DEEP HYBRID SYSTEM OF COMPUTATIONAL INTELLIGENCE FOR TIME SERIES PREDICTION

Yevgeniy Bodyanskiy, Iryna Pliss, Dmytro Peleshko, Olena Vynokurova

Abstract: In this paper the deep hybrid system of computational intelligence for time series forecasting is proposed. As the layer of such hybrid system we propose to use the hybrid generalized additive type-2 fuzzy-wavelet-neural network. Proposed deep stacking forecasting hybrid system of computational intelligence is enough simple in computational implementation due to parallelizing process of implemented computations, has high learning rate convergency due to dismissal from errors backpropagation and using the rapid adaptive algorithms for tuning its parameters, has also flexibility due to possibility of tuning the activation-membership functions. Proposed system is aimed for solving the wide range Data Stream Mining tasks, such as forecasting, identification, emulation, classification and pattern recognition in online mode.

Keywords: deep learning, computational intelligence, time series forecasting, hybrid type-2 fuzzy system.

ACM Classification Keywords: I.2 ARTIFICIAL INTELLIGENCE, I.5 PATTERN RECOGNITION.

Introduction

Nowadays hybrid systems of computational intelligence [Rutkowski, 2008; Du and Swamy, 2014; Mumford and Jain, 2009] are wide spread for solving different type of Data Mining problems under uncertainty conditions when the investigated objects or processes are nonstationary, nonlinear, stochastic, chaotic, etc. Among these problems the most complicated task is forecasting because it has highest level of uncertainty, which is defined by multivariance of future facts and events. Specifically, hybrid systems have shown the best results for solving such type of tasks due to universal approximation properties and capability to learning of artificial neural networks [Haykin, 1999; Nelles, 2001], transparency and interpretability of obtained results and high learning rate of tuning synaptic weights of neuro-fuzzy systems [Jang and Mizutani, 1997; Bodyanskiy et all, 2001; Bodyanskiy et all, 2005; Bodyanskiy et all, 2008], including type-2 fuzzy neural networks [Aliev and Guirimov, 2014], the possibility of effective description of nonstationary signal local characteristics due to wavelet neural networks [Alexandridis and Zapranis, 2014] and flexibility of wavelet-neuro-fuzzy systems [Abiyev and
Kaynak, 2008; Bodyanskiy and Vynokurova, 2013], including type-2 fuzzy-wavelet neural networks [Bodyanskiy et al., 2015].

Recent years deep neural networks [LeCun et al., 2015; Schmidhuber, 2015; Goodfellow et al., 2016] attract the attention of many researchers. These networks provide higher quality of data processing including forecasting problem [Langkvist Längkvist et al., 2014; Huang et al., 2014; Setlak et al., 2016] in comparison with the conventional shallow neural networks. But the learning process of deep networks is enough time-consuming and demands heavy span time, leading to impossibility of information processing in real time. Nevertheless, such tasks are appeared quite often, especially in the case when data are fed to the processing in on-line mode and time for processing is strictly constrained. These problems have appeared in the context of Dynamic Data Mining [Lughofer, 2011] and Data Stream Mining [Aggarwal, 2007; Bifet, 2010] and even accelerated deep systems [Bodyanskiy et al., 2016], which are constructed based on conventional architectures, cannot manage to adjust their parameters in a time between two next observations.

In the connection with that topical problem is synthesis of forecasting deep hybrid system of computational intelligence, which allows to process the nonlinear nonstationary time series in on-line mode.

**Architecture of deep forecasting hybrid system**

Figure 1 shows the architecture of proposed forecasting deep systems of computational intelligence, which is a hybrid of cascade-correlation neural network of Fahlman-Lebiere [Fahlman and Lebiere, 1990], cascaded neuro-fuzzy systems [Bodyanskiy et al., 2014; Bodyanskiy and Tyshchenko, 2016; Bodyanskiy et al., 2016a], stacking deep neural networks [Schmidhuber, 2015] and hybrid generalized additive type-2 fuzzy-wavelet-neural network [Bodyanskiy et al., 2015], that form layers of deep system. At that number of layers can be increased during learning process of system.

