Automation and Remote Control, Vol. 61, No. 7, Part 1, 2000, pp. 1113–1124. Translated from Avtomatika i Telemekhanika, No. 7, 2000, pp. 55–67.

STOCHASTIC SYSTEMS

Recurrent Neural Network Detecting Changes in the Properties of Nonlinear Stochastic Sequences

E. V. Bodyanskii and S. A. Vorob'ev

State Technical University of Radioelectronics, Kharkov, Ukraine

Received February 16, 1998

Abstract—An approach is proposed to detecting changes in the properties of stochastic sequences obeying nonlinear equations of autoregression moving average. It is assumed that the changes in the sequence properties can have both the parametric and structural nonstationarity (change of order) forms. An architecture of multilayer recurrent neural network and neuron adjustment algorithms providing the maximal rate of learning are proposed. The possibility of diagnosing stochastic sequences of arbitrary structure, high speed, and simplicity of computations are the advantages of the approach proposed.

1. INTRODUCTION

The problem of detecting changes in the properties of stochastic sequences which is closely related with technical and medical diagnosis was widely covered in the literature [1–5]. Numerous approaches related mostly with the concepts of mathematical statistics, random-process theory, pattern recognition, cluster analysis, and so on were proposed. Without going into criticism of the existing results, we just note that the rigid assumptions about the statistical properties of series limit the possibilities of these methods.

The multimodel approach [1, 5–8] where the diagnosed signal is passed through a set of models each of which relies on an individual hypothesis about the nature of possible changes seems to be more universal. If certain hypotheses are satisfied indeed, the updating signals at the outputs of the corresponding models must be small. Therefore, the decision mechanism is based in essence on finding the model with the minimal updating at the output, the probability of the corresponding hypothesis being maximal. The advantages of this approach are certain, but in real life the sequence is usually so complicated and diversified that none of the (usually, linear) models reflects fully its changing properties.

The recent years witnessed a burst of studies in the area of theory and application of the artificial neural networks, including the problems of diagnosis [9–15]. The proposed diagnostic neural networks mostly realize the concepts of the theory of classification in the presence of learning sample; here, the network cannot detect process states that were not envisioned in advance.

The present paper proposes an architecture of the multilayer recurrent artificial neural network and algorithms to adjust its parameters which combine the advantages of the multimodel approach and approximating properties of the predicting neural networks [16–19] with nonlinear activation functions. Changes in the properties of a stochastic sequence are reflected by the diagnostic vector whose elements are the synaptic weights of the output neuron.

2. ARCHITECTURE OF THE DIAGNOSTIC NEURAL NETWORK

The proposed architecture of the diagnostic recurrent neural network (Fig. 1a) consists of elementary neurons differing in the form of the activation functions and learning algorithms, which generally are the gradient procedures of unconditional or conditional optimization.

0005-1179/00/6107-1113$25.00

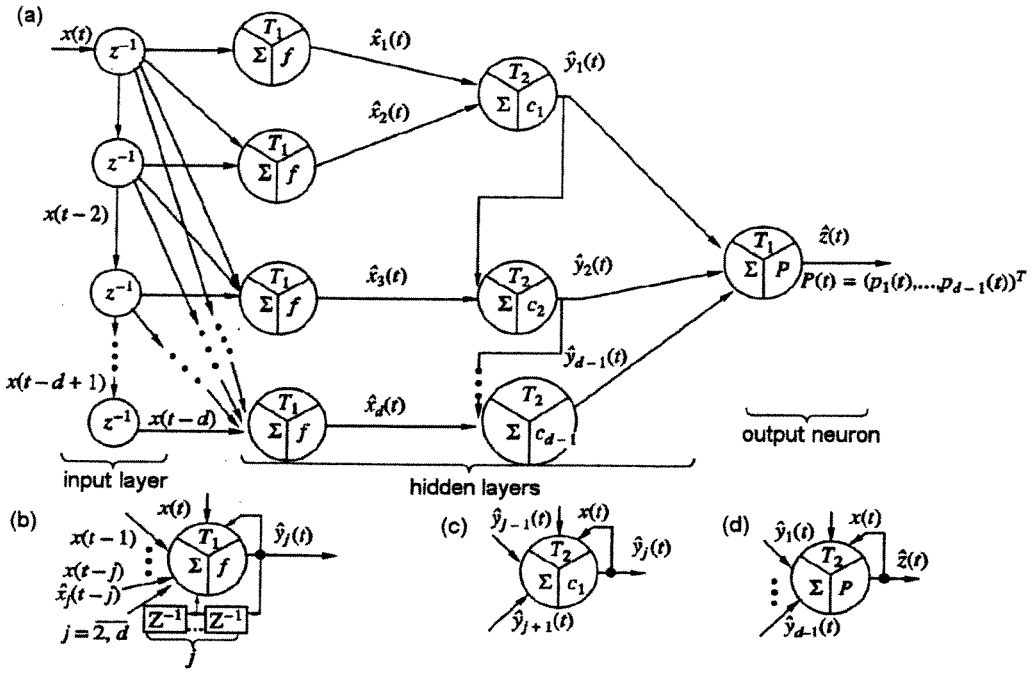

Fig. 1. (a) Architecture of the diagnostic network. (b), (c) Neurons of the hidden layers. (d) Output neuron.

The multilayer structure featuring optimal approximation and prediction [20, 21] is the prototype of the proposed artificial neural network. Structures such as the multilayer perceptron or recurrent Hopfield network respond to variations in the properties of the analyzed sequence by temporal deterioration of their predicting properties which are later restored as the neurons in layers learn. At the same time, these structures cannot establish the state of the monitored sequence after imbalance. This problem can be solved using the concept of multimodel approach which currently is rather well substantiated and developed [1, 5]—true, only for the linear systems. Realization of the multimodel approach by the technologies of neural networks would extend the class of solved diagnostic problems to the nonlinear objects.

