

A Net with Complex Weights

Boris Igel'nik, *Senior Member, IEEE*, Massood Tabib-Azar, *Senior Member, IEEE*, and Steven R. LeClair

Abstract—In this article a new neural-network architecture suitable for learning and generalization is discussed and developed. Although similar to the radial basis function (RBF) net, our computational model called the net with complex weights (CWN) has demonstrated a considerable gain in performance and efficiency in number of applications compared to RBF net. Its better performance in classification tasks is explained by the cross-product terms in internal representation of its basis function introduced parsimoniously. Implementation of CWN by the ensemble approach is described. A number of examples, solved using CWN and other networks, are used to illustrate the desirable characteristics of CWN.

Index Terms—Adaptive stochastic optimization, basis functions, complex weights, ensemble of nets, recursive linear regression.

I. INTRODUCTION

THERE are a number of adaptive computational architectures (let us call them nets) for approximating multivariate mappings with application in regression and classification tasks. Some examples of such architectures are nonlinear perceptrons [1], [2], radial basis functions (RBFs) [3], projection pursuit nets [4], [5], hinging hyperplanes [6], probabilistic nets [7], random nets [8], high-order nets [9], and wavelets [10], to name a few. The mathematical model implemented in some of these nets can be expressed in the following form:

$$\tilde{f}(x) = \sum_{n=1}^N a_n g \left[\sum_{i=1}^d w_{ni} \psi(x_i, c_{ni}) \right] \quad (1)$$

where $x = (x_1, \dots, x_d) \in D \subset \mathbb{R}^d$, D is a closed bounded set in \mathbb{R}^d , taken as the standard unit cube $D = \{x = (x_1, \dots, x_d) | 0 \leq x_1 \leq 1, \dots, 0 \leq x_d \leq 1\}$, without loss of generality. The computational (and analytical) model \tilde{f} approximates an unknown function f , defined to be continuous on D . The parameter $a = (a_1, \dots, a_N)$, the *external parameter*, and the parameters $w_n = (w_{n1}, \dots, w_{nd})$ and $c_n = (c_{n1}, \dots, c_{nd})$, the *internal parameters*, are adjustable on the data, as well as the number of nodes N . The univariate function g is called the *external* or *activation function*. The univariate functions ψ_i , $i = 1, \dots, d$ are called the *internal functions*. They are the same for all functions f from the class of functions, defined and continuous on D , in approximations such as nonlinear perceptrons or RBF nets. The internal functions are adjustable on the data in projection

pursuit. High-order networks use not only sum of univariate functions in internal representation but terms depending on two, three, or more input variables.

The multitude of computational models reflects the following fact: none of these architectures can be uniformly better than all other models. For example, use of homogeneous basis functions inevitably leads to inefficiency for some applications. It should be noticed as well that the Kolmogorov's superposition theorem, which gives the most theoretically efficient representation of a multivariate continuous function through superpositions and sums of univariate functions [11], requires internal function dependent on data.

Having these facts in mind, we have suggested and successfully applied the ensemble approach (EA) for learning and generalization [12], [13] for some tasks. The EA uses a mathematical model which is more general than (1)

$$\tilde{f}(x) = \sum_{n=1}^N a_n g[\varphi(x, w_n, c_n)] \quad (2)$$

where

- g univariate external function;
- φ multivariate internal representation;
- a, w_n , and c_n adjustable parameters.

One of the features of the EA is that it has a finite but expandable set of external functions (currently containing logistic function, hyperbolic tangent, Gaussian, second derivative of Gaussian, thin plate function and cube) and a set of internal representations (currently nonlinear perceptron, RBF, and product of univariate neurons). This feature gives an opportunity to adjust not only the parameters but the type of basis functions for a particular application. Currently, the basis function is discretely and manually adjusted, but we are working on an automatic and continuous adjustment mode as well.

Recently, we have suggested and tested, both on mathematical examples and applications, a new approximation model called the net with complex weights (CWN), which has some advantageous characteristics in complex applications. The CWN computational model is of the following form:

$$\tilde{f}(x) = a_0 + \sum_{n=1}^N a_n g[\langle w_n, x - c_n \rangle \cdot \langle \bar{w}_n, x - c_n \rangle] \quad (3)$$

where

- the parameters a_n real numbers;
- c_n real vectors;
- the parameters w_n complex vectors.

In (3) \bar{w}_n stands for the complex conjugate of w_n and $\langle w_n, x - c_n \rangle$ is the inner product of two vectors w_n and $x - c_n$. Unlike the neural, or RBF, or Kolmogorov's nets [14], [15], the internal representation in a basis function is not a weighted sum

Manuscript received January 4, 1999; revised August 23, 1999.

B. Igel'nik is with Pegasus Technologies, Incorporated, Mentor, OH 44060 USA.

M. Tabib-Azar is with the Electrical Engineering and Computer Science Department, Case Western Reserve University, Cleveland, OH 44106 USA.

S. R. LeClair is with the Material Directorate, Wright Laboratory, WL/MLIM 2977 P St., Wright-Patterson AFB, OH 45433-7746 USA.

Publisher Item Identifier S 1045-9227(01)02051-3.

of univariate functions, but constitutes a quadratic function of d variables with cross-product terms. These high-order terms are introduced in a parsimonious way. Instead of $d(d + 1)/2 + 1$ parameters for the general quadratic function, only $2d + 1$ parameters are used.

Our motivation for use of CWN is given in Sections II and Appendix. In Section II we use the benchmark XOR problem [16] to demonstrate the advantage of CWN over RBF and some other approximation models. We present the theorem on universal approximation capability of CWN in Appendix. Implementation of CWN by EA is described in Section III. Mathematical and application examples where CWN had superiority over RBF net are presented in Section IV. Conclusion and future work are given in Section V.

The use of complex parameters in neural networks is described in [17]–[20]. Unlike our network, these works make use of complex analytic and nonanalytic activation functions and an architecture of nonlinear multilayer perceptron. In addition, their method of training is different from the EA. However, they obtained similar results. The universal approximation capability of nets with complex parameters, savings in computation time, and improved efficiency, were demonstrated in different applications as compared with nets with real parameters.

The initial incentive for considering CWN was its possible implementation with quantum devices. Currently, the quantum device and quantum integrated circuit technologies are not sufficiently developed to enable implementation of the CWN architecture. Thus, we set out to show the advantages of the CWN algorithm in certain complex computational tasks.

II. XOR PROBLEM

In this section we compare the efficiency of a single-node CWN with the efficiency of any other single-node net without cross-product terms in the internal representation, in solving the benchmark XOR problem. These other nets are subdivided into two subcases of nets: those with fixed and with adjustable internal functions. We show that the efficiency of the CWN, measured in terms of required adjustable parameters, is superior to the efficiency of other nets with comparable size.

The XOR problem is to find a curve $f(x, y) = 0$ that separates the points $A(0, 0)$, $C(1, 1)$ from the points $B(1, 0)$ and $D(0, 1)$ as shown in Fig. 1. That means that there exists a model $z = f(x, y)$ such that the points A and C are on the one side of the curve $f(x, y) = 0$ and the points B and D are on another side of the curve. We can prove the following proposition.

Proposition 1: Any net of the form

$$f(x, y) = g(\psi_1(x) + \psi_2(y)) \quad (4)$$

where g is a monotonic fixed univariate function, ψ_1, ψ_2 are arbitrary fixed-shape univariate functions can not solve the XOR problem.

Proof: By contradiction suppose that a net of the form (4) can solve such problem. Denoting $x_0 = \psi_1(0)$, $x_1 = \psi_1(1)$, $y_0 = \psi_2(0)$, $y_1 = \psi_2(1)$, adding, if necessary, some constants

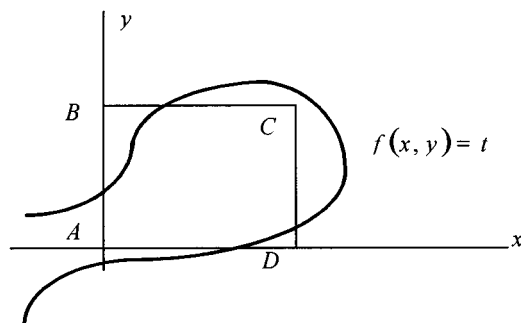


Fig. 1. Geometric illustration of the XOR problem.

to the functions ψ_1, ψ_2 , and using monotonicity of the function g , one obtains

$$\begin{cases} x_0 + y_0 > 0 \\ x_1 + y_1 > 0 \\ x_0 + y_1 < 0 \\ x_1 + y_0 < 0. \end{cases}$$

Summing the first and second, and then third and fourth inequalities yields the contradiction

$$\begin{cases} x_0 + y_0 + x_1 + y_1 > 0 \\ x_0 + y_1 + x_1 + y_0 < 0. \end{cases}$$

Therefore, without using the cross-product of the variables x and y in the fixed internal representation of the basis function, it is impossible to solve the XOR problem with one basis function and a fixed internal representation.

Consider the case where the function g is fixed but internal representation is adaptive. That means that we can change the shape of the functions ψ_1 and ψ_2 depending on data. For this case we prove the following proposition.

Proposition 2: There exists a net f of the form (4) with fixed monotonic univariate function g and adaptive-shape differentiable functions ψ_1, ψ_2 , formed from polynomials, that solves the XOR problem. Any such net should have at least eight parameters.