![Figure 1. Forecasting deep hybrid system of computational intelligence](image-url)
The zero layer of system is formed by elements with pure delay $z^{-1}$, at that the current observation of processed time series $x(k)$ is fed to input of first element. Thus, the zero layer forms prehistory of process $X(k) = (x(k-1), x(k-2), \ldots, x(k-n))^T \in \mathbb{R}^n$, which is fed to input of the first hidden layer $L_1$. The output of this layer is estimated $\hat{x}^{[1]}(k) = f^{[1]}(X(k))$, where $f^{[1]}(X(k))$ is nonlinear transformation which is implemented by HGAT2FWNN [Bodyanskiy et al, 2015]. Input of second layer $L_2$ is signal $X^{[2]}(k) = (X^T(k), \hat{x}^{[1]}(k))^T \in \mathbb{R}^{n+1}$, and its output is $\hat{x}^{[3]}(k)$; to the input of third layer $L_3$ the signal $X^{[3]}(k) = (X^T(k), \hat{x}^{[1]}(k), \hat{x}^{[2]}(k))^T = (X^{[2]}(k), \hat{x}^{[2]}(k))^T \in \mathbb{R}^{n+2}$ is fed, and finally, the vector $X^{[g]}(k) = (X^T(k), \hat{x}^{[1]}(k), \hat{x}^{[2]}(k), \ldots, \hat{x}^{[g-1]}(k))^T = (X^{[g-1]}(k), \hat{x}^{[g-1]}(k))^T \in \mathbb{R}^{n+g-1}$ is fed to input of $g-$th layer $L_g$, and the signal-forecast $\hat{x}^{[g]}(k)$ is formed in the output of system. Such architecture allows to avoid using the errors backpropagation procedures, which are implemented in batch mode and doesn’t allow to realize online data processing. In our case the learning process is involved sequentially layer by layer and each layer is learned by high-speed adaptive algorithms. At that each next layer is distinct from previous layer only additional input, and output signal from previous layer is fed to this input.

Hybrid generalized additive type-2 fuzzy-wavelet-neural network

Figure 2 shows the architecture of hybrid generalized additive type-2 fuzzy-wavelet-neural network (HGAT2FWNN) in first hidden layer $L_1$ with $n$ inputs $x_i(k) = x(k-i)$, $i = 1, 2, \ldots, n$ and single output $\hat{x}^{[1]}(k)$. This system consists of four layers of information processing; the first and second layers are similar to the layers of TSK-neuro-fuzzy system [Takagi and Sugeno, 1985; Sugeno and Kang, 1988; Takagi and Hayashi, 1991]. The only difference is that the odd wavelet membership functions “Mexican Hat”, which are “close relative” of Gaussians, are used instead of conventional bell-shaped Gaussian membership function in the first hidden layer

$$\phi_i(x_i(k)) = (1 - \tau_i^2(k)) \exp(-\tau_i^2(k) / 2)$$

where $\tau_i(k) = (x_i(k) - c_i) \sigma_i^{-1}$; $c_i, \sigma_i$ are the centre and width of the corresponding membership function implying that $\sigma \leq c \leq \sigma$; $\sigma_i \leq \sigma_h$; $i = 1, 2, \ldots, n$; $h = 1, 2, \ldots, h$; $n$ is the input number; $h$ is the membership function number.

It is necessary to note that using the wavelet functions instead of common bell-shaped positive membership functions gives the system more flexibility [Mitaim and Kosko, 1996], and using odd wavelets for the fuzzy reasoning does not contradict the ideas of fuzzy inference, because the negative values of these functions can be interpreted as non-membership levels [Mitaim and Kosko, 1997].
Thus, if the input vector $X(k)$ is fed to the system input, then in the first layer the $hn$ levels of membership functions $\phi_i(x_i(k))$ are computed and in the hidden layer $h$ vector product blocks perform the aggregation of these memberships in the form

$$\tilde{x}_i(k) = \prod_{i=1}^{n} \phi_i(x_i(k)).$$

This means that the input layers of HGAT2FWNN transform the information similarly to the neurons of the wavelet neural networks [Alexandridis and Zapranis, 2014; Bodyanskiy et al, 2005a], which form the multidimensional activation functions
\[ \prod_{i=1}^{n} (1 - r_i^2(k)) \exp(-r_i^2(k)/2) \tag{3} \]

providing a scatter partitioning of the input space.