The monitored stochastic sequence $\{x(t)\}$, $t = 1, 2, \ldots$ is fed into the input layer of the network which is a sequence of pure-delay elements z^{-1} : $(z^{-1}x(t) = x(t-1))$. As the result, the layer outputs a set of delayed values of the time series $x(t-1)$, $x(t-2), \ldots, x(t-d)$; here, the greater d, the wider the diagnostic possibilities of the neural network.

The first hidden layer consists of neurons of the McCulloch–Pitts type (see Fig. 1b) to whose summing inputs the delayed values of the summed sequence $x(t)$ and—through the feedback—the delayed values of the prediction $\widehat{x}_j(t)$, $j = 1, 2, \ldots, d$, are fed. The inputs of neurons T_1 correspond to the inputs of the algorithm for adjustment of the synaptic weights; and f describes the nonlinear function of neuron activation. As the result of processing the signal $x(t)$ by the first-layer neurons, they output the prediction estimates

$$\begin{cases} \widehat{x}_1(t) = f\left(x(t-1), \widehat{x}_1(t-1)\right), \\ \widehat{x}_2(t) = f\left(x(t-1), x(t-2), \widehat{x}_2(t-1), \widehat{x}_2(t-2)\right), \\ \vdots \\ \widehat{x}_d(t) = f\left(x(t-1), \ldots, x(t-d), \widehat{x}_d(t-1), \ldots, \widehat{x}_d(t-d)\right), \end{cases} \tag{2.1}$$

corresponding to the process of nonlinear autoregression moving average (NARMA-process) [17, 18] of the order ranging from 1 to d. Therefore, the neurons of the first hidden layer make up the

elementary "bricks" from which the second hidden layer "assembles" the optimal predictions of the sequence $x(t)$. The task of the network is to determine in real time the current value of the order of the NARMA-process and time of its possible changes.

The neurons of the second hidden layer T_2 (see Fig. 1c) join the output pairs of the neurons T_1 with the aim of getting the estimates $\widehat{y}_j(t)$, $j = 1, 2, \ldots, d-1$:

$$\begin{cases} \widehat{y}_1(t) = \varphi\left(\widehat{x}_1(t), \widehat{x}_2(t), c_1\right), \quad \widehat{x}_1(t) \equiv \widehat{y}_0(t), \\ \widehat{y}_2(t) = \varphi\left(\widehat{y}_1(t), \widehat{x}_3(t), c_2, c_1\right), \\ \vdots \\ \widehat{y}_{d-1}(t) = \varphi\left(\widehat{y}_{d-2}(t), \widehat{x}_d(t), c_{d-1}, c_{d-2}, \ldots, c_1\right) \end{cases} \tag{2.2}$$

and weight coefficients c_j characterizing precision of the predictions $\widehat{y}_{j-1}(t)$ and $\widehat{x}_{j+1}(t)$ that are joined and the joint prediction $\widehat{y}_j(t)$. It deserves noting that although the second layer formally performs pairwise joining, it is not precisely the case at the physical level. If the first neuron of the second hidden layer constructs the joint prediction on the basis of $\widehat{x}_1(t)$ and $\widehat{x}_2(t)$, then $\widehat{y}_2(t)$ already contains $\widehat{x}_1(t)$, $\widehat{x}_2(t)$, and $\widehat{x}_3(t)$; $\widehat{y}_3(t)-\widehat{x}_1(t)$, $\widehat{x}_2(t)$, $\widehat{x}_3(t)$, and $\widehat{x}_4(t)$, and so on. It is precisely in the second hidden layer that the optimal one-step predictions are generated which differ from each other in the value of the prehistory used. The vector of current weights $C(t) = (c_1(t), c_2(t), \ldots, c_{d-1}(t))^T$ describes the quality of prediction attained in the second hidden layer at each current time instant; and changes in the relations between its elements are indicative by themselves of the changes in the structure and parameters of the monitored sequence $x(t)$. We note also that already at the level of this layer one can establish from the number of the corresponding neuron the number of "physical" first-layer neurons required to approximate the signal $x(t)$.

As the result of learning, the neural network ensures optimal approximation of the monitored sequence. Here, the estimates of the "contributions" (diagnostic attributes $P(t) = (p_1(t), p_2(t), \ldots, p_{d-1}(t))^T$) of each prediction $\widehat{y}_j(t)$ to the general model of the monitored signal are generated in the single neuron of the output layer T_3 (see Fig. 1d). The maximal weight $p_j(t)$, which is the counterpart of the hypothesis that the "true" state of the sequence $x(t)$ is best described by the estimate $\widehat{y}_j(t)$, corresponds to the greatest "contribution." The maximal value of $p_j(t)$ defines the order of the diagnosed NARMA-sequence at the current time t; and the continuous updating of the vector $P(t)$ by the appropriate neuron adjustment algorithms enables one to determine the instant of imbalance in $x(t)$.

Upon the occurrence of imbalance accompanied by changes in the properties of the monitored signal, the "contributions" of individual $\widehat{y}_j(t)$ change correspondingly; and the optimal prediction is made by another combination of the second-layer neurons. It is only natural that in this case the corresponding $p_j(t)$ vary, which fact is reflected by the output neuron.

