DETECTION OF NARMA-SEQUENCE ORDER USING RECURRENT ARTIFICIAL NEURAL NETWORKS

Ye.V. Bodyanskiy*, S.A. Vorobyov**, A. Stephan***

*Dept. of Technical Cybernetics, Kharkov State Technical University of Radio Electronics, 310726, av. Lenina, 14, Kharkov, Ukraine, e-mail: bodya@kture.kharkov.ua

**Control Systems Research Laboratory, Kharkov State Technical University of Radio Electronics, 310726, av. Lenina, 14, Kharkov, Ukraine, e-mail: svor@kture.kharkov.ua and fax: +38 0572 409113

***Dr. Stephan&Partner, System- und Softwarehaus, Grenzhammer 8, D-98693, Ilmenau, Germany, e-mail: Andreas.Stephan@rz.tu-ilmenau.de

Keywords: Artificial neural networks, adaptation, fault detection, non-linear black box modelling

Abstract

Recently artificial neural networks have been advocated as a possible technique for fault detection. Neural networks are noise tolerant and their ability to generalise the knowledge as well as to adapt during use are extremely interesting properties. The approach to solution of properties changes detection problem for stochastic sequences circumscribed by the nonlinear equations of an autoregressive-moving average is proposed in the paper. The architecture of multilayer recurrent neural network and algorithms of neuron parameter tuning ensuring maximum speed of learning process are proposed. The virtue of the proposed approach is the possibility of stochastic sequences of any structure diagnosing, high speed and computing simplicity.

1 Introduction

The early detection and diagnosis of faults in industry and ecology monitoring is important from the viewpoint of plant safety as well as reduced manufacturing costs. The conventional method of fault detection is to use static and dynamic model of the process [Iser1, Fran1], rule-based expert systems [Patt1, Iser2], pattern recognition [Deit1].

In recent years artificial neural networks have been advocated as a possible technique for fault diagnosis too. Artificial neural networks can handle non-linear and underdetermined processes like black box because no process model is needed, but neural networks learn the diagnosis by means of the information of the training data. Neural networks are noise tolerant and their ability to generalise the knowledge as well as to adapt during use are extremely interesting properties.

Researches devoted to applying of neural networks in fault diagnosis have been quite active during the last years [McDu1, Kram1, Sors1]. The classification of individual measurement patterns is not always unique in dynamic situation. But for the particular class of object it is possible to design the fault detection procedure. Autoregressive and moving average (ARMA) model is common description of dynamical objects. The ARMA model cannot be used for non-linear dynamical systems and processes, which are of greatest interest in the field of fault detection. So the non-linear autoregressive and moving average (NARMA) model is quite satisfying for diagnosis problem of real non-linear objects.

The architecture of multilayer recurrent artificial neural networks and algorithms of networks parameters tuning combining the multimodel approach and approximation property of predicting neural networks with non-linear activation functions are proposed in the paper. The changes of stochastic sequence properties are fixed with the help of diagnosing vector with elements corresponding to synaptic weights of output neuron.

2 Architecture of the diagnosing neural network

The proposed architecture of diagnosing recurrent neural network is shown on fig. 1 and represents a network of elementary neurones which were differed by an aspect of activation functions and tuning algorithms being in common case the gradient procedures of unconditional or conditional optimisation.

Inspected stochastic sequence \( \{x(t)\}, t=1,2,... \) moves on the input layer of the network representing a line-up of the unit delay operators \( z^{-1} : (z^{-1}x(t) = x(t-1)) \). Therefore, delayed values of time series \( x(t-1), x(t-2),..., x(t-d) \) are formed on the output of this layer. Thus, increasing of delay value \( d \) improves diagnosing possibilities of neural networks.

The first hidden layer is formed by neurones of a McCulloch-Pitts type. The delayed values of the inspected sequence \( x(t) \) are moved on summing inputs of these neurones. The delayed
values of the prediction \( \hat{x}_j(t) \), \( j = 1,2,\ldots,d \) are moved on the same neurone of a feedback.