As a result, the signals in the output of the second layer can be written in the form

\[ \tilde{x}_i(k) = \prod_{i=1}^{n} \left( 1 - \frac{(x_i(k) - c_{\tilde{x}_i})^2}{\sigma_{\tilde{x}_i}^2} \right) \exp \left( -\frac{(x_i(k) - c_{\tilde{x}_i})^2}{2\sigma_{\tilde{x}_i}^2} \right). \tag{4} \]

To provide the required approximation properties, the third layer of the system is formed based on type-2 fuzzy wavelet neuron (T2FWN) [Bodyanskiy and Vynokurova, 2011; Bodyanskiy et al., 2012]. This neuron consists of two adaptive wavelet neurons (AWN) [Bodyanskiy et al., 2005a; Bodyanskiy et al., 2014a], whose prototype is a wavelet neuron of T. Yamakawa [Yamakawa et al., 1998]. Wavelet neuron differs from the popular neo-fuzzy neuron [Yamakawa et al., 1992; Uchino and Yamakawa, 1997] that uses the odd wavelet functions instead of the common triangular membership functions. The use of odd wavelet membership functions, which form the wavelet synapses \( W_{i}, W_{j}, \ldots, W_{n} \), provides higher quality of approximation in comparison with nonlinear synapses of neo-fuzzy neurons. Figure 3 shows the architecture of wavelet-neuron.

In such a way the wavelet neuron performs the mapping in the form

\[ y_{w}^{w}(\tilde{x}(k)) = \sum_{i=1}^{n} f_i(\tilde{x}_i(k)) \tag{5} \]

where \( \tilde{x}(k) = (\tilde{x}_1(k), \ldots, \tilde{x}_i(k), \ldots, \tilde{x}_n(k))^T \), \( y_{w}^{w}(\tilde{x}(k)) \) is the scalar output of wavelet neuron. Each wavelet synapse \( W_{i} \) consists of \( p \) wavelet membership functions \( \tilde{\phi}_j(\tilde{x}_i) \), \( j = 1, 2, \ldots, p \) (\( p \) is a wavelet membership function number in the wavelet neuron) and the same number of the tuning synaptic weights \( w_{j} \). Thus, the transform that is implemented by each wavelet synapse \( W_{i} \) in the \( k \)-th instant of time, can be written in form

\[ f_i(\tilde{x}_i(k)) = \sum_{j=1}^{p} w_{j}(k-1)\tilde{\phi}_j(\tilde{x}_i(k)) \tag{6} \]

(here \( w_{j}(k-1) \) is the value of synaptic weights that are computed based on previous \( k-1 \) observations), and the general wavelet neuron performs the nonlinear mapping in the form

\[ y_{w}^{w}(\tilde{x}(k)) = \sum_{i=1}^{n} \sum_{j=1}^{p} w_{j}(k-1)\tilde{\phi}_j(\tilde{x}_i(k)) \tag{7} \]

i.e., in fact, this is the generalised additive model [Hastie and Tibshirani, 1990] that is characterised by the simplicity of computations and high approximation properties.
Figure 3. Wavelet-neuron architecture

Under uncertain, stochastic or chaotic conditions, it is more effective to use the adaptive wavelet neuron (AWN) instead of common wavelet neuron. The adaptive wavelet neuron is based on the adaptive wavelet function in the form

\[
\phi_{jl}(x_k) = \frac{1}{\sqrt{2\pi \sigma_{\phi}^2}} \exp\left(-\frac{(x_k - c_{\phi})^2}{2\sigma_{\phi}^2}\right)
\]

where \( 0 \leq \alpha_{\phi} \leq 1 \) is the shape parameter of adaptive wavelet function, if \( \alpha_{\phi} = 0 \) it is conventional Gaussian, if \( \alpha_{\phi} = 1 \) it is the wavelet “Mexican Hat”, and if \( 0 < \alpha_{\phi} < 1 \) it is some hybrid activation-membership function (see Figure 4).
Here it should be noted that if the learning process of wavelet neuron of T. Yamakawa is the tuning of the synaptic weights $w_j$, then the learning process of adaptive wavelet neuron consists of the tuning not only synaptic weights but also centres $c_j$, widths $\sigma_j$ and shape parameters $\alpha_j$ of wavelet functions. However, if synaptic weights $w_j$ can be tuned using the second-order optimisation algorithms, such as a recurrent least squares method, then the optimisation of the operation speed in the gradient learning algorithms of $c_j$, $\sigma_j$, $\alpha_j$ is significantly difficult.