Proof: We give the explicit construction of such a net. The construction is in Fig. 2. First we construct a line separating the points A, B , and D from the point C in Fig. 2 and choose $\alpha > 1$, $\beta > 1$. The line EF , with

$$\psi_1(x) + \psi_2(y) = 0, \quad 0 \leq x \leq \alpha \quad (5)$$

where

$$\psi_1(x) = x/\alpha - 0.5, \quad \psi_2(y) = y/\beta - 0.5 \quad (6)$$

is a solution of the problem if $\beta - 1 < \beta/\alpha$. We then construct two parabolas which, together with line EF , make the final separation of A and C from B and D . First, consider the parabola FGH with the equation $x = a_0 + a_1y + a_2y^2$ and choose $\gamma < 0$ and the coefficients a_0, a_1, a_2 so that they satisfy the conditions

$$\begin{cases} x(0) = a_0 = \alpha \\ -0.5a_1/a_2 = \gamma \\ \frac{dy}{dx}(\alpha) = -\frac{1}{a_1} = -\frac{\beta}{\alpha}. \end{cases} \quad (7)$$

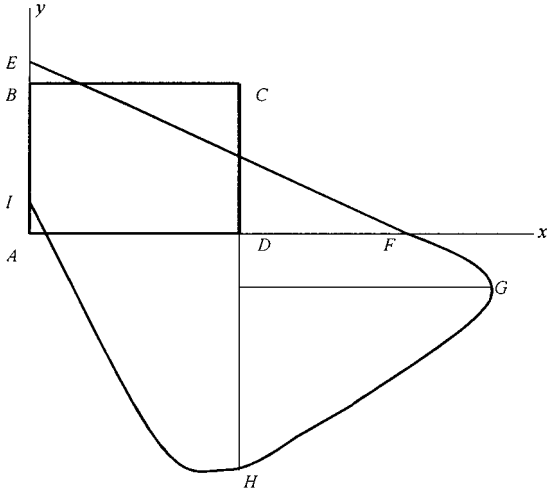


Fig. 2. Geometric illustration of the solution of the XOR problem by a net with adaptive-shape internal representation.

The conditions in (7) guarantee that the line and the parabola are connected continuously and smoothly and that the point G in Fig. 2 has the ordinate equal to γ . Thus, the equation of the parabola FGH can be written as

$$\psi_1(x) + \psi_2(y) = 0, \quad \gamma - \sqrt{\gamma^2 - \beta\gamma(1 - 2/\alpha)} \leq y \leq 0 \quad (8)$$

where

$$\psi_1(x) = 2\beta\gamma x/\alpha, \quad \psi_2(y) = -(y^2 - 2\gamma y + \beta\gamma). \quad (9)$$

Next, we choose δ , $0 < \delta < 1$ and the parabola HI with the equation $y = b_0 + b_1x + b_2x^2$ so, that the points $F(0, \gamma - \sqrt{\gamma^2 - \beta\gamma(1 - 2/\alpha)})$ and $I(0, \delta)$ can be continuously and smoothly connected by this parabola. A simple calculation yields

$$\begin{aligned} b_0 &= \delta \\ b_1 &= 2(\gamma - \delta - \lambda) + \beta\gamma/\lambda \\ b_2 &= -\beta\gamma/\lambda - \gamma + \delta + \gamma \end{aligned} \quad (10)$$

where

$$\lambda = \sqrt{\gamma^2 - \beta\gamma(1 - 2/\alpha)}. \quad (11)$$

The equation of the parabola GI can be written in the form

$$\psi_1(x) + \psi_2(y) = 0, \quad 0 \leq x \leq 1 \quad (12)$$

$$\psi_1(x) = b_0 + b_1x + b_2x^2, \quad \psi_2(y) = -y. \quad (13)$$

It can be shown that using polynomial splines it is impossible to solve the XOR problem with one basis function and with less than eight parameters. As mentioned before the reason for the inability is the lack of cross-product terms.

The CWN has such cross-product terms. Considering again our benchmark example with the XOR problem we let

$$\begin{aligned} & [(x - 0.5) \cos \theta_1 + (y - 0.5) \cos \theta_2]_1^2 \\ & + [(x - 0.5) \sin \theta_1 + (y - 0.5) \sin \theta_2]^2 - a^2 = 0 \end{aligned} \quad (14)$$

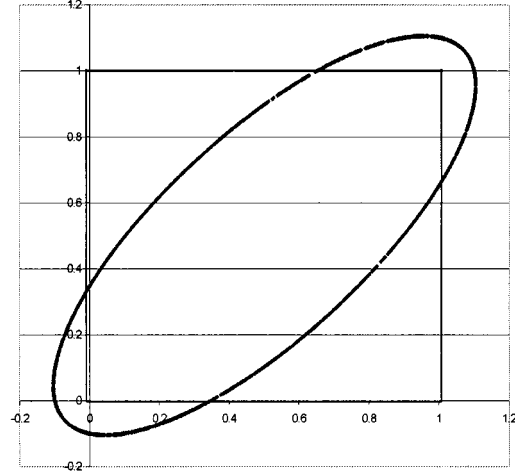


Fig. 3. Geometric illustration of the XOR problem solution by CWN.

be the equation of separating curve. The equation (14) can be easily derived from (3) for CWN with one nonlinear basis function using Euler's formula for complex weights. Transforming the variables x and y to new variables u and v by the turn of coordinate axes on the angle $\pi/4$

$$\begin{aligned} x - 0.5 &= u\sqrt{2}/2 - v\sqrt{2}/2 \\ y - 0.5 &= u\sqrt{2}/2 + v\sqrt{2}/2. \end{aligned} \quad (15)$$

and substituting (15) in (14), one obtains

$$2u^2 \cos^2 \frac{\theta_2 - \theta_1}{2} + 2v^2 \sin^2 \frac{\theta_2 - \theta_1}{2} - a^2 = 0$$

which is an ellipse's equation with axes parallel to coordinate axes. Therefore, in coordinates x and y , (5) also constitutes an equation of an ellipse with the angle between the axes of the ellipse and the coordinate axes equal $\pi/4$. This is shown in Fig. 3.

Substitution of the coordinates of the points A , B , C , and D in the left-hand side of (14) yields

$$\begin{cases} \cos^2(\theta_2 - \theta_1) - a^2 > 0 \\ \sin^2(\theta_2 - \theta_1) - a^2 < 0. \end{cases} \quad (16)$$

The simultaneous inequalities (16) are satisfied, for example, if

$$\begin{cases} a^2 = 1/2, \\ 0 < \theta_2 - \theta_1 < \pi/4. \end{cases}$$

Therefore, there exists a CWN with only one basis function, which solves the XOR problem, and that requires not more than four parameters, provided that the position of the ellipse's center is adjustable.

III. THE ENSEMBLE APPROACH (EA) AND CWN

1) *EA, Basic Ideas:* EA [12], [13] is a new method for training and generalization. We describe it for the general case since the peculiarities of the CWN architecture affect only a small block of the entire algorithm. Unlike the gradient methods of optimization for adjusting parameters of the model,

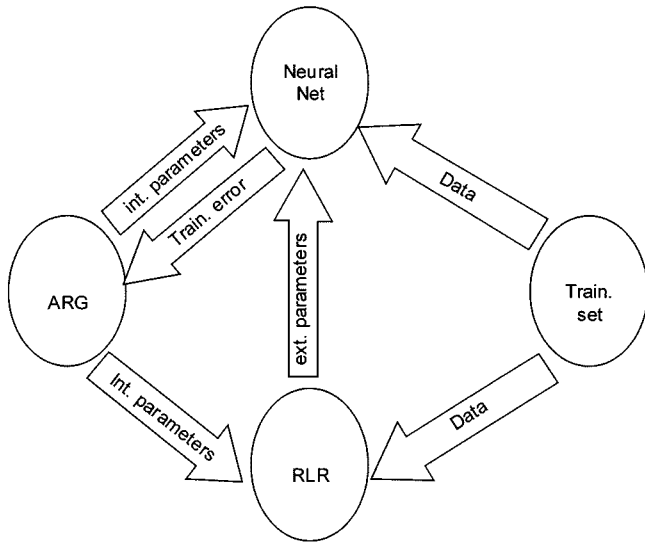


Fig. 4. Schematic illustration of EA.

the task of optimization is divided into two stages in EA: the recursive linear regression (RLR) [21] and the adaptive stochastic optimization (ASO) [12], [13]. The ensemble of nets with randomly chosen internal parameters is generated. For each net from the ensemble the values of the external parameters are optimized by RLR. The optimization of the internal parameters is made through the stochastic search over the ensemble. Thus, the two stages of optimization in EA, both global, are RLR and the stochastic search.

The simple stochastic search [22], as well as the simple quasistochastic search [23], are computationally slow procedures. That is why we have replaced them with the ASO. In ASO, the ensemble of internal parameters, generated by adaptive random generator (ARG), is divided into a number of portions. The distribution of each univariate component of an internal parameter in each portion is learned using current information about net's performance in the previous portions of the ensemble. Starting with the uniform distribution of the internal parameters in the first portion, we correct the distribution in subsequent portions on criteria of the minimal training error. This is shown in Fig. 4.

2) *Different Modes of EA:* EA can operate in sequential or nonsequential, local, or nonlocal modes. In sequential mode the training is performed one node at a time. It starts with a simplest net $f_0(x) = a_0$ and calculates the optimal value of a_0 . Suppose the optimal net (the best net in the ensemble) with $n - 1$ nodes has been built

$$f_{n-1}(x) = a_0 + \sum_{i=1}^{n-1} a_i g[\varphi(x, w_i, c_i)]. \quad (17)$$

In building the optimal net with n nodes

$$f_n(x) = a_0 + \sum_{i=1}^n a_i g[\varphi(x, w_i, c_i)] \quad (18)$$

the internal parameters $w_i, c_i, i = 1, \dots, n - 1$ retain their values from the previous step and only the internal parameters w_n and c_n are optimized. Thus, in this mode the ensemble contains the internal parameters of only one node. However, the external parameters a_0, a_1, \dots, a_{n-1} in (18) are different from those in (17) because they are recalculated by RLR. The use of RLR is especially efficient in this mode. The essential decrease of the size of searching space makes the sequential procedure faster than the nonsequential one. The same reason makes the sequential procedure theoretically less accurate. The justification for using sequential mode lies in the following result (proved for nonlinear perceptrons only) [24].

The upper bound of the training error can be achieved by an iterative sequence of approximations of the form

$$f_n(x) = \alpha_n f_{n-1}(x) + a_n g[\varphi(x, w_n, c_n)]. \quad (19)$$

Thus, even with more restrictions on the search space a near-optimal accuracy can be achieved. We, however, have made a practical correction to this theoretical result because of its asymptotic nature.

The nonsequential mode assumes learning the internal parameters of all nodes simultaneously. Theoretically speaking, it has an advantage over the sequential mode in accuracy. Practically, however, this advantage can be implemented only for nets with a small number of nodes. In particular, we have recently used the nonsequential mode for learning Lennard–Jones potentials in a multiatom system [25]. This problem can be solved using a relatively small number of nodes.

