of a perceptron learning algorithm, Proc. IEEE 78 (10) (October 1990) 1599–1604.
[172] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Glorennec, H. Hjalmarsson, A. Juditsky, Nonlinear black-box modeling in system identification: a unified overview, Automatica 31 (12) (1995) 1691–1724.
[173] M. Soo Kim, C. Guest, Modification of backpropagation networks for complex-valued signal processing in frequency domain, International Joint Conference on Neural Networks, IJCNN'90, Vol. 3, 1990, pp. 27–31.
[174] B. Stiles, I. Sandberg, J. Ghosh, Complete memory structures for approximating nonlinear discrete-time mappings, IEEE Trans. Neural Networks 8 (6) (November 1997) 1397–1409.
[175] J. Sum, L.W. Chan, Fuzzy self-organizing map, WCNN'94, San Diego, CA, June 1994, pp. I-732-I-737.
[176] S. Tan, J. Vandewalle, On the design of feedforward neural networks for binary mappings, Neurocomputing 6 (1994) 565–582.
[177] A. Tarraf, I. Habib, T. Saadawi, Intelligent traffic control for ATM broadband networks, IEEE Commun. Mag. 33 (10) (October 1995) 76–82.
[178] J.G. Taylor (Ed.), Mathematical Approaches to Neural Networks, North-Holland, Amsterdam, 1993.
[179] J.G. Taylor, M.D. Plumbley, Information theory and neural networks, in: J.G. Taylor (Ed.), Mathematical Approaches to Neural Networks, North-Holland, Amsterdam, 1993, pp. 307–340.
[180] M. Thomas, M. Weidner, S. Durrani, Digital amplitude-phase keying with M-ary alphabets, IEEE Trans. Commun. 22 (2) (February 1974) 168–180.
[181] L. Tsoukalas, R. Uhrig, Fuzzy and Neural Approaches in Engineering, Wiley-Interscience, New York, 1997.
[182] M. Varanasi, B. Aazhang, Multistage detection in asynchronous code-division multiple access communications, IEEE Trans. Commun. 38 (April 1990) 509–519.
[183] S. Verdù, Minimum probability of error for asynchronous Gaussian multiple-access channels, IEEE Trans. Inform. Theory 32 (1) (January 1986) 85–96.
[184] E. Wan, Neural network classification: a Bayesian interpretation, IEEE Trans. Neural Networks 1 (4) (December 1990) 303–305.
[185] X. Wang, S. Wicker, An artificial neural net Viterbi decoder, IEEE Trans. Commun. 44 (2) (February 1996) 165–171.
[186] J. Watterson, An optimum multilayer perceptron neural receiver for signal detection, IEEE Trans. Neural Networks 1 (4) (December 1990) 298–300.
[187] Z. Wei, Y. Yinglin, Approximation property of multilayer neural network and its application in nonlinear simulation, IJCNN'91, Vol. 1, 1992, pp. 171–176.
[188] P. Werbos, Backpropagation through time: What it does and how to do it, IEEE Proc. 78 (10) (October 1990) 1550–1560.
[189] H. White, Learning in artificial neural networks: a statistical perspective, Neural Computation 1 (1989) 425–464.
[190] B. Widrow, Billelo, Non linear adaptive signal processing for inverse control, Proceedings of WCNN'94, San Diego, CA, June 1994, pp. III-3-13.
[191] B. Widrow, M. Lehr, 30 years of adaptive neural networks : perceptron, madaline, and back propagation, Proc. IEEE 78 (9) (September 1990) 1415–1442.
[192] B. Widrow, S. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1985.
[193] B. Widrow, R. Winter, Neural nets for adaptive filtering and adaptive pattern recognition, IEEE Comput. Mag. (March 1988) 25–39.
[194] R. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, Neural Computation 1 (1989) 270–280.
[195] Z. Xiang, G. Bi, Complex neural model with its applications to m-QAM data communications in the presence of co-channel interferences, IJCNN'92, Vol. 2, 1992, pp. 305–308.
[196] Z. Xiang, G. Bi, A new polynomial perceptron based 64QAM cellular mobile communications receivers, IEEE Trans. Signal Process. 43 (12) (December 1995) 3094–3098.
[197] J. Xuan, T. Adali, X. Liu, Information geometry of maximum partial likelihood estimation for channel equalization, Proceedings of ICASSP'96, 1996, 3533–3536.
[198] H. Yang, S.I. Amari, Adaptive on-line learning algorithms for blind separation – maximum entropy and minimal mutual information, Neural Comutation 8 (10) (October 1998).
[199] J. Yang, C. Chen, J. Lee, Neural networks for maximum likelihood error correcting systems, Proceedings of IJCNN'90, pp. I-493–I-498.

[200] L. Yin, J. Astola, Y. Neuvo, A new class of nonlinear filters-Neural filters, IEEE Trans. Signal Process. 41 (3) (March 1993) 1201–1222.

[201] H. Yu, S. Bang, An improved time seies prediction by applying the layer-by-layer learning method to FIR neural networks, Neural Networks 10 (9) (1997) 1717–1729.

[202] Q. Zhang, Using wavelet network in nonparametric estimation, IEEE Trans. Neural Networks 8 (2) (March 1997) 227–236.

[203] Q. Zhang, A. Benveniste, Wavelet networks, IEEE Trans. on Neural Networks 3 (6) (November 1992).

[204] Z.Z. Zhang, N. Ansari, Structure and properties of generalized adaptive neural filters for signal enhancement, IEEE Trans. Neural Networks 7 (4) (July 1996).

[205] M. Zoyo, Conception et Réalisation d'Amplificateurs de Puissance Microondes à l'Etat Solide et à Fort Rendement Pour des Applications Spatiales en Bande S et en Bande X, Ph.D. Thesis, Université Paul Sabatier, Toulouse, France, 1996.