3. ADJUSTMENT ALGORITHM FOR THE NEURONS OF THE FIRST HIDDEN LAYER

The output of the jth neuron of the first hidden layer is representable as

$$\begin{aligned} \widehat{x}_j(t) &= f_j\left(\sum_{i=1}^{j} \omega_{ji}(t) x(t-i) + \sum_{i=1}^{j} \omega_{j,i+j}(t) \widehat{x}_j(t-i) + \omega_{j0}(t)\right) \\ &= f_j\left(\omega_j^T(t) X_j(t)\right) = f_j\left(\widetilde{X}_j(t)\right), \end{aligned} \tag{3.1}$$

where $f_j(\cdot)$ is the activation function of the jth neuron; $\omega_j(t) = (\omega_{j0}(t), \omega_{j1}(t), \ldots, \omega_{jj}(t), \omega_{j,1+j}(t), \ldots, \omega_{j,2j}(t))^T$ is the $((2j+1) \times 1)$-vector of the adjusted synaptic weights; $X_j(t) = (1, x(t-1),$

$\ldots, x(t-j), \hat{x}_j(t-1), \ldots, \hat{x}_j(t-j))^T$ is the vector of generalized inputs; $\tilde{X}_j(t) = \omega_j^T(t)X_j(t)$, $j = 1, 2, \ldots, d$; and $t = 1, 2, \ldots$ is the current discrete time.

Expression (3.1) describes the nonlinear stochastic sequence of autoregression moving average of the jth order. As was noted in [17], it is precisely the determination of the value of j that is the most difficult problem. In this connection, it is advisable to make use of sufficiently large values of d in particular problems.

By introducing the prediction error of the jth neuron

$$\varepsilon_j(t) = x(t) - \hat{x}_j(t) = x(t) - f_j\left(\tilde{X}_j(t)\right), \tag{3.2}$$

the gradient procedure of adjusting the synaptic weights is representable as [22, 23]

$$\omega_j(t+1) = \omega_j(t) + \eta_j(t)\varepsilon_j(t)\nabla_{\omega_j} f_j\left(\tilde{X}_j(t)\right) = \omega_j(t) + \eta_j(t)\varepsilon_j(t)G_j(t), \tag{3.3}$$

where $\eta_j(t)$ is the parameter of search step which is usually taken to be constant; and $\nabla_{\omega_j} f_j(\tilde{X}_j(t)) = G_j(t)$ is the gradient of the function of activation by the synaptic weights.

We note that for the most popular activation functions

$$\begin{cases} f_j'\left(\tilde{X}_j(t)\right) = \tanh\left(\gamma_j\tilde{X}_j(t)\right) = \dfrac{1 - \exp\left(-2\gamma_j\tilde{X}_j(t)\right)}{1 + \exp\left(-2\gamma_j\tilde{X}_j(t)\right)}, \\ f_j''\left(\tilde{X}_j(t)\right) = \dfrac{1}{1 + \exp\left(-\gamma_j\tilde{X}_j(t)\right)}, \end{cases} \tag{3.4}$$

the gradients are as follows:

$$\begin{cases} \nabla_{\omega_j} f_j'\left(\tilde{X}_j(t)\right) = G_j'(t) = \gamma_j\left(1 - \left(f_j'\left(\tilde{X}_j(t)\right)\right)^2\right)X_j(t) \\ = \gamma_j\left(1 - \hat{x}_j^2(t)\right)X_j(t), \\ \nabla_{\omega_j} f_j''\left(\tilde{X}_j(t)\right) = G_j''(t) = \gamma_j f_j''\left(\tilde{X}_j(t)\right)\left(1 - f_j''\left(\tilde{X}_j(t)\right)\right)X_j(t) \\ = \gamma_j\hat{x}_j(t)\left(1 - \hat{x}_j(t)\right)X_j(t). \end{cases} \tag{3.5}$$

Convergence of the gradient procedures of the type (3.3) is provided over a fairly long interval of variations of the step $\eta_j(t)$. For the determinate case, this parameter must satisfy the conditions $0 < \eta_j(t) < 2/\mathrm{Lp}$ with Lp for the Lipschitz constant of the optimized function, and for the stochastic case it must satisfy the Dvoretsky conditions. It seems natural to choose here the step providing the maximal convergence rate.

Maximization of the function

$$W_j(t) = \|\tilde{\omega}_j(t)\|^2 - \|\tilde{\omega}_j(t-1)\|^2 \tag{3.6}$$

(here $\tilde{\omega}_j(t) = \omega_j - \omega_j(t)$ and ω_j is the optimal vector of synaptic weights) leads to the nonconstructive estimate

$$\eta_j(t) = \frac{(\omega_j - \omega_j(t))^T G_j(t)}{\varepsilon_j(t)\|G_j(t)\|^2}. \tag{3.7}$$

However, if the relationship

$$\begin{aligned} (\omega_j - \omega_j(t))^T G_j(t) &\leqslant f_j(\omega_j^T X_j(t)) - f_j(\omega_j^T(t)X_j(t)) \\ &= x(t) - f_j(\omega_j^T(t)X_j(t)), \end{aligned} \tag{3.8}$$

which is valid for the convex functions, is satisfied, then it follows from (3.7) that

$$0 < \eta_j(t) < \|G_j(t)\|^2. \tag{3.9}$$

By considering the one-step variant of the Marquardt algorithm [24]

$$\omega_j(t+1) = \omega_j(t) + (G_j(t)G_j^T(t) + \rho(t)E)^{-1}G_j(t)\varepsilon_j(t), \tag{3.10}$$

where $\rho(t) > 0$ and E is the identity matrix, and using the well-known relationships

$$\begin{cases} \lim\limits_{\rho(t)\to 0}(G_j(t)G_j^T(t) + \rho(t)E)^{-1} = (G_j(t)G_j^T(t))^+, \\ (G_j(t)G_j^T(t))^+G_j(t) = (G_j^T(t))^+ = G_j(t)\|G_j(t)\|^{-2} \end{cases} \tag{3.11}$$

of the theory of pseudoinverse matrices, one can write the speed-optimal variant of (3.3) [25]

$$\omega_j(t+1) = \omega_j(t) + \frac{x(t) - \widehat{x}_j(t)}{\|G_j(t)\|^2}G_j(t) \tag{3.12}$$

which in the linear case coincides with the Widrow–Hoff algorithm for adjusting the synaptic weights.