The nonlocal procedure is the standard one when one net, trained on the whole training set, is used for prediction for all patterns in the testing set. The local net builds separate net for each testing pattern by training only on a subset of a training set, consisting of K nearest neighbors to the testing pattern. In problems where time of testing is not crucial, the local mode may give more accurate results in prediction than the nonlocal one. In particular, we have used local net in the formers–nonformers problem with gained advantage. We, however, recommend use of local net with caution, because it destroys continuity of the mapping on the whole input space. Local net is not appropriate, as well, if one intends to make use of the net as an analytical model.

A. Different External and Internal Functions

As was mentioned in the introduction, EA can incorporate different types of external and internal functions. The basic types are traditional: nonlinear perceptron (P), RBF (R), and product of univariate neurons (U), as shown in Fig. 5, with an expandable list of external functions. The user of EA has an opportunity to manually choose the type of architecture. Recently, we added two new types of architecture: a net with complex coefficients (CWN) and a net for learning Lennard–Jones potentials (LJ). These architectures are shown in Fig. 6. CWN uses the same set of external functions as RBF, while LJ uses a special type of a node described later in this section.

In practice, we use CWN with one value of the parameter w for all nodes and all components of input vector. Therefore, it

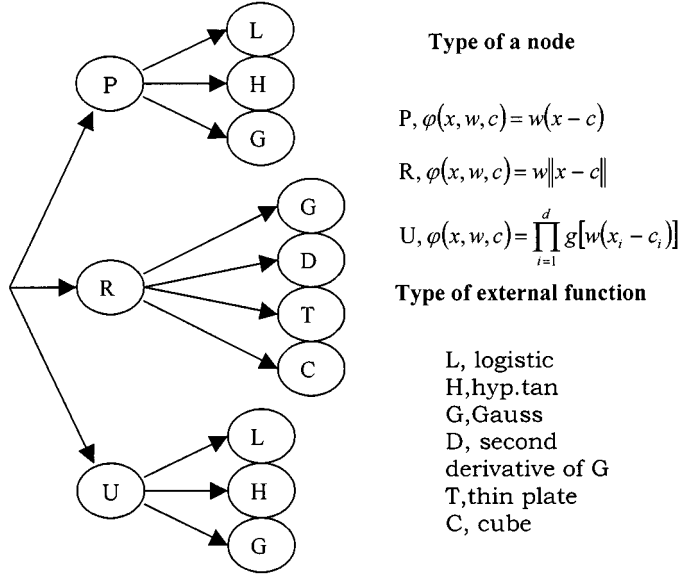


Fig. 5. Three basic architectures in EA.

has the following form:

$$\begin{aligned}
y &= \tilde{f}(x_1, \dots, x_d) \\
&= a_0 + \sum_{n=1}^N a_n g \\
&\quad \times \left[\sqrt{w^2 \sum_{j=1}^d e^{i\theta_{nj}} (x_j - c_{nj}) \sum_{j=1}^d e^{-i\theta_{nj}} (x_j - c_{nj})} \right]
\end{aligned} \tag{20}$$

where the model is in the coordinate form, all the parameters $a_0, a_n, c_{nj}, \theta_{nj}, w$ are real, w is the absolute value and θ_{nj} is the argument (phase) of a complex parameter. We assume that the input variables x_1, \dots, x_d are scaled so that $0 \leq x_1 \leq 1, \dots, 0 \leq x_d \leq 1$.

The values of the internal parameters are specified by the following inequalities:

$$\begin{aligned}
0 &\leq c_{nj} \leq 1 \\
0 &\leq \theta_{nj} \leq \pi - \Delta, \quad \pi + \Delta \leq \theta_{nj} \leq 2\pi
\end{aligned}$$

where $\Delta > 0$ is any number that is small compared to π (in practice $\Delta = 0.01$). The limitation on the choice of phase is explained in Appendix. We divide all data that are available for learning into two sets, the training set E_T and the generalization set $E_G, P = |E_T|$. The training set is used for adjusting the parameters $a_0, a_n, c_{nj}, \theta_{nj}$ on the criteria of the minimal training error, while the testing set is used for determining the optimal number of nodes N in sequential mode. The parameter w can be adjusted manually.

The LJ net was especially built for a solution of the following problem. The energy of interaction E between two atoms with

the distance r between them can be described through the Lennard–Jones potentials as

$$E = a_1 r^{-c_1} + a_2 r^{-c_2}.$$

For this simple system, the values of the parameters a_1, a_2, c_1 , and c_2 are known. For those values of the parameters the system of two atoms has one stable state and, therefore, the energy as a function of distance has one minimum. We considered the system with many atoms of two types, α and β , with one and more than one stable states. For simplicity of notations, we consider here only the system with one stable state. A net can describe the energy of this system as

$$\begin{aligned}
\tilde{f}(r, c) &= a_1 \sum_{i=1}^{M_{\alpha\beta}} r_i^{-c_1} + a_2 \sum_{i=1}^{M_{\alpha\beta}} r_i^{-c_2} \\
&+ a_3 \sum_{i=M_{\alpha\beta}+1}^{M_{\alpha\beta}+M_{\alpha}} r_i^{-c_3} + a_4 \sum_{i=M_{\alpha\beta}+1}^{M_{\alpha\beta}+M_{\alpha}} r_i^{-c_4} \\
&+ a_5 \sum_{i=M_{\alpha\beta}+M_{\alpha}+1}^{M_{\alpha\beta}+M_{\alpha}+M_{\beta}} r_i^{-c_5} + a_6 \sum_{i=M_{\alpha\beta}+M_{\alpha}+1}^{M_{\alpha\beta}+M_{\alpha}+M_{\beta}} r_i^{-c_6}
\end{aligned}$$

where $M_{\alpha\beta}, M_{\alpha}, M_{\beta}$ are the number of pairs of type $\alpha\beta, \alpha, \beta$ respectively, r_i is the distance between two atoms from the i th pair. The training data is a set of vectors $(r_1, r_2, \dots, r_M, E)$ and the task is to evaluate the parameters $a_i, c_i, i = 1, \dots, 6$. The specific of this task is that, although the number of inputs can be large, the number of nodes is limited to 6. This circumstance allowed using nonsequential mode with the advantage in accuracy. In general case, when the number of stable states is not known, the number of nodes is not limited. Therefore, the nonsequential mode may lose that advantage.

The two major stages of EA, the recursive linear regression and the adaptive stochastic optimization are described below specifically for CWN, although these stages are the same for any node architecture used in EA.

B. Recursive Linear Regression (RLR)

For evaluation of the training error the external parameters $a = (a_0, a_1, \dots, a_n)$ should be calculated. This is done by recursive linear regression. Let $c_{j \text{ opt}} = (c_{j1 \text{ opt}}, \dots, c_{jd \text{ opt}})$, $\theta_{j \text{ opt}} = (\theta_{j1 \text{ opt}}, \dots, \theta_{jd \text{ opt}})$ be the optimal internal parameters of the j th node, and $c_j = (c_{j1}, \dots, c_{jd})$, $\theta_j = (\theta_{j1}, \dots, \theta_{jd})$ be the internal parameters of a member of the ensemble for j th node

$$p_{ij} = g_j(x_i, c_{j \text{ opt}}, \theta_{j \text{ opt}}), \quad p_{i0} = 1, \quad p_{in} = g_n(x_i, c_n, \theta_n),$$

$$i = 1, \dots, P, \quad j = 1, \dots, n - 1$$

$$p_j = (p_{1j}, \dots, p_{Pj})^T, \quad j = 0, 1, \dots, n, \quad P_n = [p_0, \dots, p_n]$$

$$y = (y_1, \dots, y_P)^T, \quad E_T = \{(x_i, y_i), i = 1, \dots, P\}.$$

Denote $P_{n+}, n = 0, 1, \dots, N$ the $(n+1) \times P$ matrix, pseudoinverse to $P \times (n+1)$ matrix P_n . Then, P_{n+} is calculated

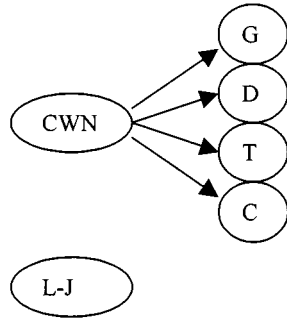


Fig. 6. Additional architectures in EA.

recursively by the following formulas:

$$P_{0+} = (1/P, \dots, 1/P) \quad (21)$$

$$P_{n+} = \left[\frac{P_{n-1,+}(I - p_n \nu_n^T)}{\nu_n^T} \right], \quad 1 \leq n \quad (22)$$

$$\nu_n = \frac{(I - P_{n-1}P_{n-1,+})p_n}{\|(I - P_{n-1}P_{n-1,+})p_n\|^2} \quad (23)$$

where I is $P \times P$ unit matrix. It is assumed that the quantity $\|(I - P_{n-1}P_{n-1,+})p_n\|$ satisfies the inequality

$$\|(I - P_{n-1}P_{n-1,+})p_n\| > \varepsilon \quad (24)$$

where $\varepsilon > 0$ is a small, positive number. Since vector $P_{n-1}P_{n-1,+}p_n$ is the orthogonal projection of the vector p_n on the linear subspace spanned by the vectors p_0, p_1, \dots, p_{n-1} , the vector $(I - P_{n-1}P_{n-1,+})p_n$ is the component of p_n , perpendicular to $\text{span}(p_0, p_1, \dots, p_{n-1})$, as shown in Fig. 7. Therefore, the condition (24) means that the basis functions, too close to that of a linear combination of the previously chosen basis functions, are thrown away from the ensemble.

Formula (22) also has a rather simple geometric illustration, shown in Fig. 8. Indeed, multiplying matrices $P_n = [P_{n-1}, p_n]$ and P_{n+} , given by (22), in the block form, and then multiplying the result by y , one obtains

$$P_n P_{n+} y = P_{n-1} P_{n-1,+} y + (\nu_n^T y) (I - P_{n-1} P_{n-1,+}) p_n. \quad (25)$$

Equation (25) says that the projection of vector y on $\text{span}(p_0, p_1, \dots, p_n)$, which is the left side of (25), equals to the projection of y on the $\text{span}(p_0, p_1, \dots, p_{n-1})$, which is $P_{n-1} P_{n-1,+} y$, plus the vector, collinear with the vector $(I - P_{n-1} P_{n-1,+}) p_n$. In Fig. 9, in the case $n = 1$, the $\text{span}(p_0, p_1, \dots, p_{n-1})$ coincides with the line parallel to p_0 .