To overcome this difficulty, we can set the boundary of possible changes of adaptive wavelet function parameters $c_j \leq c_j \leq \bar{c}_j$, $\sigma_j \leq \sigma_j \leq \bar{\sigma}_j$, $\alpha_j \leq \alpha_j \leq \bar{\alpha}_j$ and introduce the type-2 fuzzy wavelet membership functions. These functions form the type-2 fuzzy wavelet neuron (T2FWN) and are shown in Figure 5.

---

**Figure 4. Adaptive wavelet function**

**Figure 5. Type-2 fuzzy-wavelet membership function with different type uncertainties**
T2FWN consists of two AWNs, where the signal $\tilde{x}(k)$ is fed to the inputs of it. Thus, one AWN uses low boundary values $c_\beta, \sigma_\beta, \alpha_\beta$, while the other AWN uses high boundary values $\bar{c}_\beta, \bar{\sigma}_\beta, \bar{\alpha}_\beta$. It is important to notice that neurons $\underline{\text{AWN}}$, $\overline{\text{AWN}}$ are trained independently of each other based on common reference signal $x(k)$.

As a result, the transformation that is implemented by $\underline{\text{AWN}}$ can be written in the form

$$y^w_w(\tilde{x}(k)) = \sum_{j=1}^{n} \sum_{l=1}^{p} w_{jl}(k-1) \tilde{\phi}_j(\tilde{x}_l(k))$$

and $\overline{\text{AWN}}$ -

$$\overline{y}^w_w(\tilde{x}(k)) = \sum_{j=1}^{n} \sum_{l=1}^{p} \overline{w}_{jl}(k-1) \overline{\phi}_j(\tilde{x}_l(k))$$

where

$$\tilde{\phi}_j(\tilde{x}_l(k)) = \left(1 - \alpha_j \frac{(\tilde{x}_l(k) - c_j)^2}{\sigma_j^2}\right) \exp\left(-\frac{(\tilde{x}_l(k) - c_j)^2}{2\sigma_j^2}\right),$$

$$\overline{\phi}_j(\tilde{x}_l(k)) = \left(1 - \alpha_j \frac{(\tilde{x}_l(k) - \bar{c}_j)^2}{\bar{\sigma}_j^2}\right) \exp\left(-\frac{(\tilde{x}_l(k) - \bar{c}_j)^2}{2\bar{\sigma}_j^2}\right).$$

In the type-reduction block, the signals $y^w_w(\tilde{x}(k))$ and $\overline{y}^w_w(\tilde{x}(k))$ are united in the simplest way and form the output signal of T2FWN

$$f(\tilde{x}(k)) = c(k)\overline{y}^w_w(\tilde{x}(k)) + (1 - c(k))y^w_w(\tilde{x}(k))$$

where $c(k)$ is the tuning parameter that defines closeness of signals $y^w_w(\tilde{x}(k))$ and $\overline{y}^w_w(\tilde{x}(k))$ to reference signal $x(k)$.

Finally, the forth (output) layer of system that consists of elementary sum and division blocks implements the defuzzification in the form

$$\hat{x}^{(4)}(k) = \frac{f(\tilde{x}(k))}{\sum_{i=1}^{n} \tilde{x}_i(k)} = c(k)\frac{\overline{y}^w_w(\tilde{x}(k))}{\sum_{i=1}^{n} \tilde{x}_i(k)} + (1 - c(k))\frac{y^w_w(\tilde{x}(k))}{\sum_{i=1}^{n} \tilde{x}_i(k)}.$$
where

\[ w(k-1) = (w_{11}(k-1), w_{12}(k-1), \ldots, w_{p1}(k-1), w_{12}(k-1), \ldots, w_{pp}(k-1))^T, \]