We note that for the activation function like (3.4), algorithm (3.12) is represented as

$$\begin{cases} \omega_j'(t+1) = \omega_j'(t) + \dfrac{x(t) - \widehat{x}_j(t)}{\gamma_j(1-\widehat{x}_j^2(t))\|X_j(t)\|^2}X_j(t), \\ \omega_j''(t+1) = \omega_j''(t) + \dfrac{x(t) - \widehat{x}_j(t)}{\gamma_j\widehat{x}_j(t)(1-\widehat{x}_j(t))\|X_j(t)\|^2}X_j(t). \end{cases} \tag{3.13}$$

The following exponentially weighted modification

$$\begin{cases} \omega_j(t+1) = \omega_j(t) + r_j^{-1}(t)\,(x(t) - \widehat{x}_j(t))\,G_j(t), \\ r_j(t) = \alpha r_j(t-1) + \|G_j(t)\|^2, \quad 0 \leqslant \alpha \leqslant 1, \quad r_j(0) = 1, \end{cases} \tag{3.14}$$

which coincides with (2.12) for $\alpha = 0$ and, in the linear case, with the Goodwin–Ramadge–Caines procedure of stochastic approximation [26] for $\alpha = 1$, can be introduced to render additional smoothing properties to algorithm (3.12).

4. ADJUSTMENT ALGORITHM FOR THE NEURONS OF THE SECOND HIDDEN LAYER

In the second layer of the proposed neural network, the outputs of the first layer are joined pairwise as

$$\widehat{y}_j(t) = c_j(t)\widehat{y}_{j-1}(t) + (1 - c_j(t))\widehat{x}_{j+1}(t), \tag{4.1}$$

where $\widehat{y}_0(t) \equiv \widehat{x}_1(t)$, $j = 1, 2, \ldots, d-1$, and the weights $c_j(t)$ define the comparative accuracy of the predictions $\widehat{y}_{j-1}(t)$, $\widehat{x}_{j+1}(t)$ and make the prediction $\widehat{y}_j(t)$ unbiased.

To determine the value of $c_j(t)$ which makes $\widehat{y}_j(t)$ optimal, we introduce $(t \times 1)$-vectors of observations and errors

$$\begin{cases} X(t) = (x(1), x(2), \ldots, x(t))^T, \\ \widehat{Y}_j(t) = (\widehat{y}_j(1), \widehat{y}_j(2), \ldots, \widehat{y}_j(t))^T, \\ \widehat{X}_j(t) = (\widehat{x}_j(1), \widehat{x}_j(2), \ldots, \widehat{x}_j(t))^T, \\ V_j(t) = X(t) - \widehat{Y}_j(t), \; V_{j-1}(t) = X(t) - \widehat{Y}_{j-1}(t), \\ V_{x,j+1}(t) = X(t) - \widehat{X}_{j+1}(t) \end{cases}$$

and write the evident relationship

$$V_j(t) = c_j(t)V_{j-1}(t) + (1 - c_j(t))V_{x,j+1}(t). \tag{4.2}$$

Then, we solve the equation

$$\frac{\partial \|V_j(t)\|^2}{\partial c_j(t)} = 0 \tag{4.3}$$

and obtain

$$\begin{cases} c_j(t) = V_{x,j+1}^T(t)\dfrac{V_{x,j+1}(t) - V_{j-1}(t)}{\|V_{x,j+1}(t) - V_{j-1}(t)\|^2}, \\ 1 - c_j(t) = V_{j-1}^T(t)\dfrac{V_{j-1}(t) - V_{x,j+1}(t)}{\|V_{j-1}(t) - V_{x,j+1}(t)\|^2}. \end{cases} \tag{4.4}$$

Now, one can demonstrate that

$$\begin{cases} \|V_j(t)\|^2 - \|V_{x,j+1}(t)\|^2 = -\dfrac{\left(\|V_{x,j+1}(t)\|^2 - V_{j-1}^T(t)V_{x,j+1}(t)\right)^2}{\|V_{j-1}(t) - V_{x,j+1}(t)\|^2} \leqslant 0, \\ \|V_j(t)\|^2 - \|V_{j-1}(t)\|^2 = -\dfrac{\left(\|V_{j-1}(t)\|^2 - V_{j-1}^T(t)V_{x,j+1}(t)\right)^2}{\|V_{j-1}(t) - V_{x,j+1}(t)\|^2} \leqslant 0, \end{cases} \tag{4.5}$$

that is, the accuracy of the joint prediction $\widehat{y}_j(t)$ can be never worse than that of the joint predictions $\widehat{y}_{j-1}(t)$ and $\widehat{x}_{j+1}(t)$. The weight coefficient $c_j(t)$ defines the "contribution" of $\widehat{y}_{j-1}(t)$ to $\widehat{y}_j(t)$ and, therefore, closeness of the physical process $x(t)$ to $\widehat{y}_{j-1}(t)$ or $\widehat{x}_{j+1}(t)$. Changes in $c_j(t)$ can signal changes in the properties of the sequence $x(t)$, and the vector $C(t) = (c_1(t), c_2(t), \ldots, c_{d-1}(t))^T$ can be used as that of diagnostic attributes.