Finally, the vector a of optimal external parameters is calculated as

$$a = P_{n+} y. \quad (26)$$

C. Adaptive Stochastic Optimization (ASO)

The adaptive stochastic optimization is used to select the values of the internal parameters which yield the best net in

Type of a node

$$\text{CWN}, \varphi(x, w, c, \theta) = w \sqrt{\frac{\sum_{j=1}^d e^{i\theta_j} (x_j - c_j)}{\sum_{j=1}^d e^{-i\theta_j} (x_j - c_j)}}$$

$$\text{L-J}, \varphi(r, c, M) = \sum_{i=1}^M r^{-c}$$

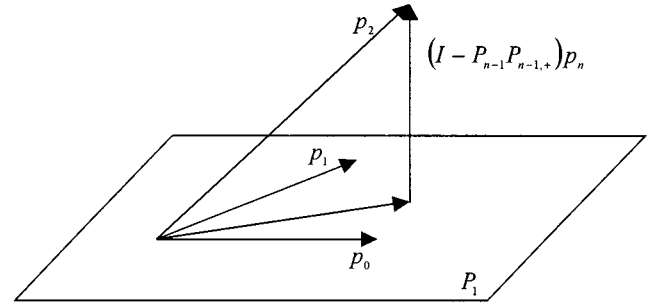
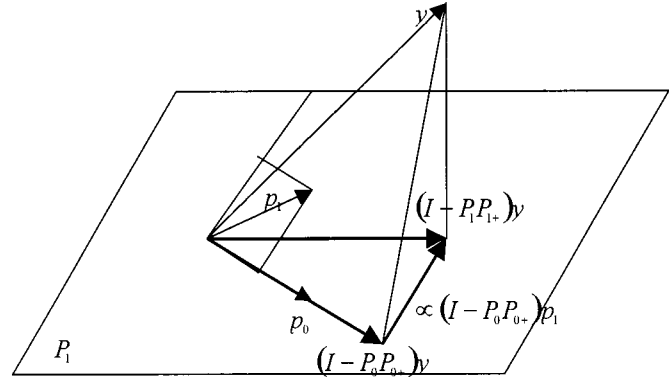

 Fig. 7. Geometric illustration of the vector $(I - P_{n-1}P_{n-1,+})p_n$.


Fig. 8. Geometric illustration of formula (22).

the ensemble. The whole ensemble of K possible choices of the parameters c_{nj} and θ_{nj} is divided in M portions each having L members so that $K = ML$. In the first portion, the parameters c_{nj} and θ_{nj} are generated from the intervals $[0, 1]$ and $[0.2\pi] - [\pi - \Delta, \pi + \Delta]$ respectively, using uniform distribution of the parameters on the respective intervals. After the first portion of the parameters c_{nj}, θ_{nj} has been chosen, the parameters a_0, a_1, \dots, a_n , the net output, and the training error have been calculated, the net with the minimal training error has been identified. The internal parameters c_{njopt} and θ_{njopt} of this optimal net are kept in memory and used to correct the distribution of the parameters in the next portion. For this and all subsequent portions, instead of the uniform, the triangle distribution is used. The graphs of the probability density functions $p(c_{nj}), p(\theta_{nj}), p(c_{nj}, \theta_{nj}) = p(c_{nj})p(\theta_{nj})$ of the parameters c_{nj}, θ_{nj} for a portion $m, m = 2, \dots, M$ are shown in Fig. 8. Here $c_{nj, m-1, opt}$ and $\theta_{nj, m-1, opt}$ are optimal

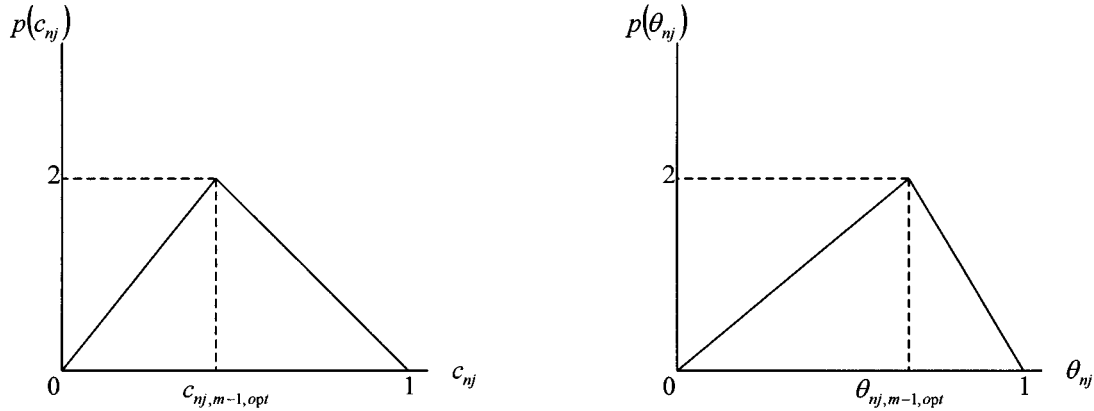


Fig. 9. Graphs of probability density function of the internal parameters in portions $m = 2, 3, \dots$.

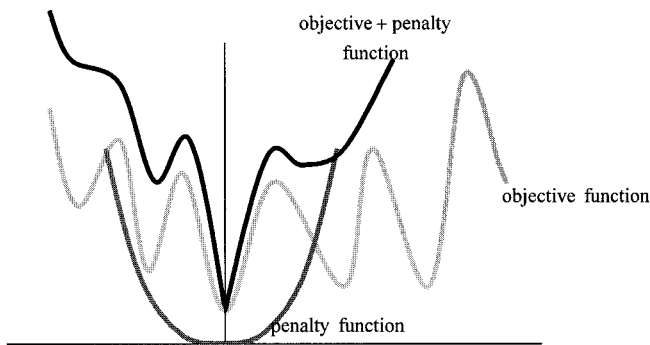


Fig. 10. Using penalty function for eliminating local minima.

values of the parameters c_{nj} and θ_{nj} found after completing group $(m - 1)$. The parameters θ_{nj} are actually sampled from the interval $[0, 1]$, and multiplied by 2π , and then all values of the parameters between π and $\pi - \Delta$ are replaced by $\pi - \Delta$, while all values between π and $\pi + \Delta$ are replaced by $\pi + \Delta$.

The justification for this procedure is given in [13]. Suppose, instead of a triangle, a Gaussian distribution, centered at the estimate of the point of global minimum of training error prior to m th portion, is used in m th portion. Suppose additionally that the width of the Gaussian distribution is decreasing to zero with m approaching infinity. Then ASO is equivalent to the following procedure of eliminating local minima. Add to the objective function (training error) $F(c, \theta)$ a quadratic penalty function $\zeta(c, \theta) = \mu[(c - c_*)^T(c - c_*) + (\theta - \theta_*)^T(\theta - \theta_*)]$, where (c_*, θ_*) is a point of global minimum of $F(c, \theta)$, μ , reciprocal of the width of Gaussian, is the parameter, tending to infinity with $m \rightarrow \infty$. Then, the quadratic penalty function will eventually dominate in the sum, and the sum will behave as a function with the same global minimum as the objective function, but without local minima. The univariate case is shown in Fig. 10. However, the location of global minimum is unknown and we have to use its current estimate. That is why the current estimate is updated and this procedure goes further in iterative manner.

The triangle distribution serves as a rough approximation for the Gaussian with the practical advantage that it has no adjustable parameters.

1) *The Stopping Rule:* The process of growing a net node by node is stopped if the maximal number of nodes has been exceeded, or for a long period (measured in number of nodes)

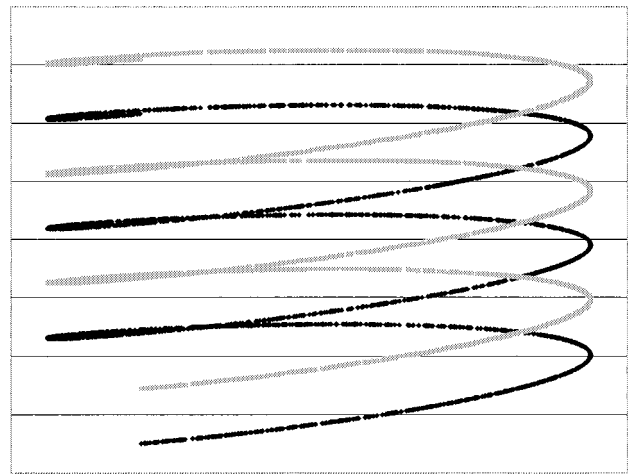


Fig. 11. Two helices.

where the generalization error does not change significantly. The EA cast away this period in the net used for prediction.