\[ \tilde{\psi}(\tilde{x}(k)) = (\tilde{\psi}_{11}(\tilde{x}(k)), \tilde{\psi}_{12}(\tilde{x}(k)), \ldots, \tilde{\psi}_{p1}(\tilde{x}(k)), \tilde{\psi}_{12}(\tilde{x}(k)), \ldots, \tilde{\psi}_{pp}(\tilde{x}(k)))^T, \]

\[ \tilde{w}(k-1) = (\tilde{w}_{11}(k-1), \tilde{w}_{12}(k-1), \ldots, \tilde{w}_{p1}(k-1), \tilde{w}_{12}(k-1), \ldots, \tilde{w}_{pp}(k-1))^T, \]

\[ \tilde{\tilde{\psi}}(\tilde{x}(k)) = (\tilde{\psi}_{11}(\tilde{x}(k)), \tilde{\psi}_{12}(\tilde{x}(k)), \ldots, \tilde{\psi}_{p1}(\tilde{x}(k)), \tilde{\psi}_{12}(\tilde{x}(k)), \ldots, \tilde{\psi}_{pp}(\tilde{x}(k)))^T. \]

### Adaptive Learning of HGAT2FWNN

The learning process of the layer \( L_i \) is the tuning of synaptic weight vectors \( w(k) \) and \( \tilde{w}(k) \) of the neurons \( \text{AWN}_1, \text{AWN}_2 \) and scalar parameter \( c(k) \) in the type-reduction block.

Since the output signals of wavelet neurons depend linearly on the synaptic weights, for their settings it is possible to use the exponentially weighted recursive least squares method, which is, in fact, the second-order optimisation procedure

\[
\begin{align*}
\{ w(k) &= w(k-1) + P(k-1) \frac{(x(k) - w^T(k) \tilde{\psi}(\tilde{x}(k)))}{\beta + \tilde{\psi}^T(\tilde{x}(k)) P(k-1) \tilde{\psi}(\tilde{x}(k))} \tilde{\psi}(\tilde{x}(k)), \\
\beta &= \frac{1}{P(k-1)} \left( P(k-1) \frac{\tilde{\psi}(\tilde{x}(k)) P(k-1) \tilde{\psi}(\tilde{x}(k))}{\beta + \tilde{\psi}^T(\tilde{x}(k)) P(k-1) \tilde{\psi}(\tilde{x}(k))} \right), \\
\tilde{w}(k) &= \tilde{w}(k-1) + \tilde{P}(k-1) \frac{(x(k) - \tilde{w}^T(k) \tilde{\psi}(\tilde{x}(k)))}{\beta + \tilde{\psi}^T(\tilde{x}(k)) \tilde{P}(k-1) \tilde{\psi}(\tilde{x}(k))} \tilde{\psi}(\tilde{x}(k)), \\
\tilde{\beta} &= \frac{1}{\tilde{P}(k-1)} \left( \tilde{P}(k-1) \frac{\tilde{\psi}(\tilde{x}(k)) \tilde{P}(k-1) \tilde{\psi}(\tilde{x}(k))}{\beta + \tilde{\psi}^T(\tilde{x}(k)) \tilde{P}(k-1) \tilde{\psi}(\tilde{x}(k))} \right)
\end{align*}
\]

(15)

where \( 0 < \beta \leq 1 \) – the forgetting factor.
To calculate the parameter $c(k)$ we can introduce an optimal adaptive procedure.

For off-line learning, we can write the learning error in the form

$$ e(k) = y(k) - \hat{y}(k) = y(k) - c\bar{y}^w(k) - (1 - c)\bar{y}^w(k) = y(k) - c\bar{y}^w(k) - \bar{y}^w(k) + c\bar{y}^w(k) $$

and the global learning criterion for the type-reduction block in the form

$$ E(k) = \sum_k (x(k) - \hat{y}^{[1]}(k))^2 = \sum_k (x(k) - c\bar{y}^w - (1 - c)\bar{y}^w(k))^2 = $$

$$ = \sum_k (x(k) - \bar{y}^w(k))^2 + 2\sum_k (c(x(k) - \bar{y}^w(k)))(\bar{y}^w(k) - \bar{y}^w(k))) + \sum_k c^2 (\bar{y}^w(k) - \bar{y}^w(k))^2. $$