To operate in the real-time mode, it is advisable to represent (4.4) in the recurrent form. By introducing the notation

$$\begin{cases} E_j(t) = V_{x,j+1}(t) - V_{j-1}(t), \; v_{j-1}(t+1) = x(t+1) - \widehat{y}_{j-1}(t+1), \\ v_{x,j+1}(t+1) = x(t+1) - \widehat{x}_{j+1}(t+1), \; e_j(t+1) = v_{x,j+1}(t+1) - v_{j-1}(t+1), \end{cases}$$

one can finally write

$$\begin{cases} c_j(t+1) = \dfrac{\Gamma_j(t)}{\Gamma_j(t+1)}c_j(t) + \dfrac{v_{x,j+1}(t+1)e_j(t+1)}{\Gamma_j(t+1)}, \\ \Gamma_j(t+1) = \Gamma_j(t) + e_j^2(t+1). \end{cases} \tag{4.6}$$

In some cases, it is recommendable to use in algorithm (4.6) the monitored sequence $x(t)$ and its predictions, rather than the updating signals. Taking into account that

$$\begin{aligned} E_j(t) &= X(t) - \widehat{X}_{j+1}(t) - X(t) + \widehat{Y}_{j-1}(t) = \widehat{Y}_{j-1}(t) - \widehat{X}_{j+1}(t), \\ e_j(t+1) &= x(t+1) - \widehat{x}_{j+1}(t+1) - x(t+1) + \widehat{y}_{j-1}(t+1) \\ &= \widehat{y}_{j-1}(t+1) - \widehat{x}_{j+1}(t+1), \end{aligned}$$

the algorithm for adjustment of the neurons of the second hidden layer is representable as

$$\begin{cases} c_j(t+1) = \dfrac{\Gamma_j(t)}{\Gamma_j(t+1)}c_j(t) + \dfrac{v_{x,j+1}(t+1)(\widehat{y}_{j-1}(t+1) - \widehat{x}_{j+1}(t+1))}{\Gamma_j(t+1)}, \\ \Gamma_j(t+1) = \Gamma_j(t) + (\widehat{y}_{j-1}(t+1) - \widehat{x}_j(t+1))^2. \end{cases} \tag{4.7}$$

5. ADJUSTMENT ALGORITHM FOR THE OUTPUT NEURON

The output layer of the diagnostic network consists of a single neuron T_3 where the outputs $\hat{y}(t) = (\hat{y}_1(t), \hat{y}_2(t), \ldots, \hat{y}_{d-1}(t))^T$ of the second hidden layer are joined as

$$\hat{z}(t) = \sum_{j=1}^{d-1} p_j(t)\hat{y}_j(t) = P^T(t)\hat{y}(t). \tag{5.1}$$

Here, if the constraints

$$\begin{cases} \sum\limits_{j=1}^{d-1} p_j(t) = P^T(t)I = 1, \\ p_j(t) \geqslant 0, \quad j = 1, 2, \ldots, d-1, \end{cases} \tag{5.2}$$

where I is the $((d-1) \times 1)$ vector of unities, are imposed on the elements of the vector $P(t) = (p_1(t), p_2(t), \ldots, p_{d-1}(t))^T$, then they can be treated as the probabilities of certain hypotheses one of which states that the true structure of the process is most close to the structure of the prediction $\hat{y}_j(t)$ having the maximal probability $p_j(t)$.

To determine the diagnostic vector of probabilities $P(t)$, we consider the Lagrangian

$$\begin{aligned} L(P, \lambda, \mu) &= \sum_{i=1}^{t} \left(x(i) - \sum_{j=1}^{d-1} p_j \hat{y}_j(i) \right)^2 + \lambda \left(\sum_{j=1}^{d-1} p_j - 1 \right) - \sum_{j=1}^{d-1} \mu_j p_j \\ &= \left(X(t) - \hat{Y}(t)P \right)^T \left(X(t) - \hat{Y}(t)P \right) + \lambda(P^T I - 1) - \mu^T P, \end{aligned} \tag{5.3}$$

where $\hat{Y}(t) = \left(\hat{Y}_1(t), \hat{Y}_2(t), \ldots, \hat{Y}_{d-1}(t) \right)$ is the $(t \times (d-1))$ matrix; λ is the indefinite Lagrangian multiplier; and μ is the $((d-1) \times 1)$ vector of the nonnegative indefinite Lagrangian multipliers meeting the conditions for additional nonrigidity.

The vector $P(t)$ can be established either by solving the system of Kuhn–Tucker equations

$$\begin{cases} \nabla_P L(P, \lambda, \mu) = -2\hat{Y}(t)X(t) + 2\hat{Y}^T(t)\hat{Y}(t)P + \lambda I - \mu = 0, \\ \dfrac{\partial L(P, \lambda, \mu)}{\partial \lambda} = P^T I - 1 = 0, \\ \dfrac{\partial L(P, \lambda, \mu)}{\partial \mu_j} = -p_j \leqslant 0; \quad \mu_j \geqslant 0; \quad j = 1, 2, \ldots, d-1, \end{cases} \tag{5.4}$$

or, which is more convenient in real time, by the Arrow–Hurwitz procedure which generally has the form

$$\begin{cases} P(t+1) = P(t) - \gamma_P(t)\nabla_P L(P, \lambda, \mu, t), \\ \lambda(t+1) = \lambda(t) + \gamma_\lambda(t)\partial L(P, \lambda, \mu, t)/\partial \lambda, \\ \mu(t+1) = \Pr_+(\mu(t) + \gamma_\mu(t)\nabla_\mu L(P, \lambda, \mu, t)), \end{cases} \tag{5.5}$$

and in essence is the weight adjustment algorithm for the output neuron T_3. Here, $\gamma_P(t)$, $\gamma_\lambda(t)$, and $\gamma_\mu(t)$ are the parameters of the step of search and $\Pr_+(\cdot)$ is the projector on the positive orthant.