2) *Multioutput Case:* Consider a net with s outputs

$$\tilde{\mathbf{f}}(x) = \mathbf{a}_0 + \sum_{n=1}^N \mathbf{a}_n g[\varphi(x, c, w)] \quad (27)$$

where $\tilde{\mathbf{f}}$, \mathbf{a}_0 , \mathbf{a}_n are s -dimensional column-vectors. In this case only formula (26) should be changed to

$$\mathbf{A} = P_n + \mathbf{Y} \quad (28)$$

where \mathbf{A} is a $(N + 1) \times s$ matrix

$$\mathbf{A} = [\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_N] \quad (29)$$

\mathbf{Y} is a $P \times s$ matrix of target function values

$$\mathbf{Y} = \begin{bmatrix} y_{11}, \dots, y_{1s} \\ \dots \dots \dots \\ y_{P1}, \dots, y_{Ps} \end{bmatrix}. \quad (30)$$

IV. APPLICATION AND MATHEMATICAL EXAMPLES

Example 1—Two Helices: We consider it a difficult task to discern the data placed on two helices close to each other as shown in Fig. 11. $P = 10000$ points are randomly placed on

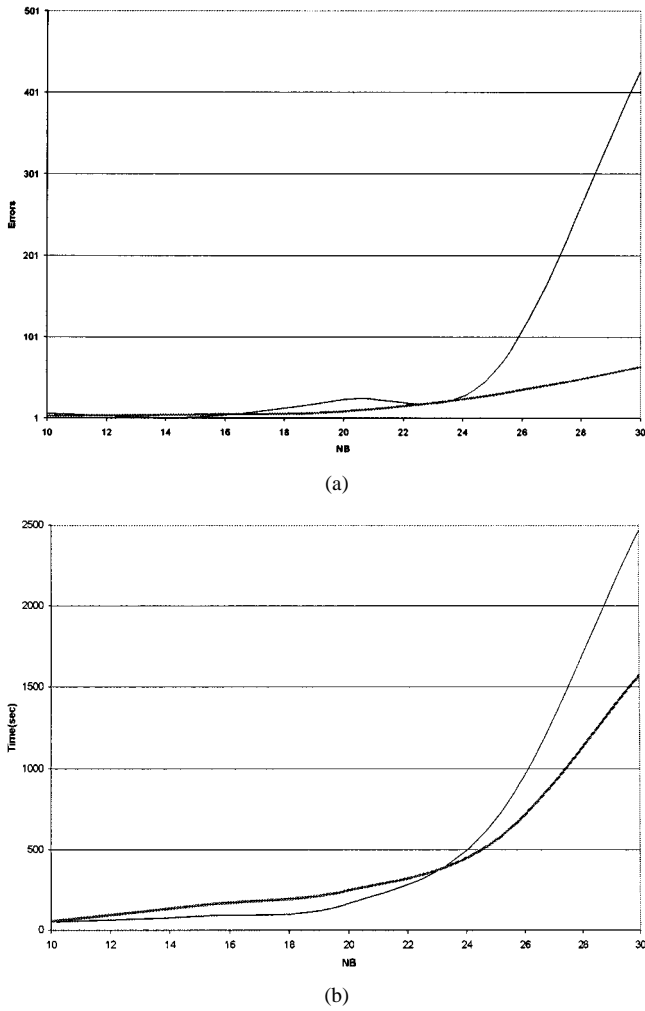


Fig. 12. Errors and time versus NB , case $R = 12, N = 15$.

two helices. The coordinates of these points are calculated by the equations

$$\begin{aligned} x &= R \cos t, & y &= R \sin t, & z &= Vt \\ x &= R \cos t, & y &= R \sin t, & z &= V(t + \pi) \end{aligned}$$

where the parameter $t, 0 \leq t \leq 2\pi N$ is chosen randomly and uniformly. These coordinates are the inputs of two neural nets, CWN and RBF. The outputs of the nets take values of zero or one depending on the helix where the point with coordinates equals to the input, is placed. The number of loops N equals to 12 or 15, $V = 1$, and $R = 8$ or 12, 2500 patterns of the data were used for testing. The algorithm for training is local sequential. The parameter NB is the maximum number of nearest neighbors. For each testing pattern, a net with recursively increasing number of nodes is trained on the set of NB nearest neighbors belonging to the training set (7500 patterns). The training stops if the training error becomes less than THRESHOLD (which was 0.05) or the number of nodes becomes larger than NB . The complexity of task, as a rule, will increase with NB increasing in the range of parameters we used. This can be explained as follows. The minimal distance from a tested pattern to a helix, not containing this pattern, equals π . The average number of patterns, lying on the same helix as tested pattern within distance

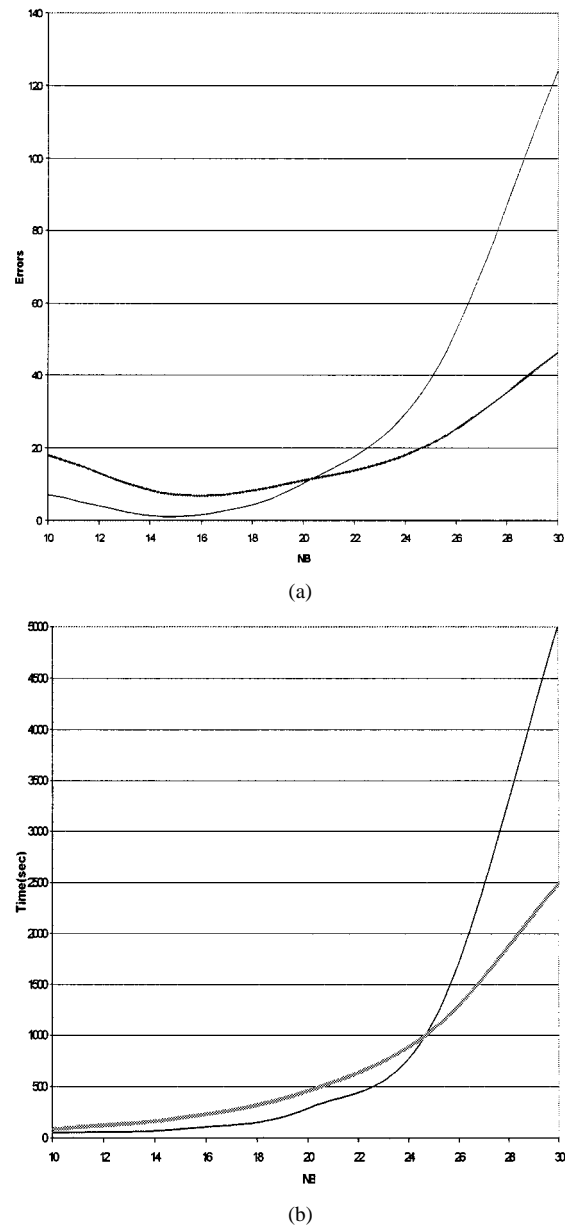
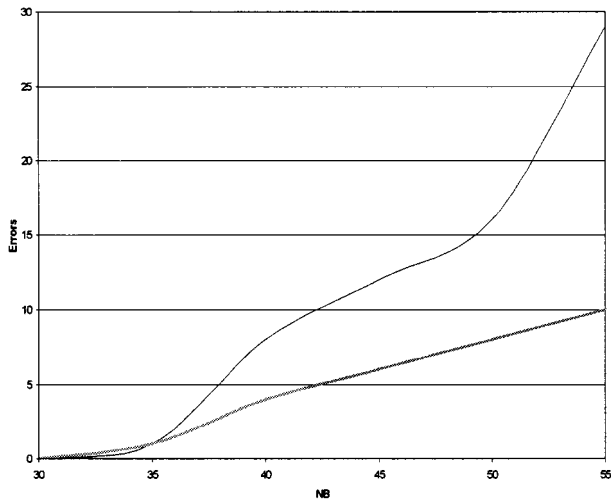


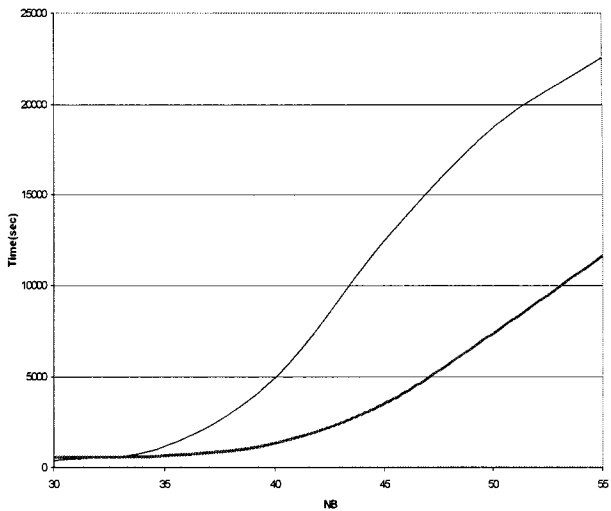
Fig. 13. Errors and time versus NB , case $R = 8, N = 15$.

π , equals to P/NR . If $NB \approx P/NR$ then almost each nearest neighbor will be from the same helix as tested pattern. The average value of nodes actually used for testing and the number of errors will be small. With an increase of NB more and more neighbors from another helix appear. The task of classification then becomes more difficult, and the number of errors increases. The time of training increases as well because of two reasons: a simple increase of the training set size and an increase of the average number of net nodes. Of course, statistical fluctuations from the averages in the distribution of the data on the helices play important role. The importance of them increases with decrease of NB .

The results of experiments are shown in Figs. 12–15. The comparison between CWN (thick curves) and RBF are made in the accuracy in testing, and the time for training and testing with varying NB . All graphs have demonstrated that CWN is superior than RBF, both in accuracy and time, when the number



(a)

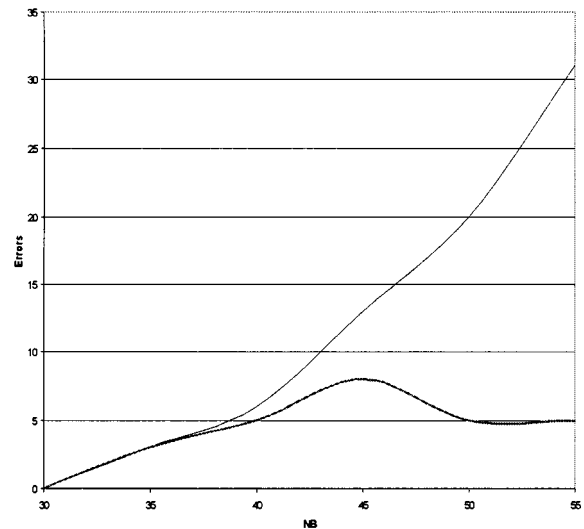


(b)

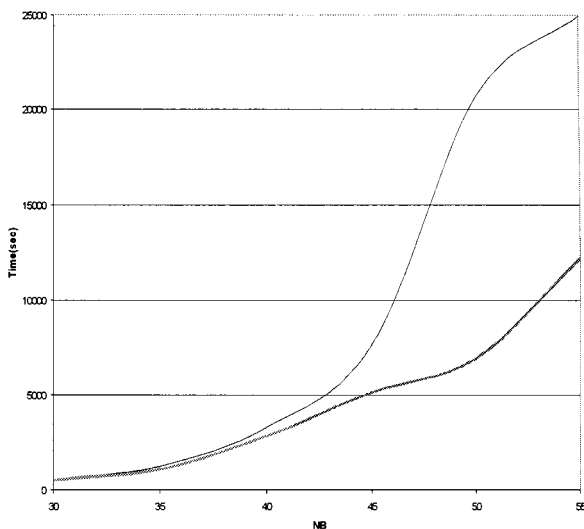
Fig. 14. Errors and time versus NB , case $R = 12$, $N = 12$.

of nearest neighbors becomes larger than some value of NB . That means that CWN is superior for more difficult versions of this task. Then, the task is easier and can be solved with small and approximately the same average number of nodes. The advantage of RBF over CWN in computational time for one node plays the major role. Easier tasks can be solved sometimes more accurately and efficiently by RBF.