Using criterion (1), we can provide optimization in the form

$$ \frac{\partial E(k)}{\partial c} = 2\sum_k ((x(k) - \bar{y}^w(k))(\bar{y}^w(k) - \bar{y}^w(k))) + 2\sum_k c(\bar{y}^w(k) - \bar{y}^w(k))^2 = 0 $$

and the expression for calculating parameter $c$ can be written as follows

$$ c = -\frac{\sum_k ((x(k) - \bar{y}^w(k))(\bar{y}^w(k) - \bar{y}^w(k)))}{\sum_k (\bar{y}^w(k) - \bar{y}^w(k))^2}. $$

Using the global learning criterion (17), we can obtain the optimal adaptive procedure for tuning parameter $c$.

Based on (19), we can write the value of parameter $c$ for $k$ and $k+1$ instant time

$$ c(k) = \frac{\sum_{i=1}^k ((x(i) - \bar{y}^w(i))(\bar{y}^w(i) - \bar{y}^w(i)))}{\sum_{i=1}^k (\bar{y}^w(i) - \bar{y}^w(i))^2} $$

or in the recurrent form

$$ \begin{cases} 
\gamma(k) = \gamma(k-1) + (\bar{y}^w(k) - \bar{y}^w(k))^2, \\
c(k) = c(k-1) + \frac{c(k-1)}{\gamma(k)} \left( x(k) - \bar{y}^w(k) \right) \bar{y}^w(k) \left( \bar{y}^w(k) - \bar{y}^w(k) \right) / \gamma(k) 
\end{cases} $$

It is clear that the condition of optimal learning is the closeness of signals $\bar{y}^w(k)$ and $\bar{y}^w(k)$, “subtending” of type-2 fuzzy wavelet membership function to common “Mexican Hat” wavelet and approximating of parameter $c(k)$ to the value 0.5.

The learning process of layers $L_2, L_3, \ldots, L_g$ is the same. At that after the next observation $x(k)$ is fed, all layers tune their parameters sequentially.
For making a one-step forecast, the signal in the form 
\[ X(k+1) = (x(k), x(k-1), \ldots, x(k-n+1))^{\top} \]
is fed to the output tuned system, after that the feature observation 
\[ \hat{x}^{[n]}(k+1) \] is appeared.

**Conclusions**

Proposed deep stacking forecasting hybrid system of computational intelligence is enough simple in computational implementation due to parallelizing process of implemented computations, has high learning rate due to dismissal from errors backpropagation and using the rapid adaptive algorithms for tuning its parameters, has also flexibility due to possibility of tuning the activation-membership functions. Proposed system is aimed at solving the wide range Data Stream Mining tasks, such as forecasting, identification, emulation, classification and pattern recognition in online mode. Experiments have confirmed the effectiveness of proposed system under consideration.

**Acknowledgement**

The paper is published with partial support by the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua)

**Bibliography**


2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), 2016, pp. 257-262. DOI 10.1109/DSMP.2016.7583555


Authors' Information

**Yevgeniy Bodyanskiy** – Dr. Sc., Prof., Control System Research Laboratory, Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine; e-mail: yevgeniy.bodyanskiy@nure.ua

Major Fields of Scientific Research: Computational Intelligence, Hybrid Evolving Systems, Data Stream Mining, Clustering, Forecasting.

**Iryna Pliss** – Ph.D., Leading researcher, Control System Research Laboratory, Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine; e-mail: iryna.pliss@nure.ua

Major Fields of Scientific Research: Data Mining, Computational Intelligence, Hybrid Evolving Systems, Clustering, Forecasting.

**Dmytro Peleshko** – Dr. Sc., Prof., University “IT Step Academy”, 83a Zamarstynivs’ka st., 79019, Lviv, Ukraine; e-mail: dpeleshko@gmail.com

Major Fields of Scientific Research: Computational Intelligence, Pattern Recognition, Computer Vision, Image and Video Processing.

**Olena Vynokurova** – Dr. Sc., Prof., Control System Research Laboratory, Kharkiv National University of Radio Electronics, 14 Nauky Ave., Kharkiv, 61166, Ukraine; e-mail: vynokurova@gmail.com

Major Fields of Scientific Research: Computational Intelligence, Hybrid Wavelet Neural Systems, Data Stream Mining, Clustering, Forecasting.