Neurocomputing 230 (2017) 409–416

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Adaptive multivariate hybrid neuro-fuzzy system and its on-board fast learning

Ye Bodyanskiy[a], O. Vynokurova[a,*], G. Setlak[b], D. Peleshko[c], P. Mulesa[d]

[a] Kharkiv National University of Radio Electronics, 14 Nauky av., 61166 Kharkiv, Ukraine
[b] Rzeszow University of Technology, 12 Al. Powstancow Warszawy, 35-959 Rzeszow, Poland
[c] Lviv Polytechnic National University, 12 S. Bandery st., 79013 Lviv, Ukraine
[d] Uzhhorod National University, 3 Narodna Square, 88000 Uzhhorod, Ukraine

ARTICLE INFO

Communicated by J. Zhang

Keywords:
Data mining
Computational intelligence
Multivariate hybrid neuro-fuzzy system
Adaptive learning
Prediction
On-board applications

ABSTRACT

In the paper the multivariate adaptive hybrid neuro-fuzzy system is proposed that allows to process nonstationary information disturbed by noises in sequential mode and also has smaller number of tuned parameters comparatively with known neuro-fuzzy systems. This proposed system can be used in on-board applications and, first of all, industrial plants, smart homes (energy management, climate control, home electronic devices including security system, etc.).

1. Introduction

Nowadays the computational intelligence methods and systems are widespread for solving of different Data Mining tasks, problems of intelligent control, prediction, identification, pattern recognition, etc. [1–4] under conditions of uncertainty, nonlinearity, stochasticity, chaotic states, different kinds of disturbances and noises due to their universal approximation properties and learning possibilities based on data that describe the operation of investigated signal, process or plant.

Currently the most known and popular approaches are connected with the artificial neural networks such as multilayer perceptrons that are learned using backpropagation tuning algorithms. Nevertheless, the training set must be defined a priori, and the learning process is implemented using many epochs of the synaptic weights training. In this case, we cannot use such systems for solving tasks in sequential mode, when the data are fed to the inputs in a sequential order in real time.

At this time systems of computational intelligence are used widely in on-board applications and, first of all, industrial plants, smart homes [5–11] (energy management, climate control, home electronic devices including security system, etc). These tasks need increased learning speed which has to take place in real time, the simplicity of implementation, the possibility of operation under different kind of disturbances and noise conditions and also nonstationary changeable environment.

Generally, implementing of sequential learning process is possible for neural networks, whose output signal depends linearly from tuned synaptic weights, for example, Radial Basis Function Networks (RBFN) [1–4,12–14] and Normalized Radial Basis Function Networks (NRBFN) [15–17], however their using is often complicated by, so called, curse of dimensionality. In addition, the key moment is not computational complexity, but a problem is obtaining of data sets from the real plant that can be too small for estimating of large synaptic weights number.

Neuro-fuzzy systems that combine the learning abilities of neural networks and transparency- interpretability of the soft computing methods, have a range of advantages ahead of the conventional neural networks. It should be noticed TSK-system [18–21] and ANFIS [22–25], whose output signal depends linearly from the synaptic weights and has less number of synaptic weights than RBFN or NRBFN. The more complex hybrid systems of computational intelligence are well-known and have improved approximation properties, for example, the hybrid fuzzy wavelet neural networks [26–28], but learning algorithms complexity limits their using in sequential mode.

Hence it is necessary to synthesize a multivariate adaptive hybrid neuro-fuzzy system that allows to process nonstationary information that is disturbed by noises in on-board sequential mode, has smaller number of tuned parameters comparatively with known neuro-fuzzy

* Corresponding author.
E-mail addresses: yevgeniy.bodyanskiy@nure.ua (Y. Bodyanskiy), olena.vynokurova@nure.ua (O. Vynokurova), gsetlak@prz.edu.pl (G. Setlak), dpeleshko@gmail.com (D. Peleshko), ppmulesa@gmail.com (P. Mulesa).

http://dx.doi.org/10.1016/j.neucom.2016.12.042
Received 13 March 2016; Received in revised form 24 November 2016; Accepted 13 December 2016
Available online 18 December 2016
0925-2312/

systems, is simple in the computational implementation (due to the paralleling of the information processing) and don't demand previous defining of the training set, i.e. to implement the learning process started with the first observation, which is fed to the system.