The inputs of neurones \( T_1 \) correspond to inputs of synaptic weights tuning algorithm and \( f_j(\bullet) \) are nonlinear activation functions of neurones. The results of signal \( x(t) \) handling by neurones of the first hidden layer are the prediction estimates on their outputs

\[
\begin{align*}
\hat{x}_1(t) &= f(x(t-1), \hat{x}_1(t-1)), \\
\hat{x}_2(t) &= f(x(t-1), x(t-2), \hat{x}_2(t-1), \hat{x}_3(t-2)), \\
&\vdots \\
\hat{x}_d(t) &= f(x(t-1), \ldots, x(t-d), \hat{x}_d(t-1), \ldots, \hat{x}_d(t-d)).
\end{align*}
\]

The prediction estimates appropriate to the nonlinear autoregressive-moving average process (NARMA-process) [Chng1] from 1 up to \( d \) order. The network solves tasks of the NARMA-process order definition and change detection in real time.

Neurones of the second hidden layer \( T_2 \) join outputs of neurones of the first hidden layer \( T_1 \) for deriving estimates \( \hat{y}_j(t) \), \( j = 1,2,\ldots,d-1 \)

\[
\begin{align*}
\hat{y}_1(t) &= \varphi(\hat{x}_1(t), \hat{x}_2(t), c_1), \\
\hat{y}_2(t) &= \varphi(\hat{y}_1(t), \hat{x}_3(t), c_2, c_1), \\
&\vdots \\
\hat{y}_{d-1}(t) &= \varphi(\hat{y}_{d-2}(t), \hat{x}_{d-1}(t), c_{d-1}, c_{d-2}, \ldots, c_1),
\end{align*}
\]

and weighting coefficients \( c_j \) that define contribution of joined \( \hat{y}_{j-1}(t), \hat{y}_{j+1}(t) \) to the integrated \( \hat{y}_j(t) \) predictions. The weighting coefficients vector \( C(t) = (c_1(t), c_2(t), \ldots, c_{d-1}(t))^T \) describes quality of predictions achieved in the second hidden layer. Modification of relations between its elements testifies to a modification of a structure and parameters of an inspected sequence \( x(t) \).

The output layer of network consists of one neurone \( T_3 \). Output of this neurone is the prediction \( \hat{z}(t) \) and vector of diagnostic indications \( P(t) = (p_1(t), p_2(t), \ldots, p_{d-1}(t))^T \). Elements of \( p_j(t) \) correspond to probabilities that “true” condition of the process \( x(t) \) is described by estimation \( \hat{y}_j(t) \) in the best way. A maximum value of \( p_j(t) \) determines an order of a diagnosed NARMA-sequence at

---

**Fig. 1** Architecture of the Diagnosing Neural Network
instant $t$ and continuous improvement of a vector $P(t)$ using appropriate algorithms of neurones tuning allows discovering the fault of the process $x(t)$ moment.

3 Algorithm of the first hidden layer neurones tuning

The output signal of $j$-th neurone of the first hidden layer can be represented as

$$
\hat{x}_j(t) = f_j \left( \sum_{i=1}^{j} \omega_{ji}(t)x(t-i) + \sum_{j=1}^{j} \omega_{ji+1}(t)x(t-i) + \omega_{jo}(t) \right) = f_j (\omega_j^T(t)X_j(t)) = f_j (\tilde{X}_j(t)),
$$

where $f_j(\bullet)$ is activation function of $j$-th neurone;

$$
\omega_j(t) = (\omega_{jo}(t),\omega_{j1}(t),... ,\omega_{j1}(t),\omega_{j2}(t))^T
$$

is $((2j+1)\times 1)$-vector of customised synaptic weights; $X_j(t) = (x(t-1),...,x(t-j),\hat{x}_j(t-1),...$,$\tilde{x}_j(t-j))^T$ is vector of generalised inputs; $\tilde{X}_j(t) = \omega_j^2(t)X_j(t)$, $j=1,2,...,d$; $t=1,2,...$ is flowing discrete time.

The expression (1) represents exposition of a nonlinear autoregressive-moving average $j$-th order stochastic sequence, and, as it was marked in [Auss1], just the definition of a “$j$” value is the most complicated problem.