Example 2—The Formers–Nonformers Problem [26]: A body of data constitutes 6358 patterns of ternary systems (systems of three chemical elements) with 15 features of the elements in the system, five for each element. These are Zunger radius, valence, melting temperature, Mendeleev number, and electrical negativity. For each system, it is known whether it can or can not form a compound. This information is available through long and expensive experimentation and lengthy calculations. The task is to build a neural net that can accurately predict the possible formation of a compound for a new system, not available in the database. It was found empirically [26], [27] that the Mendeleev number is superior to other features in this task. The comparison between CWN net and the RBF net in accuracy was made using only Mendeleev numbers



(a)



(b)

Fig. 15. Errors and time versus NB , case $R = 8$, $N = 12$.

as inputs for two nets, both using the ensemble approach for learning and generalization. Different modes of training and testing were tried, including nonlocal sequential (NLS), nonlocal nonsequential (NLNS), and local sequential (LS) with 40 nearest neighbors. The results of testing on the subset of data consisting of 1589 patterns (4769 patterns were used as a training set) are shown in Table I. CWN demonstrates an obvious superiority over RBF net.

Our experiments have indicated that the activation function for the CWN can be chosen from the same set of functions as for RBF net. In particular, the minimum of generalization error is achieved with the “thin plate” activation function $f(t) = t^2 \log(t)$. The dependencies of testing error on the number of nodes are shown in Fig. 16. Two flat regions on both curves indicate that the nets converged to a local minima but were able to escape.

Finding out that CWN + LS is the best choice we continued experimentation adding other features to Mendeleev number. Better results were obtained only using all five properties (15

TABLE I
RESULTS OF TESTING

Node type	Mode type	# of misclassifications	Accuracy in %
CWN	NLS	48	97.0
RBF	NLS	60	96.3
CWN	NLNS	60	96.3
RBF	NLNS	72	95.5
CWN	LS	15	99.1
RBF	LS	40	97.5

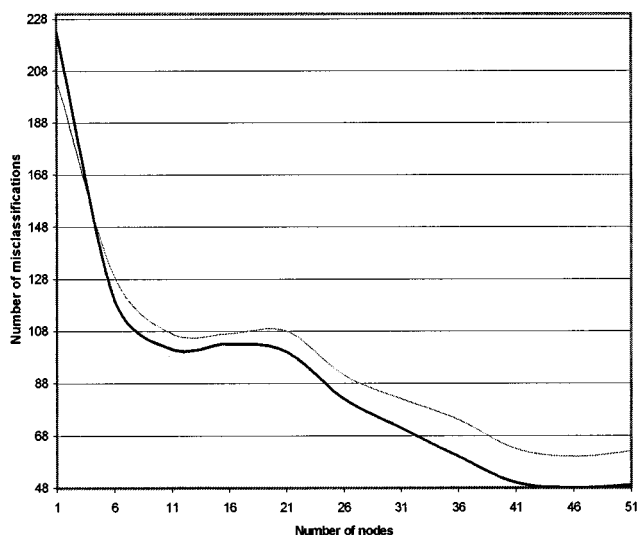


Fig. 16. Testing error for CWN (thick curve) and RBF net versus number of nodes.

inputs). 99.5% accuracy in testing was achieved for the same training and testing sets.

Example 3—Cellular Automata Model of Thin Film Growth: CWN and RBF net were compared in the task of building cellular automata (CA) based model of thin film growth [28]. Schematic representation of this process is shown in Fig. 17. The atoms of types A and B are sent to the substrate by two heated sources. Those atoms which make bonds between each other and/or with the substrate form a film surface. The geometrical features of the surface, such as average roughness for example, are of great importance for the quality of the film. Depending on the current state of the surface and substrate an incoming atom can form different types of bonding with the surface or remain in the vapor. For the current state of the model, six possible states of the atom are assumed. These are AA bonded, AB bonded, absorbed, wall-absorbed, cliff-absorbed, and vapor. In the CA model, it is supposed that the actual state of an atom depends not on the entire substrate and surface, but only on the states of the atoms in the neighborhood of the incoming atom. The neighborhood constitutes 26 cells that together with an incoming atom form a cube in three dimensions (3-D) with the incoming atom in the center of the cube. This is shown in Fig. 18, where a cubical neighborhood and its three layers are presented. The incoming atom is in the center of the middle layer. Surrounding cells filled by atoms of type A or B, or empty. The state of the incoming atom can be determined given the state of the

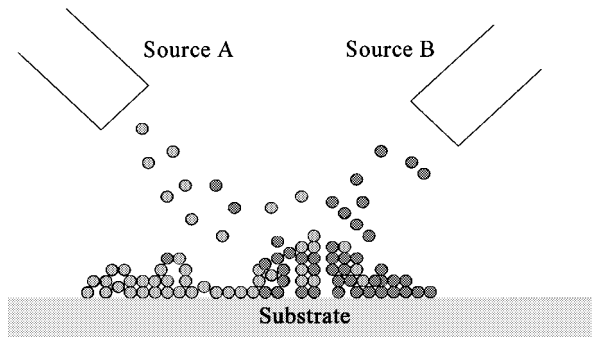


Fig. 17. Schematic representation of thin film growth.

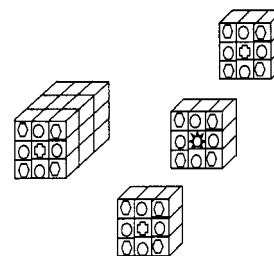


Fig. 18. 3-D structure of incoming atom neighborhood.

neighborhood, temperature and some probabilities calculated by using laws of statistical physics. It is impossible for that model to operate in a reasonable time, given that calculations should be made for millions of atoms. That is why the neural net is used. After training on a number of known examples it can predict the current state of the incoming atom.

In the current state of the model, we use two discrete variables characterizing the neighborhood, temperature, and three probabilities (altogether six variables), as inputs to a neural net, and one discrete output taking six possible values. The number of patterns used for training is 3208, and the number of patterns used for testing is 1069. The comparison between RBF and CWN is made in terms of the number of misclassifications of the output state and the time required for prediction of the state of incoming atom. The results are shown in Table II.

Example 4—Learning Dependency of the Optical Thickness of Thin Film on Its Spectral Pattern: The data set consists of 676 points describing dependency of the optical thickness of a thin film (output) on its spectral pattern (input) [29]. The input constitutes a 33-dimensional vector. The output values were uniformly distributed in the range [0.5, 5.5] with the average value equal to three. Thus, 1% of error corresponds to 0.03 or 0.0009 MSE. Three quarters of the data (507 patterns) were used for training and one quarter (169 patterns) for testing. This is an

TABLE II
COMPARISON OF RBF NET AND CWN IN ACCURACY AND TIME

Type of net	# of training patterns	# of testing patterns	# of misclassified testing patterns	% of correct testing patterns	Time per one testing pattern
RBF	3208	1069	70	93.4	14 μ sec
CWN	3208	1069	60	94.4	12 μ sec

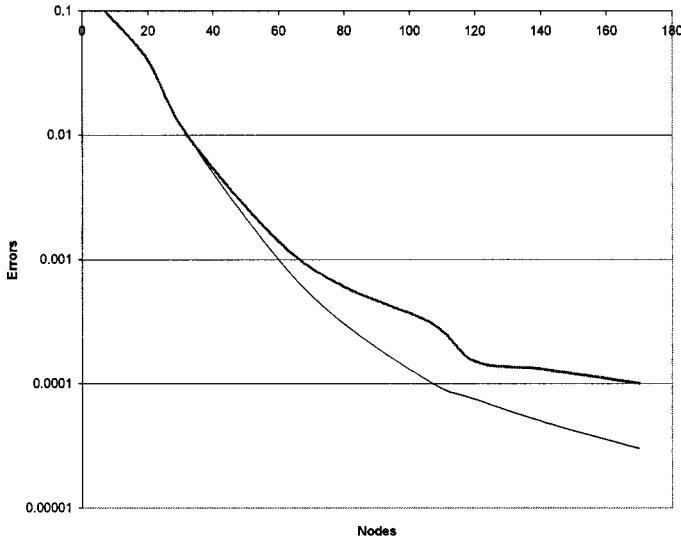


Fig. 19. Training and testing (thick curve) errors versus number of nodes for CWN.

example of learning a continuous function with a large number of variables. The results of training and testing for a CWN are shown in the graphs in Fig. 19. A level of 1% of testing error was achieved with a net of 68 nodes (0.000 895 MSE), while the training error was 0.8% (0.000 597 MSE). The best results were obtained with a net of 170 nodes: testing error 0.37% (0.000 121 MSE), training error 0.16% (0.000 031 MSE). The corresponding results for RBF net of the same size were: testing error 0.5% (0.000 225 MSE), training error 0.27% (0.000 063 MSE). The level of 1% of testing error (0.000 911 MSE) was achieved with the net of 75 nodes, with testing error 0.83% (0.000 624 MSE).

These examples confirm that CWN has a visible advantage in accuracy and efficiency of learning and generalization compared with the RBF net. These advantages will become even greater when the quantum computers will be able making calculations with complex numbers.

V. CONCLUSION AND FUTURE WORK

The new architecture of neural network, suggested in this paper, has a solid motivation and has proved a visible advantage on the RBF net in performance and efficiency in a number

of applications. Our future work will concentrate both on applications of this architecture, particularly in the area of smart sensors, and on theoretical development of a new, completely adaptive architecture.