By taking into account (5.4), system (5.5) is representable as

$$\begin{cases} P(t+1) = P(t) + \gamma_P(t)\left(2\xi(t)\hat{y}(t) - \lambda(t)I + \mu(t)\right), \\ \lambda(t+1) = \lambda(t) + \gamma_\lambda(t)(P^T(t)I - 1), \\ \mu(t+1) = \Pr_+(\mu(t) - \gamma_\mu(t)P(t)), \end{cases} \tag{5.6}$$

where $\xi(t) = x(t) - P^T(t)\hat{y}(t) = x(t) - \hat{z}(t)$ is the prediction error of the network output layer.

To optimize the rate of adjustment of the output layer, the first relationship in (5.6) is multiplied from the left by $\hat{y}^T(t)$ and both sides of the resulting equation are subtracted from $x(t)$:

$$x(t) - \hat{y}^T(t)P(t+1) = x(t) - \hat{y}^T(t)P(t) - \gamma_P(t)\Big(2\xi(t)\|\hat{y}(t)\|^2 - \lambda(t)\hat{y}^T(t)I + \hat{y}^T(t)\mu(t)\Big). \tag{5.7}$$

The expression in the left-hand side of (5.7) is the *a posteriori* error $\bar{\xi}(t)$ after one cycle of adjustment:

$$\bar{\xi}(t) = \xi(t) - \gamma_P(t)(2\xi(t)\|\hat{y}(t)\|^2 - \lambda(t)\hat{y}^T(t)I + \hat{y}^T(t)\mu(t)). \tag{5.8}$$

By solving the equation

$$\frac{\partial \bar{\xi}^2(t)}{\partial \gamma_P(t)} = 0, \tag{5.9}$$

one can readily obtain the optimal parameter of search step

$$\gamma_P(t) = \frac{\xi(t)}{2\xi(t)\|\hat{y}(t)\|^2 - \lambda(t)\hat{y}^T(t)I + \hat{y}^T(t)\mu(t)}. \tag{5.10}$$

and finally write the adjustment algorithm of the output layer as

$$\begin{cases} P(t+1) = P(t) + \dfrac{\xi(t)\,(2\xi(t)\hat{y}(t) - \lambda(t)I + \mu(t))}{2\xi(t)\|\hat{y}(t)\|^2 - \lambda(t)\hat{y}^T(t)I + \hat{y}^T(t)\mu(t)}, \\ \lambda(t+1) = \lambda(t) + \gamma_\lambda(t)(P^T(t)I - 1), \\ \mu(t+1) = \mathrm{Pr}_+(\mu(t) - \gamma_\mu(t)P(t)). \end{cases} \tag{5.11}$$

One can easily see that if conditions (5.2) are satisfied in the course of learning, then algorithm (5.11) assumes automatically the form

$$P(t+1) = P(t) + \frac{x(t) - P^T(t)\hat{y}(t)}{\|\hat{y}(t)\|^2}\hat{y}(t) \tag{5.12}$$

which is the Widrow–Hoff algorithm that gained wide acceptance in the theory of artificial neural networks [27].

Therefore, the proposed artificial neural network is an extension of the multilayer structures which—along with prediction which is traditional for the theory of artificial neural networks—enables early real-time detection of the imbalances.

6. RESULTS OF MODELING

Example 1. We consider by way of example the problem of prediction and detection of variations in the properties of the system obeying the difference equation

$$x(t) = 0.3\,x(t-1) + 0.6\,x(t-2) + f(u(t)),$$

where the nonlinear function $f(u(t)) = u^3(t) + 0.3\,u^2(t) - 0.4\,u(t)$ with $u(t) = \sin(2\pi t/250)$ until $t = 250$ and $u(t) = \sin(2\pi t/250) + \sin(2\pi t/250)$ after $t = 250$.

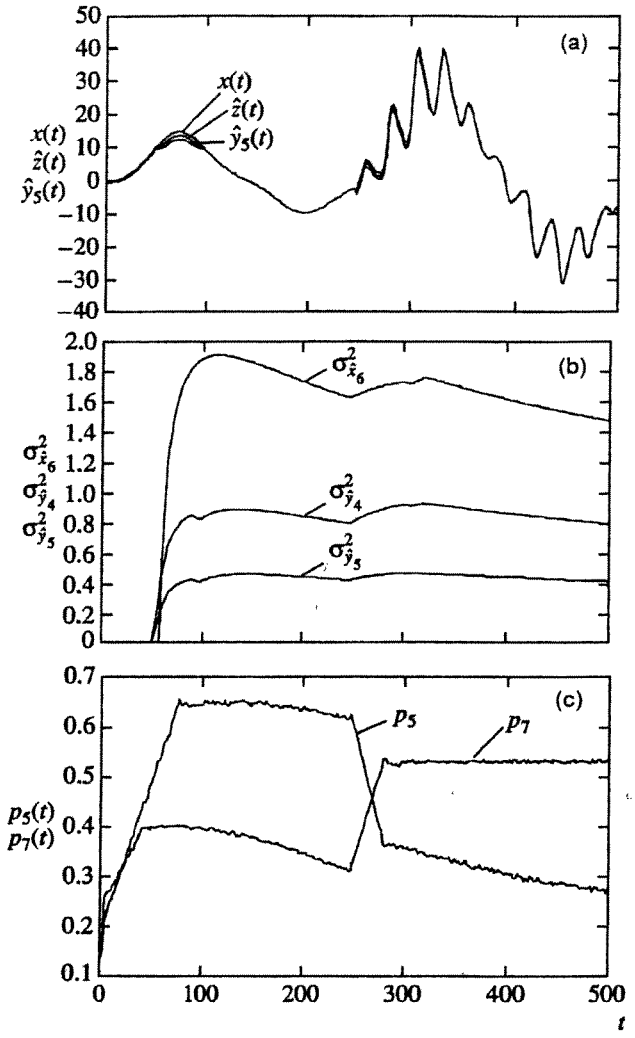

Fig. 2. (a) Diagnosed sequence and its predictions. (b) Variances of the prediction errors on some neurons of the hidden layers. (c) Components of the diagnostic vector $P(t)$: $p_5(t)$ and $p_7(t)$.