2. Neuro-fuzzy systems by Takagi-Sugeno-Kang and Wang-Mendel

The most popular neuro-fuzzy systems, whose output signal depends linearly from tuning synaptic weights, are Takagi-Sugeno-Kang systems (TSK systems) and its simplified version Wang-Mendel system [29,30] (such system is TSK system of zeroth order). The advantages of Wang-Mendel system are relatively small number of tuning parameters (for example, comparatively to conventional multilayer perceptron) and possibility of using Gauss-Newton optimization procedure for learning of system (the second order optimization algorithms), for example, recurrent least squares method, which characterized by high convergence rate.

The architecture of Wang-Mendel system consists of concatenated layers of information processing: the first layer is fuzzification layer, the second hidden layer is aggregation layer, the third hidden layer is synaptic weights layer, the forth hidden layer is adder units layer, and finally, the output layer is defuzzification layer. The input vector signal $x(k) = (x_1(k), \ldots, x_i(k), \ldots, x_n(k))^T$ (here $k = 1, 2, \ldots$ is discrete instant time) in the first layer, which contained nh membership functions $\mu_{li}(x_i)$ (here $l = 1, 2, \ldots, h$, h is a number of membership functions for each scalar input x_i, $i = 1, 2, \ldots, n$) is exposed to fuzzification. With the result that nh signals are appeared in the output of the first layer, which are fed to the second hidden layer with h multilayer units, which are realized an aggregation operation.

These aggregated signals are fed to the third layer of synaptic weights, which are permanently adjusted under a learning process. In the fourth layer the elementary sum operation of signals from outputs of second and third hidden layers, and finally, in the output layer the defuzzification operation is realized by m elementary division units (here m is the number of system output). In this way, m signals $\hat{y}_p(k)$ ($p = 1, \ldots, m$) are appeared in the output of system, which are response to input signals x_i ($i = 1, 2, \ldots, n$).

From formal point of view, this neuro-fuzzy system implements nonlinear mapping $x \in \Re^n \Rightarrow \hat{y} \in \Re^m$. This nonlinearity is realized by the first layer with nonlinear activation functions $0 < \mu_{li}(x_i(k)) \le 1$, where, as usual, bell-shaped functions are used, such as conventional Gaussians

$$\mu_{li}(x_i(k)) = \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma_i^2}\right). \tag{1}$$

These functions have nonstrictly local receptive fields, due to this fact we can avoid appearing of gaps in fuzzificated space, which is connected with scatter partitioning [16]. It can be also noticed, the centers c_{li} and widths σ_i parameters of Gaussians can be choose either empirically or tuning with computationally tedious error backpropagation algorithms. It is clear, that in this case we cant talk about online learning of system.

A choice of width parameter σ_i can be simplified a little if all input variables are coded in some fixed interval, for example, $0 \le x_i(k) \le 1$, that allows to choose this parameter equivalent for all inputs, where the number of membership functions is the same for each input.

The aggregation layer implements a multiplication operator of one-dimensional membership functions for each input in the form

$$\tilde{x}_l(k) = \prod_{i=1}^{n} \mu_{li}(x_i(k)). \tag{2}$$

In the result of this operation, instead of one-dimensional membership functions, we obtain multidimensional bell-shaped activation functions of radial basis function networks, which allow to implement increasing of input space dimension. If as membership functions are used Gaussians with the same width parameter then output signals of the second hidden layer can be written in form

$$\tilde{x}_l(k) = \prod_{i=1}^{n} \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\| x(k) - c_l \|^2}{2\sigma^2}\right) \tag{3}$$

where $c_l = (c_{l1}, \ldots, c_{li}, \ldots, c_{ln})^T$ is vector of centers parameters of multidimensional activation functions.