Let us introduce an error of $j$-th neurone prediction

$$
E_j(t) = x(t) - \hat{x}_j(t) = x(t) - f_j (\tilde{X}_j(t)). \tag{2}
$$

Then it is possible to write the gradient procedure of synaptic weights tuning as [Nare1]:

$$
\begin{align*}
\omega_j(t+1) &= \omega_j(t) + \eta_j(t)E_j(t) \nabla_{\omega_j} f_j (\tilde{X}_j(t)) = \\
&= \omega_j(t) + \eta_j(t)E_j(t)G_j(t), \\
\nabla_{\omega_j} f_j (\tilde{X}_j(t)) &= G_j(t) = \gamma_j f_j (\tilde{X}_j(t))(1 - f_j (\tilde{X}_j(t)))X_j(t) = \gamma_j \tilde{x}_j(t)(1 - \tilde{x}_j(t))X_j(t),
\end{align*}
\tag{3}
$$

where $\eta_j(t)$ is search step parameter; $\nabla_{\omega_j} f_j (\tilde{X}_j(t)) = G_j(t)$ is gradient of synaptic weights sigmoidal activation function $f_j (\tilde{X}_j(t))$ :

$$
f_j (\tilde{X}_j(t)) = \frac{1}{1 + \exp(-\gamma_j \tilde{x}_j(t))}.
$$

The convergence of gradient procedures of type (3) is ensured in a rather broad interval of a step parameter $\eta_j(t)$ variation.

Let us consider one-step variant of Marquardt algorithm [Marq1]:

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

where $\rho(t) > 0$; $E$ is unity matrix. Using well known relations for pseudoinverse matrices:

$$
\lim_{\rho(t)\to 0} (G_j(t)G_j^T(t) + \rho(t)E)^{-1} = (G_j(t)G_j^T(t))^*,
$$

we can write variant of algorithm (3) with highest rate of tuning as:

$$
\omega_j(t+1) = \omega_j(t) + \frac{x(t) - \tilde{x}_j(t)}{\|G_j(t)\|^2} G_j(t) = \\
= \omega_j(t) + \frac{x(t) - \tilde{x}_j(t)}{\gamma_j \tilde{x}_j(t)(1 - \tilde{x}_j(t))\|X_j(t)\|^2} X_j(t). \tag{5}
$$

Algorithm (5) is identical to Widrow-Hoff synaptic weights tuning algorithm in linear case. To supply additional smoothing properties to algorithm (5) it is possible to introduce exponential-weight modification:

$$
\begin{align*}
\omega_j(t+1) &= \omega_j(t) + r_j(t)(x(t) - \tilde{x}_j(t))G_j(t), \\
r_j(t) &= a r_j(t-1) + \|G_j(t)\|^2, \tag{6}
\end{align*}
$$

that is conterminous with (5) for $\alpha = 0$ and with stochastic approximation Goodwin-Ramadge-Caines procedure [Good1] for $\alpha = 1$.

4 Algorithm of the second hidden layer neurones tuning

The joining of predictions $\hat{y}_{j-1}(t)$ and $\hat{x}_{j+1}(t)$ is used for calculation of prediction $\hat{y}_j(t)$ . So, the model for $\hat{y}_j(t)$ calculation is of the following form:

$$
\hat{y}_j(t) = c_j(t)\hat{y}_{j-1}(t) + (1 - c_j(t))\hat{x}_{j+1}(t), \tag{7}
$$

where $\hat{y}_0(t) \equiv \hat{x}_1(t), j = 1,2,...,d-1$, and weight $c_j(t)$ sets a comparative exactitude of the predictions $\hat{y}_{j-1}(t)$ and $\hat{x}_{j+1}(t)$ and ensure also an unbiasedness of the prediction $\hat{y}_j(t)$.
To find value of \( c_j(t) \) ensuring an optimality of \( \hat{y}_j(t) \) let us enter \((t \times 1)\)–vectors of observations and errors.

\[
\begin{align*}
X(t) &= (x(1), x(2), \ldots, x(t))^T, \\
\hat{Y}_j(t) &= (\hat{y}_j(1), \hat{y}_j(2), \ldots, \hat{y}_j(t))^T, \\
\hat{X}_j(t) &= (\hat{x}_j(1), \hat{x}_j(2), \ldots, \hat{x}_j(t))^T, \\
V_j(t) &= X(t) - \hat{Y}_j(t), V_{j-1}(t) = X(t) - \hat{Y}_{j-1}(t), \\
V_{x,j+1}(t) &= X(t) - \hat{X}_{j+1}(t).
\end{align*}
\]

Then solving the equation

\[
\frac{\partial}{\partial c_j(t)} \frac{1}{2} \|V_j(t)\|^2 = 0,
\]

we obtain

\[
\begin{align*}
c_j(t) &= V_{x,j+1}(t) \frac{V_{x,j+1}(t) - V_{j-1}(t)}{\|V_{x,j+1}(t) - V_{j-1}(t)\|^2}, \\
1 - c_j(t) &= V_{j-1}(t) \frac{V_{j-1}(t) - V_{x,j+1}(t)}{\|V_{j-1}(t) - V_{x,j+1}(t)\|^2}.
\end{align*}
\]