APPENDIX

UNIVERSAL APPROXIMATION CAPABILITY OF THE CWN

Defining appropriate metrics in the space of continuous functions on the standard unit hypercube $I^d = \{(x_1, \dots, x_d) | 0 \leq x_1 \leq 1, \dots, 0 \leq x_d \leq 1\}$, we prove that CWN has the universal approximation capability. That is, for any function f from that space and any $\varepsilon > 0$ there exists a CWN such that the distance between CWN and f is less than ε .

Suppose the external function g satisfies the conditions

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left| g \left(\sum_{j=1}^d t_j^2 \right) \right| dt_1 \cdots dt_d < \infty, \quad (A1)$$

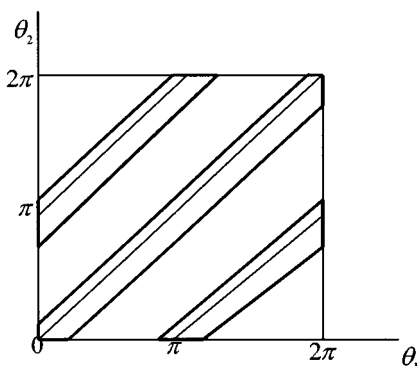
$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g \left(\sum_{j=1}^d t_j^2 \right) dt_1 \cdots dt_d \neq 0. \quad (A2)$$

First, we prove the following lemma.

Lemma 1: If a univariate function g satisfies the conditions (A1) and (A2), and the area of integration over $\theta_1, \dots, \theta_d$ in (A3) is such, that $\theta_1 \neq \theta_2, \theta_2 + \pi, \theta_2 - \pi$ (A), then for any $\alpha > 0$ there exists a constant C_α , such that

$$C_\alpha \int_0^{2\pi} \cdots \int_0^{2\pi} d\theta_1 \cdots d\theta_d \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 dy_1 \cdots dy_d \right] = 1. \quad (A3)$$

Comment: In practice we use a random sample of parameters $\theta_1, \dots, \theta_d$. The probability that the condition (A) is fulfilled equals to one. That is why we omit this condition in writing the limits of integration over $\theta_1, \dots, \theta_d$.


 Fig. 20. The stripes excluded from integration over θ_1 and θ_2 .

Proof: Assuming that the condition (A) is satisfied, we introduce new variables

$$t_1 = \sum_{j=1}^d y_j \cos \theta_j, \quad t_2 = \sum_{j=1}^d y_j \sin \theta_j, \quad t_j = \sqrt{\alpha} y_j$$

for $j = 3, \dots, d$

and calculate the Jacobian

$$\frac{d(t_1, \dots, t_d)}{d(y_1, \dots, y_d)} = \begin{vmatrix} \cos \theta_1 & \cos \theta_2 & \cos \theta_3 & \dots & \cos \theta_d \\ \sin \theta_1 & \sin \theta_2 & \sin \theta_3 & \dots & \sin \theta_d \\ 0 & 0 & \sqrt{\alpha} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \sqrt{\alpha} \end{vmatrix}$$

$$= (\sqrt{\alpha})^{d-2} \sin(\theta_2 - \theta_1).$$

Equation (A3) then can be written as

$$\frac{C_\alpha}{\sqrt{\alpha}^{d-2}} \int_0^{2\pi} \dots \int_0^{2\pi} d\theta_1 \dots d\theta_d \frac{1}{\sin(\theta_2 - \theta_1)} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}$$

$$\times g \left(\sum_{j=1}^d t_j^2 \right) dt_1 \dots dt_d$$

$$= \frac{C_\alpha}{\sqrt{\alpha}^{d-2}} (2\pi)^{d-2} \int_0^{2\pi} \int_0^{2\pi} d\theta_1 d\theta_2 \frac{1}{\sin(\theta_2 - \theta_1)}$$

$$\times \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} g \left(\sum_{j=1}^d t_j^2 \right) dt_1 \dots dt_d = 1.$$

The integral $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(\sum_{j=1}^d t_j^2) dt_1 dt_2$ exists and does not equal to zero by virtue of (A1) and (A2). The integral $\int_0^{2\pi} \int_0^{2\pi} (d\theta_1 d\theta_2 / \sin(\theta_2 - \theta_1))$ exists by virtue of (A). It can be made nonzero by excluding small asymmetric stripes around the lines $\theta_1 = \theta_2$, $\theta_1 = \theta_2 \pm \pi$ in the square $0 \leq \theta_1 \leq 2\pi$, $0 \leq \theta_2 \leq 2\pi$, as shown in Fig. 20. Therefore

$$C_\alpha = \sqrt{\alpha}^{d-2} \left/ \left[(2\pi)^{d-2} \int_0^{2\pi} \int_0^{2\pi} \frac{d\theta_1 d\theta_2}{\sin(\theta_2 - \theta_1)} \right. \right.$$

$$\left. \left. \times \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g \left(\sum_{j=1}^d t_j^2 \right) dt_1 dt_2 \right] \right.$$

We then derive a limit-integral representation of a continuous multivariate function f , defined on the standard unit hypercube I^d . This representation is contained in the following lemma.

Lemma 2: Let f be a continuous function, defined on I^d , and g be a univariate function, satisfying the conditions (A1), (A2) and (A) of lemma 1. Denote D the hypercube $\{(\theta_1, \dots, \theta_d) | 0 \leq \theta_1 \leq 2\pi, \dots, 0 \leq \theta_d \leq 2\pi\}$ with excluded parallelepipeds with bases in the plane $\theta_1 O \theta_2$, as shown in Fig. 4, and edges parallel to axes $O \theta_3, \dots, O \theta_d$. Then for any x from the interior of I^d the following limit-integral representation is true:

$$f(x_1, \dots, x_d)$$

$$= \lim_{w \rightarrow \infty} \int_{I^d} \int_D d\theta_1 \dots d\theta_d w^d$$

$$\times g \left[w^2 \left(\sum_{j=1}^d (x_j - c_j) e^{i\theta_j} \right) \left(\sum_{j=1}^d (x_j - c_j) e^{-i\theta_j} \right) \right]$$

$$\times f(c_1, \dots, c_d) dc_1 \dots dc_d. \quad (\text{A4})$$

Proof: Without loss of generality we can assume, that the function g is divided by the constant C_α in (A3), so that g satisfies the following equation for any constant $\alpha > 0$

$$\int_D d\theta_1 \dots d\theta_d \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty}$$

$$\times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right]$$

$$\times dy_1 \dots dy_d = 1. \quad (\text{A5})$$

As it will be seen from the proof of the Theorem, this assumption can be made without calculating the constant C_α . Applying Euler's formula one obtains

$$\sum_{j=1}^d e^{i\theta_{nj}} (x_j - c_{nj}) \sum_{i=1}^d e^{-i\theta_{nj}} (x_j - c_{nj})$$

$$= \left(\sum_{j=1}^d (x_j - c_{nj}) \cos \theta_{nj} \right)^2 + \left(\sum_{j=1}^d (x_j - c_{nj}) \sin \theta_{nj} \right)^2.$$

Replacement of the variables c_1, \dots, c_d by the variables $y_1 = w(x_1 - c_1), \dots, y_d = w(x_d - c_d)$ in (20) yields

$$f(x_1, \dots, x_d)$$

$$= \lim_{w \rightarrow \infty} \int_D d\theta_1 \dots d\theta_d \int_{w(x_1-1)}^{wx_1} \dots \int_{w(x_d-1)}^{wx_d}$$

$$\times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right]$$

$$\times f(x_1 - y_1/w, \dots, x_d - y_d/w) dy_1 \dots dy_d. \quad (\text{A6})$$

Further proof is based on the following observations. First, since $0 < x_1 < 1, \dots, 0 < x_d < 1$, the limits of integration in the integral over y_1, \dots, y_d in (A6) are approaching $\pm\infty$ when

$w \rightarrow \infty$. Second, if $y_j = O(\sqrt{w})$, then $x_j - y_j/w \sim x_j$ for $j = 1, \dots, d$. Third,

$$\begin{aligned} & \int_D d\theta_1 \cdots d\theta_d \int_{-\sqrt{w}}^{\sqrt{w}} \cdots \int_{-\sqrt{w}}^{\sqrt{w}} \\ & \times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right] \\ & \times dy_1 \cdots dy_d \underset{w \rightarrow \infty}{\sim} \int_D d\theta_1 \cdots d\theta_d \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \\ & \times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right] \\ & \times dy_1 \cdots dy_d = 1. \end{aligned} \quad (A7)$$

Fourth, let F be a set

$$\begin{aligned} F = & [(x_1 - 1)w, x_1 w] \times \cdots \times [(x_d - 1)w, x_d w] \\ & - [(x_1 - 1)\sqrt{w}, x_1 \sqrt{w}] \\ & \times \cdots \times [(x_d - 1)\sqrt{w}, x_d \sqrt{w}]. \end{aligned}$$

Then

$$\begin{aligned} & \int_D d\theta_1 \cdots d\theta_d \int_F \\ & \times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right] \\ & \times dy_1 \cdots dy_d \xrightarrow{w \rightarrow \infty} 0 \end{aligned}$$

and, since the continuous function f on a closed bounded set is bounded both from above and below

$$\begin{aligned} & \int_D d\theta_1 \cdots d\theta_d \int_F \\ & \times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right] \\ & \times f(x_1 - y_1/w, \dots, x_d - y_d/w) dy_1 \cdots dy_d \xrightarrow{w \rightarrow \infty} 0. \end{aligned} \quad (A8)$$

Combination of these four observations yields

$$\begin{aligned} & \lim_{w \rightarrow \infty} \int_D d\theta_1 \cdots d\theta_d \int_{w(x_1-1)}^{wx_1} \cdots \int_{w(x_d-1)}^{wx_d} \\ & \times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right] \\ & \times f(x_1 - y_1/w, \dots, x_d - y_d/w) dy_1 \cdots dy_d \\ & = \lim_{w \rightarrow \infty} \int_D d\theta_1 \cdots d\theta_d \int_{\sqrt{w}(x_1-1)}^{\sqrt{wx_1}} \cdots \int_{\sqrt{w}(x_d-1)}^{\sqrt{wx_d}} \\ & \times g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right] \end{aligned}$$

$$\begin{aligned} & \times f(x_1 - y_1/w, \dots, x_d - y_d/w) dy_1 \cdots dy_d \\ & = f(x_1, \dots, x_d) \times \lim_{w \rightarrow \infty} \int_D d\theta_1 \cdots d\theta_d \int_{\sqrt{w}(x_1-1)}^{\sqrt{wx_1}} \cdots \\ & \int_{\sqrt{w}(x_d-1)}^{\sqrt{wx_d}} g \left[\left(\sum_{j=1}^d y_j \cos \theta_j \right)^2 \right. \\ & \left. + \left(\sum_{j=1}^d y_j \sin \theta_j \right)^2 - \alpha \sum_{j=3}^d y_j^2 \right] \\ & = f(x_1, \dots, x_d). \end{aligned}$$

Making $\alpha \rightarrow 0$ completes the proof.