The problem is solved using the diagnostic neural network with $d = 7$. To adjust the synaptic weights of the neurons of the first hidden layer, algorithm (3.13) with the sigmoid activation function is used. The neurons of the second hidden layer are adjusted using algorithm (4.7); and the output neuron is adjusted using procedure (5.12). Preliminary training of the network was carried out only over 500 steps using the input sequence of random variables uniformly distributed over the interval $[-1, 1]$.

Figure 2a shows the diagnosed sequence and its predictions obtained at the output of the fifth neuron of the second hidden layer and at the output neuron of the entire network. One can readily see that the prediction error remains small even when the diagnosed sequence changes and the prediction error of the output neuron is even smaller than that of the fifth neuron of the second hidden layer. One can see from Fig. 2b that the variance of the prediction error $\sigma^2_{\hat{y}_5}(t)$ is smaller than the variances of the prediction errors $\sigma^2_{\hat{x}_6}(t)$ and $\sigma^2_{\hat{y}_4}(t)$, which corroborates the theoretical conclusions from (4.5). At the instant of changes in the monitored sequence, the variances of prediction errors increase insignificantly, which is the result of good approximating properties of the neural network. As one can see from Fig. 2c, the components of the diagnostic vector $P(t)$ are modified at the instant of changes in sequence $x(t)$. The value of $p_5(t)$ decreases, and $p_7(t)$, on

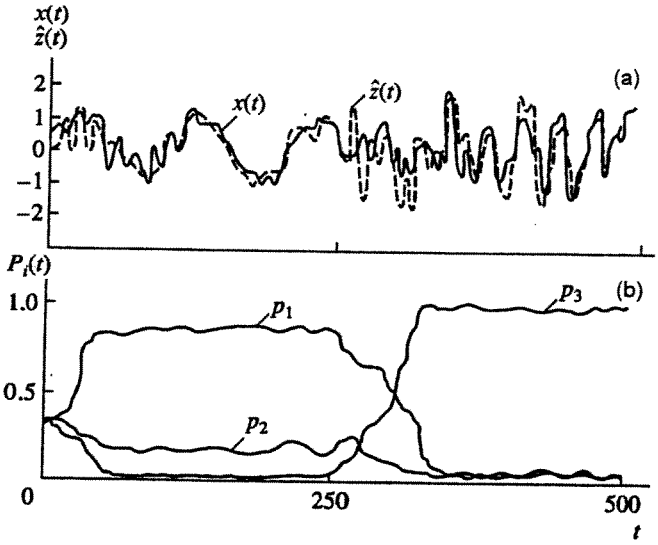

Fig. 3. (a) Diagnosed sequence and its prediction at the output neuron. (b) Diagnostic vector $P(t)$.

the contrary, increases, which is indicative of the fact that the network recognizes changes in the properties of the sequence $x(t)$. Figure 2c shows only two main components of the vector $P(t)$ to which two states of the system—before and after the changes in the properties of the diagnosed system—correspond. The rest of the components of $P(t)$ are close to zero over the entire course of modeling.

Example 2. Let us consider one more example where, in contrast to the above example, the diagnosed signal is distorted by additive noise, which complicated prediction, and the network is not trained in advance.

Let at the first 250 steps of modeling the diagnosed signal obey the model

$$x(t) = 0.93 \sin 2\pi f t + \zeta(t),$$

where $f = 10$ and $\zeta(t)$ is the sequence of random variables with zero mean and variance 0.77. At the 251st step of modeling, changes take place in the signal which results in one more harmonic, the model becoming as follows:

$$x(t) = 0.93 \sin 2\pi\, 10\, t + 0.84 \sin 2\pi\, 24\, t + \zeta(t).$$

To determine the precise number of the required input neurons, we rewrite the last model in the operator form relative to the back-shift operator:

$$\prod_{j=1}^{2} \left(1 - 2\beta_j z^{-1} + z^{-2}\right) x(t) = \zeta(t),$$

where $\beta_j = \cos 2\pi f_j$; and upon returning to the time domain, we obtain

$$x(t) = 2\beta_1 x(t-2) + \beta_2(x(t-1) + x(t-3)) - x(t-4) + \zeta(t).$$

The maximal number of delays of $x(t)$ is four. Therefore, it suffices to have only four neurons in the input layer of the network. Although the above model describes in precise terms the monitored sequence, we make use of the diagnostic network described in this paper.

The results of modeling are depicted in Fig. 3. Figure 3a shows the diagnosed sequence and its prediction at the output neuron of the network. The accuracy of prediction here is worse than in the above example where the network was trained in advance. The graphs of the components of the vector $P(t)$ are shown in Fig. 3b. At the first 250 steps, the component $p_1(t)$ has the greatest value, which is due to taking into account the first two backward-shifted values of the diagnosed sequence which suffice for describing one harmonic. After the 250th step, the components of $P(t)$ are modified and now $p_3(t)$ has the greatest value and $p_1(t)$ and $p_2(t)$ approach zero. Additionally, the rate of detecting the changes does not worsen as compared with the above example.

7. CONCLUSIONS

Although the quality of prediction is worsened if the network was not trained in advance and the signal is distorted by noise, the properties of the network that are related with the rate of detecting changes in the sequence remain the same. On the whole, model examples corroborate the theoretical results obtained in this paper. The possibility of using the proposed network for prediction and detection of changes in the properties of sequences described by the nonlinear equation of autoregression moving average and sequences reducible to the models of the type of nonlinear autoregression moving average is shown.