In the third layer of synaptic weights, where the main process of learning is implemented, output signals of the second layer are exposed to transformation in form

$$w_{pl}(k-1) \prod_{i=1}^{n} \mu_{li}(x_i(k)) = w_{pl}(k-1)\tilde{x}_l(k) \tag{4}$$

where $w_{pl}(k-1)$ forms $mh \times 1$-vector of synaptic weights, which is computed using $(k-1)$ previous observations, $p = 1, 2, \ldots, m$; $l = 1, 2, \ldots, h$.

In the fourth (the simplest) layer of system, which is formed by adder units, we compute signals in the form

$$\sum_{l=1}^{h} w_{pl}(k-1) \prod_{i=1}^{n} \mu_{li}(x_i(k)) = \sum_{l=1}^{h} w_{pl}(k-1)\tilde{x}_l(k),$$
$$\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k)) = \sum_{l=1}^{h} \tilde{x}_l(k) \tag{5}$$

which are fed to the output layer of defuzzification, where the output signals are computed in form

$$\begin{aligned}\hat{y}_p(k) &= \frac{\sum_{l=1}^{h} w_{pl}(k-1) \prod_{i=1}^{n} \mu_{li}(x_i(k))}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k))} = \frac{\sum_{l=1}^{h} w_{pl}(k-1)\tilde{x}_l(k)}{\sum_{l=1}^{h} \tilde{x}_l(k)} \\ &= \sum_{l=1}^{h} w_{pl}(k-1) \frac{\tilde{x}_l(k)}{\sum_{l=1}^{h} \tilde{x}_l(k)} = \sum_{l=1}^{h} w_{pl}(k-1) \frac{\prod_{i=1}^{n} \mu_{li}(x_i(k))}{\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k))} \\ &= \sum_{l=1}^{h} w_{pl}(k-1)\varphi_l(x(k)) = w_p^T(k-1)\varphi(x(k))\end{aligned} \tag{6}$$

where
$\varphi_l(x(k)) = \prod_{i=1}^{n} \mu_{li}(x_i(k)) (\sum_{l=1}^{h} \prod_{i=1}^{n} \mu_{li}(x_i(k)))^{-1}$,
$w_p(k-1) = (w_{p1}(k-1), \ldots, w_{pl}(k-1), \ldots, w_{ph}(k-1))^T$,
$\varphi(x(k)) = (\varphi_1(x(k)), \ldots, \varphi_l(x(k)), \ldots, \varphi_h(x(k)))^T$.

It can be noticed, that nonlinear transformation, which is realized by Wang-Mendel neuro-fuzzy system, is similar to one that is implemented by normalized radial basis function network, but contains smaller tuning parameters. This fact allows to increase a speed operation of learning process and to simplify a computational implementation.

3. Multivariable hybrid neuro-fuzzy system

Decreasing of number of tuning parameters is provided by using a scatter partitioning of input space. At that it is necessary to notice that in this case in areas, which are disposed from centers of multidimensional membership-activation functions

$$\prod_{i=1}^{n} \exp\left(-\frac{(x_i(k) - c_{li})^2}{2\sigma^2}\right) = \exp\left(-\frac{\| x(k) - c_l \|^2}{2\sigma^2}\right) \tag{7}$$

the provided quality of approximation can be nonsufficient.

The approximation quality can be improved using, for example, grid partition of input space but at that the number of tuning parameters increases rapidly, i.e. the neuro-fuzzy systems advantages are lost ahead of the conventional neural network.

For improving the approximation properties of neuro-fuzzy system we can introduce, so called, nonlinear synapses in the third hidden layer instead of usual synaptic weights w_{pl}, $p = 1, 2, \ldots, m$, $l = 1, 2, \ldots, h$. These nonlinear synapses are building elements of neo-