The main result is contained in the following theorem.

Theorem (The Universal Approximation Capability of the CWN): Define a distance between a function f , defined and continuous on I^d , and a CWN f_N

$$\begin{aligned} f_N(x) = & a_0 + \sum_{n=1}^N a_n \\ & \times g \left[w^2 \sum_{j=1}^d e^{i\theta_{nj}} (x_j - c_{nj}) \sum_{i=1}^d e^{-i\theta_{ni}} (x_j - c_{ni}) \right] \end{aligned} \quad (A9)$$

as

$$\rho(f, f_N) = \sqrt{E \int [f(x) - f_N(x)]^2 dx}. \quad (A10)$$

Then for any $\varepsilon > 0$ and any function f , defined and continuous on I^d , there exists a CWN f_N , such that

$$\rho(f, f_N) < \varepsilon. \quad (A11)$$

Proof: For any $\varepsilon > 0$ and any $\delta > 0$ there exists w such that uniformly for

$$x \in I_\delta^d = \{(x_1, \dots, x_d) | \delta \leq x_1 \leq 1-\delta, \dots, \delta \leq x_d \leq 1-\delta\}$$

the following inequality is true:

$$|f(x) - \varphi(x)| < \varepsilon/2 \quad (A12)$$

where

$$\begin{aligned} \varphi(x) = & \int_{I^d} \int_D d\theta_1 \cdots d\theta_d w^d \\ & \times g \left[w^2 \left(\sum_{j=1}^d (x_j - c_j) e^{i\theta_j} \right) \left(\sum_{j=1}^d (x_j - c_j) e^{-i\theta_j} \right) \right] \\ & \times f(c_1, \dots, c_d) dc_1 \cdots dc_d. \end{aligned}$$

Replacing the integral in the right-hand side of the last equation by the Monte-Carlo method [21], one obtains for some integer positive N and $x \in I_\delta^d$

$$|\varphi(x) - f_N(x)| < \varepsilon/2. \quad (A13)$$

The coefficients a_0, a_1, \dots, a_N in $f_N(x)$ are independent of x and subject to minimization of the training error. That is why

there is no need in calculation of the constant C_α ! Making use of the triangle inequality in (A11), (A12) yields

$$|f(x) - f_N(x)| < \varepsilon \quad \text{for all } x \in I_\delta^d. \quad (\text{A14})$$

Integrating the square of (30) over $x \in I_\delta^d$, taking the mathematical expectation E over random parameters and making $\delta \rightarrow 0$ completes the proof.

REFERENCES

- [1] S. Haykin, *Neural Networks. A Comprehensive Foundation*. New York: Macmillan, 1994.
- [2] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995.
- [3] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–293, 1989.
- [4] J. H. Friedman and W. Stuetzle, "Projection pursuit regression," *J. Amer. Statist. Assoc. (JASA)*, vol. 76, pp. 817–823, 1981.
- [5] J.-N. Hwang, S.-S. You, S.-R. Lay, and I.-C. Jou, "The cascade-correlation learning: A projection pursuit learning perspective," *IEEE Trans. Neural Networks*, vol. 7, pp. 278–288, 1996.
- [6] L. Breiman, "Hinging hyperplanes for regression, classification, and function approximation," *IEEE Trans. Inform. Theory*, vol. 39, pp. 999–1013, 1993.
- [7] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, pp. 109–118, 1990.
- [8] E. Gelenbe, "Theory of random neural network model," in *Neural Networks: Advances and Applications*, E. Gelenbe, Ed. New York: Elsevier, 1991, pp. 1–20.
- [9] C. L. Giles and T. Maxwell, "Invariance and generalization in high-order neural networks," *Appl. Opt.*, vol. 26, pp. 4972–4978, 1987.
- [10] I. Daubechies, "The wavelet transform, Time-frequency localization and signal analysis," *IEEE Trans. Inform. Theory*, vol. 36, pp. 961–1005, 1990.
- [11] A. N. Kolmogorov, "On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition," *Trans. Amer. Math. Soc.*, vol. 2, no. 28, pp. 55–59, 1963.
- [12] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions and adaptive function approximation," *IEEE Trans. Neural Networks*, vol. 6, pp. 1320–1329, 1995.
- [13] B. Igelnik, Y.-H. Pao, S. R. LeClair, and C. Y. Shen, "The ensemble approach to neural-network learning and generalization," *IEEE Trans. Neural Networks*, vol. 10, no. 1, pp. 19–30, 1999.
- [14] H. Katsura and D. A. Sprecher, "Computational aspects of Kolmogorov's superposition theorem," *Neural Networks*, vol. 7, pp. 455–461, 1994.
- [15] D. A. Sprecher, "A numerical implementation of Kolmogorov's superpositions II," *Neural Networks*, vol. 10, pp. 447–457, 1997.
- [16] M. L. Minsky and S. A. Papert, *Perceptrons*, Expanded ed. Cambridge, MA: MIT Press, 1988.
- [17] P. Arena, L. Fortuna, G. Muscato, and M. G. Xibilia, *Neural Networks in Multidimensional Domains. Fundamentals and New Trends in Modeling and Control*, ser. Lecture Notes in Control and Information Sciences, 234. New York: Springer-Verlag, 1998.
- [18] G. Georgiou and C. Koutsougeras, "Complex domain backpropagation," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 330–334, 1992.
- [19] M. S. Kim and C. C. Guest, "Modification of back-propagation for complex-valued signal processing in frequency domain," in *Proc. Int. Joint Conf. Neural Networks*, San Diego, 1990, pp. 27–31.
- [20] H. Leung and S. Haykin, "The complex backpropagation algorithm," *IEEE Trans. Signal Processing*, vol. 39, pp. 2101–2104, 1991.
- [21] A. Albert, *Regression and the Moore–Penrose Pseudoinverse*. New York: Academic, 1972.
- [22] M. Pincus, "A Monte Carlo method for the approximate solution of certain types of constrained optimization problems," *Op. Res.*, vol. 18, pp. 1225–1228, 1970.
- [23] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*. Philadelphia, PA: SIAM.
- [24] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Trans. Inform. Theory*, vol. 39, pp. 930–945, 1993.
- [25] B. Igelnik, "Learning Lennard–Jones potentials," Rep. GRCI Inc., Mar. 1999.

- [26] P. Villars, S. R. LeClair, and S. Iwata, "Interplay between large materials databases, semiempirical approaches, neuro-computing and first principles calculations," in *Proc. 2nd Int. Conf. Intell. Processing Manufacturing of Materials*, vol. 2, Honolulu, 1999, pp. 1399–1416.
- [27] Y.-H. Pao, B. F. Duan, Y. L. Zhao, and S. R. LeClair, "Analysis and visualization of category membership distribution in multivariate data," in *Proc. 2nd Int. Conf. Intell. Processing Manufact. Materials*, vol. 2, Honolulu, HI, 1999, pp. 1361–1369.
- [28] A. Jackson and M. Benedict, private communication, 1997.
- [29] S. Fairchild, private communication, 1998.
- [30] M. Kalos and P. A. Witlock, *Monte Carlo Methods*. New York: Wiley, 1986, vol. 1, Basics.



Boris Igelnik (M'97–SM'99) received the M.S. and Ph.D. degrees in electrical engineering from Moscow Institute of Telecommunications and the M.S. degree in mathematics from Moscow State University, Russia.

He is a Senior Scientist with Pegasus Technologies Inc., Mentor, OH, and an Adjunct Associate Professor in Electrical Engineering and Computer Science Department at Case Western Reserve University, Cleveland, OH. His current research interests are in the area of computational intelligence, multivariate

data visualization, optimization, and control.



Massood Tabib-Azar (S'83–M'86–SM'93) received the M.S. and Ph.D. degrees in electrical engineering from the Rensselaer Polytechnic Institute, Troy, NY.

He is an Adjunct Associate Professor in Electrical Engineering and Computer Science Department, Case Western Reserve University, Cleveland, OH. His current research interests include high-resolution evanescent microwave characterization of materials, SiC and GaN devices, optical sensors and actuators, and quantum devices and computers. He is author of three books, two book chapters, more than 110 journal publications, and numerous conference proceeding articles. He has introduced and chaired many international symposia in his fields of interest.

Dr. Tabib-Azar is a recipient of the 1991 Lilly Foundation Fellowship and he is a member of the New York Academy of Sciences, IEEE Electron Devices Society, APS, AAPT, and Sigma Xi research societies.



Steven R. LeClair received the M.S. and Ph.D. degrees in industrial engineering from Arizona State University, Tempe.

He is Chief of the Materials Process Design Branch, Materials and Manufacturing Directorate, Air Force Research Laboratory, Wright-Patterson Air Force Base, OH. In this capacity, he is responsible for developing and transitioning self-directed and self-improving process design and control systems in support of Air Force materials research. His experiences include over 20 years of research

and development of materials processing systems involving metal, ceramic, polymer and electro-optical materials and associated processes.

Dr. LeClair has been a member of the National Materials Advisory Board Committee on Materials and Process Information Highway, and an advisor to the Committee on New Materials for Sensor Technologies. He has also been a National Research Council, Postdoctoral Advisor, from 1987 to present. His research and international collaborations include serving as a member of International Federation for Information Processing (IFIP), Computer Assisted Manufacturing Working Group 5.3. He is also Regional Editor (USA) of the Editorial Board, Engineering Applications of Artificial Intelligence, Elsevier Sciences Ltd., London, England, from 1998 to present. He is a Fellow of the Society of Manufacturing Engineers and has been a licensed Professional Industrial Engineer since 1985. He was elected a Fellow of the Dayton Affiliate Societies Council in 1999.