It is possible to show that

\[
\begin{align*}
\|V_j(t)\|^2 - \|V_{x,j+1}(t)\|^2 &= -\frac{\left(\|V_{x,j+1}(t)\|^2 - \|V_{j-1}(t)\|^2\right)^2}{\|V_{j-1}(t) - V_{x,j+1}(t)\|^2} \leq 0, \\
\|V_j(t)\|^2 - \|V_{j-1}(t)\|^2 &= -\frac{\left(\|V_{j-1}(t)\|^2 - \|V_{x,j+1}(t)\|^2\right)^2}{\|V_{j-1}(t) - V_{x,j+1}(t)\|^2} \leq 0.
\end{align*}
\]

This fact show that the integrated prediction \( \hat{y}_j(t) \) can be never worse then the joined predictions \( \hat{y}_{j-1}(t) \) and \( \hat{x}_{j+1}(t) \). Weighting coefficient \( c_j(t) \) sets “contribution” of \( \hat{y}_{j-1}(t) \) in \( \hat{y}_j(t) \) and, therefore, proximity of the real process \( x(t) \) to \( \hat{y}_{j-1}(t) \) or \( \hat{x}_{j+1}(t) \). A modification of \( c_j(t) \) can be considered as an indicator of sequence \( x(t) \) property changes and vector \( C(t) = (c_i(t), \ldots, c_{j-1}(t))^T \) can be used as a vector of diagnostic indications.

For real-time application it is expedient to present (9) in the recursive form as:

\[
\begin{align*}
c_j(t + 1) &= \frac{\Gamma_j(t)}{\Gamma_j(t + 1)} c_j(t) + \\
v_{x,j+1}(t + 1) e_j(t + 1) \\
\Gamma_j(t + 1) &= \frac{\Gamma_j(t) + e_j^2(t + 1)}{\Gamma_j(t + 1)},
\end{align*}
\]

where \( v_{x,j+1}(t + 1) = x(t + 1) - \hat{x}_{j+1}(t + 1) \), \( e_j(t + 1) = v_{x,j+1}(t + 1) - v_{j-1}(t + 1) \).

Often it is more convenient to use analysed sequence \( x(t) \) and its predictions in algorithm (11) rather than innovating signal. Then the algorithm of the second hidden layer neurones tuning can be represented as:

\[
\begin{align*}
c_j(t + 1) &= \frac{\Gamma_j(t)}{\Gamma_j(t + 1)} c_j(t) + \\
v_{x,j+1}(t + 1) (\hat{y}_{j-1}(t + 1) - \hat{x}_{j+1}(t + 1)) \\
\Gamma_j(t + 1) &= \frac{\Gamma_j(t) + (\hat{y}_{j-1}(t + 1) - \hat{x}_{j+1}(t + 1))^2}{\Gamma_j(t + 1)},
\end{align*}
\]

5 Tuning procedure for output neurone

The output layer of the diagnosing neural network is formed by one neurone \( T_3 \) that joins outs of the second hidden layer by the following way

\[
\hat{z}(t) = \sum_{j=1}^{d-1} P_j(t) \hat{y}_j(t) = P^T(t) \hat{y}(t).
\]

where \( P(t) = (p_1(t), p_2(t), \ldots, p_{d-1}(t))^T, \) \( \hat{y}(t) = (\hat{y}_1(t), \hat{y}_2(t), \ldots, \hat{y}_{d-1}(t))^T. \)

Thus, if there are constraints on the elements of a vector \( P(t) \)

\[
\begin{align*}
\sum_{j=1}^{d-1} P_j(t) &= P^T(t) I = 1, \\
P_j(t) &\geq 0, j = 1, 2, ..., d - 1,
\end{align*}
\]