REFERENCES

1. Basseville, M., Benveniste, A., *et al.*, Eds., *Detection of Abrupt Changes in Signals and Dynamical Systems*, Berlin: Springer, 1986. Translated under the title *Obnaruzhenie izmeneniya svoistv signalov i dinamicheskikh sistem*, Moscow: Mir, 1989.

2. Romberg, T.M., Black, J.L., and Ledwidge, T.J., *Signal Processing for Industrial Diagnostics*, Chichester: Wiley, 1996.

3. Basseville, M. and Nikiforov, I., *Detection of Abrupt Changes. Theory and Application*, Englewood Cliffs: Prentice Hall, 1993.

4. Kerestencioglu, F., *Change Detection and Input Design in Dynamical Systems*, Taunton, UK: Research Studies Press, 1993.

5. Pouliezos, A.D. and Stavrakakis, G.S., *Real Time Fault Monitoring of Industrial Processes*, Dordrecht: Kluwer, 1994.

6. Bodyanskii, E.V. and Rudneva, I.A., On One Adaptive Algorithm to Detect Mismatches in Random Sequences, *Avtom. Telemekh.*, 1995, no. 10, pp. 101–106.

7. Bodyanskii, E.V., Pliss, I.P., and Solov'eva, T.V., Generalized Adaptive Prediction of Multidimensional Random Sequences, *Dokl. Akad. Nauk SSSR*, 1989, ser. A, no. 9, pp. 73–75.

8. Vorob'ev, S.A., Generalized Adaptive Prediction of One-Dimensional Random Processes, *Proc. Second All-Ukrainian Conf. of Young Researchers, Mathematics*, 1995, Kiev, vol. 1, pp. 33–40.

9. Venkatasubramanian, V. and Chan, K., A Neural Network Methodology for Process Fault Diagnosis, *AIChE J.*, 1989, vol. 35, pp. 1993–2002.

10. Naudi, R.S., Zafiriou, E., and McAvoy, T.J., Use of Neural Networks for Sensor Failure Detection in a Control System, *IEEE Control Syst. Mag.*, 1990, vol. 10, pp. 49–55.

11. Yamashina, H., Kumamoto, H., Okumura, S., and Ikesaki, T., Failure Diagnosis of a Servovalve by Neural Networks with New Learning Algorithm and Structure Analysis, *Int. J. Prod. Res.*, 1990, vol. 28, no. 6, pp. 1009–1021.

12. Sorsa, T., Koivo, H.N., and Koivisto, H., Neural Networks in Process Fault Diagnosis, *IEEE Trans. Syst. Man Cybern.*, 1991, vol. 21, no. 4, pp. 815–825.

13. Ray, A.K., Equipment Fault Diagnosis—A Neural Network Approach, *Comput. Industry*, 1991, vol. 16, pp. 169–177.

14. Sorsa, T. and Koivo, H.N., Application of Artificial Neural Network in Process Fault Diagnosis, *Automatica*, 1993, vol. 29, no. 4, pp. 843–849.

15. Bodyanskii, E., Vorob'ev, S., Lamonova, N., and Shtefan, A., Detection of Changes in the Properties of Stochastic Sequences by Artificial Neural Networks, *ASU Prib. Avtom.*, 1997, vol. 106, pp. 75–79.

16. Connor, J.T., Martin, R.D., and Atlas, L.E., Recurrent Neural Networks and Robust Time Series Prediction, *IEEE Trans. Neural Networks*, 1994, vol. 5, no. 1, pp. 240–254.

17. Aussem, A., Murtagh, F., and Sarazin, M., Dynamical Recurrent Neural Networks—Towards Environmental Time Series Prediction, *Int. J. Neural Syst.*, 1995, vol. 6, no. 2, pp. 145–170.

18. Pham, D.T. and Liu, X., *Neural Networks for Identification, Prediction and Control*, London: Springer, 1995.

19. Chang, E.S., Chen, S., and Mulgrew, B., Gradient Radial Basis Function Networks for Nonlinear and Nonstationary Time Series Prediction, *IEEE Trans. Neural Networks*, 1996, vol. 7, no. 1, pp. 190–194.

20. Cybenko, G., Approximation by Superposition of the Sigmoidal Function, *Math. Control Signals Syst.*, 1989, vol. 2, no. 4, pp. 303–314.

21. Hornik, K., Approximation Capabilities of Multilayer Feedforward Networks, *Neural Networks*, 1991, no. 4, pp. 251–257.

22. Narendra, K.S. and Parthasarathy, K., Identification and Control of Dynamical Systems Using Neural Networks, *IEEE Trans. Neural Networks*, 1990, vol. 1, no. 1, pp. 4–26.

23. Narendra, K.S., Adaptive Control of Dynamical Systems Using Neural Networks, *Handbook of Intelligent Control: Neuro, Fuzzy and Adaptive Approaches*, White, D. A. and Sofge, D. A., Eds., New York: Van Nostrand, 1992, pp. 141–183.

24. Marquardt, D., An Algorithm for Least Squares Estimation of Nonlinear Parameters, *SIAM J. Appl. Math.*, 1963, no. 11, pp. 431–441.

25. Bodyanskii, E.V., Adaptive Algorithms for Identification of Nonlinear Controlled Plants, *ASU Prib. Avtom.*, 1987, vol. 81, pp. 43–46.

26. Goodwin, G.C., Ramadge, P.J., and Caines, P.E., A Globally Convergent Adaptive Predictor, *Automatica*, 1981, vol. 17, no. 1, pp. 135–140.

27. Rojas, R., *Neural Networks. A Systematic Introduction*, Berlin: Springer, 1996.

This paper was recommended for publication by O.P. Kuznetsov, a member of the Editorial Board