(here \( I \) is \((d - 1) \times 1\)-vector, that consist of units) they can be interpreted as probabilities of hypothesise that the “true” structure of the process is closest to a structure of the
prediction $\hat{y}_j(t)$ whose probability $p_j(t)$ is maximum. Like approach was used in [Voro1]. For the definition of a diagnostic vector of probabilities $P(t)$ let us enter into consideration a Lagrangian

$$L(P, \lambda, \mu) = \sum_{i=1}^{d} \left( x(i) - \sum_{j=1}^{d-1} p_j \hat{y}_j(i) \right)^2 +$$

$$+ \lambda \left( \sum_{j=1}^{d-1} p_j - 1 \right) - \sum_{j=1}^{d-1} \mu_j p_j = (X(t) - \hat{Y}(t)P)^T \times (15) \times (X(t) - \hat{Y}(t)P) + \lambda(p^T I - 1) - \mu^T P,$$

where $\hat{Y}(t) = (\hat{y}(1), \hat{y}(2), ..., \hat{y}(t))$ is $(t \times (d-1))$-matrix; $\lambda$ is indeterminate Lagrange multiplier; $\mu$ is $(d-1) \times 1$-vector of nonnegative indeterminate Lagrange multipliers that satisfy to conditions of a complementary slackness. Vector $P(t)$ can be found by a solution of Kuhn-Takker system or by the help of Arrow-Hurwitz-Udzawa procedure. This procedure is more convenient for real-time application.

Finally, the output layer neurone-tuning algorithm we can write as:

$$P(t+1) = P(t) + \frac{P(t) - X(t) + \xi(t) \hat{y}(t) - \lambda(t) I + \mu(t)}{2 \xi(t) \hat{y}(t)^T (X(t) - \hat{Y}(t)P)^T (X(t) - \hat{Y}(t)P) + (\lambda(t) + \gamma_\lambda(t)(P^T I - 1)),}$$

$$\lambda(t+1) = \lambda(t) + \gamma_\lambda(t)(P^T I - 1),$$

$$\mu(t+1) = Pr_+(\mu(t) - \gamma_\mu(t)P(t)),$$

where $\xi(t) = x(t) - P^T(t) \hat{y}(t) = x(t) - \hat{y}(t)$ is prediction error of the network output layer; $\gamma_\lambda(t), \gamma_\mu(t)$ are search step parameters; $Pr_+$ is projector on a positive orthant.

It is easy to see that if the condition (14) will be carried out in learning process then the algorithm (16) can be written as the following:

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

(17)

Algorithm (17) is similar to well known in the theory of artificial neural networks Widrow-Hoff algorithm [Rojal1].

6 Simulation results

In this section simulation results of nonlinear system prediction and fault detection using diagnosing neural network are presented. Let us consider the system described by following difference equation

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

where the unknown function has the form $f(u(t)) = u^3(t) + 0.3u^2(t) - 0.4u(t)$ (here $u(t) = \sin(2\pi t) / 250$ when k<250 and $u(t) = \sin(2\pi t) / 250 + \sin(2\pi t) / 25$ when k>250).

For prediction and fault detection we use the diagnosing neural network with d=7. For tuning of the first and second hidden layer synaptic weights the algorithm (5) and (12) are used. Output neurone is tuned using algorithm (17). Training procedure of the neural network was continued only for 500 time steps using the random input with amplitude uniformly distributed in the interval [-1,1]. Fig. 2 shows the output of the system and predictions from 5-th neurone of second hidden layer and output neurone.

Fig.2. Sequence $x(t)$ and predictions $\hat{y}_5(t), \hat{y}_6(t)$.

As can be seen from the figure, the prediction errors are small even when the input is changed. Prediction error of output neurone is smaller then error of 5-th second hidden layer neurone. As seen from fig. 3, the variances of predictions $\hat{y}_6(t), \hat{y}_4(t)$ error is smaller then output prediction of 5-th neurone error prediction variance.

Fig.3. Variances of predictions $\hat{y}_6(t), \hat{y}_4(t), \hat{y}_3(t)$.

It is agreed to conclusion from expression (10). And as we can see from fig. 4, at the time instant k=250 the components of diagnosing vector of output neurone are changed. On the fig. 4 we show only two basic components of vector $P(t) : p_5(t), p_7(t)$. The vector $P(t)$ components changes indicate the fault in the system.
So the simulation result is agreed to theoretical conclusions that were discussed in the paper.

7 Conclusions

The solution of properties changes detection problem for stochastic sequence that circumscribed by the nonlinear autoregressive-moving average equation using recursive artificial neural network is proposed. The architecture of diagnosing neural network and algorithms with improved rate of tuning are proposed. The recurrent neural network is simple and ensures fast fault detection in real-time application. The simulation results confirm effectiveness of the neural network.

References


