Abstract – In this paper, a simple wavelet-neuro-system that implements learning ideas based on minimization of empirical risk and oriented on information processing in on-line mode is developed. As an elementary block of such systems, we propose using wavelet-neuron that has improved approximation properties, computational simplicity, high learning rate and capability of local feature identification in data processing. The architecture and learning algorithm for least squares wavelet support machines that are characterized by high speed of operation and possibility of learning under conditions of short training set are proposed.

Keywords – Adaptive wavelet function, forecasting, least squares support vector machine, non-linear non-stationary time series, wavelet-neuron.

I. INTRODUCTION

Nowadays computational intelligence systems are widely used to solve a large class of problems associated with information processing, which is provided in different forms, from simple tables “object-property” to complex multidimensional non-stationary stochastic time series. Systems of computational intelligence, such as artificial neural networks, neuro-fuzzy systems and wavelet systems have a high processing speed, universal approximation properties, identification of local features [1]–[7].

However, the existing approaches for their training require large volumes of training data set, wherein the volume of the original training data set should be at least two orders bigger than the number of estimated parameters of such systems. Unfortunately, this situation does not always occur, especially in solving practical tasks. In many practical problems, for example, medical diagnostics, forecasting of financial indices etc., the volume of training data set is quite insufficient for constructing and training the effective system of computational intelligence.

The solving of problems in such a situation using conventional identification theory methods [8] is not effective, for which reason the method based on empirical risk minimization was proposed by V. N. Vapnik [9]–[12] and a support vector machine was designed based on this method.

Traditional support vector machine is a computational system that minimizes the empirical risk, but from viewpoint of practical implementation it is a sufficiently complex system, because it is related to the solving of nonlinear programming task at each step apart from high dimension with constraints in form of inequalities. Thus, in such a case the idea was wonderful; however, its implementation in on-line mode was not successful [2].

Therefore, at the beginning of this century, modification of this system has been proposed, called LS-SVM (least squares support vector machine) [13]. Here principal conditions were changed in such a way that it was necessary to solve a quadratic programming problem with equality constraints at each step, and it had already created the preconditions for the neural network implementation of this approach. But still, this machine was quite complicated and therefore further attempts were made to improve, firstly, speed, to simplify computing implementation and, secondly, to reduce training sample volume.

Based on fuzzy SVM [14], [15], the wavelet least squares support vector machine was proposed and compared with all previous variants – it had improved approximation properties, but, as a result, numerical implementation had become more difficult [16], [17]. The basis of LS-SVM was radial basis function network [18], where radial basis functions were replaced by multidimensional wavelet functions. For simplification of software implementation, by reducing the number of tuning parameters and increasing the speed we have proposed to use wavelet-neuron as the basic architecture [19]–[21] that has improved approximation properties and high operation speed. However, the disadvantages of wavelet-neuron are connected with its learning algorithms that do not allow tuning all parameters under a small data set.

Alternative to learning based on optimization is the learning based on memory that is based on the concept “neurons in the data points” [18]. The most typical representative of neural networks with such learning is General Regression Neural Network (GRNN); however, it solves a task of interpolation instead of approximation that essentially complicates its usage in noisy data processing.

Therefore, the development of sufficiently simple wavelet-neuro-fuzzy systems is advisable. Such systems implement learning based on empirical risk minimization and are oriented to information processing in on-line mode. Wavelet-neuron that has improved approximation and extrapolation properties can be used as a basic element of such systems.

II. WAVELET-NEURON ARCHITECTURE

Let us consider the wavelet-neuron architecture, shown in Fig. 1. As seen, wavelet-neuron is quiet close to the standard n-input formal neuron, but instead of tuning synaptic weights it contains wavelet-synapses $WS_{i}, i = 1, 2, \ldots, n$, where the tuning parameters are not only synaptic weights, but all parameters of adaptive wavelet activation functions $\phi_{ji}(x_{i}(k))$ [21].
When input vector
\[ x(k) = (x_1(k), x_2(k), \ldots, x_n(k))^T \]
is fed to the wavelet-neuron input (here \( k = 1, 2, \ldots \) is current discrete time) its output can be written in the form

\[
\hat{y}(k) = \sum_{i=1}^{n} f_i(x(k)) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ji}(k) \varphi_{ji}(x_i(k))
\]

where \( w_{ji}(k) \) is synaptic weight, \( \varphi_{ji}(x_i(k)) \) is wavelet activation function.

The different analytical wavelet functions can be used as activation functions, but for adaptive tuning of wavelet neuron we use an adaptive wavelet activation function which was proposed in [22], [23] and has the form

\[
\varphi_{ji}(x_i(k)) = \frac{1 - \alpha_{ji}(k) r_{ji}^2(k)}{2} \exp\left(-r_{ji}^2(k)/2\right)
\]

where \( r_{ji}(k) = {x_i(k) - c_{ji}(k)} / \sigma_{ji}(k) \); \( c_{ji}(k) \) is a center parameter of adaptive wavelet function; \( \sigma_{ji}(k) \) is a width parameter of adaptive wavelet function; \( \alpha_{ji}(k) \) is a shape parameter of adaptive wavelet function.

Tuning parameter \( \alpha_{ji} \) allows changing the shape of an adaptive wavelet activation function during the training process of network, and, as a result, for \( \alpha_{ji} = 0 \) we obtain Gaussian function, and when \( \alpha_{ji} = 1 \) we obtain wavelet function “Mexican Hat”, and when \( 0 < \alpha_{ji} < 1 \) – hybrid activation function.

Figure 2 shows the adaptive wavelet activation function with different parameters \( \alpha \) and \( \sigma \).

![Adaptive wavelet activation function](image)

The learning task is to find synaptic weights \( w_{ji}(k) \), centers \( c_{ji}(k) \), widths \( \sigma_{ji}(k) \) and shape parameters \( \alpha_{ji}(k) \) of adaptive wavelet activation function on each \( k \)-th iteration, which optimizes the learning criterion.

III. LEARNING ALGORITHM FOR ALL WAVELET-NEURON PARAMETERS

When the training data set is sufficiently large, as learning criteria we can use the conventional squared error function in the form

\[
E(k) = \frac{1}{2} \left( y(k) - \hat{y}(k) \right)^2 = \frac{1}{2} \left( y(k) - \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ji}(k) \varphi_{ji}(x_i(k)) \right)^2
\]

where \( y(k) \) is a reference signal.

Introducing some denominations in the form
\[
\varphi_{ji}(x_i(k)) = (\varphi_{j1}(x_1(k)), \ldots, \varphi_{jn}(x_n(k)))^T, \quad w_{ji}(k) = (w_{ji1}(k), \ldots, w_{jin}(k))^T, \quad c_{ji}(k) = (c_{j1}(k), \ldots, c_{jn}(k))^T, \quad \sigma_{ji}(k) = (\sigma_{j1}(k), \ldots, \sigma_{jn}(k))^T, \quad \alpha_{ji}(k) = (\alpha_{j1}(k), \ldots, \alpha_{jn}(k))^T, \quad t_{ji}(k) = (t_{ji1}(k), \ldots, t_{jin}(k))^T
\]

we can write the learning algorithm in the form

\[
\begin{align*}
w_{ji}(k + 1) &= w_{ji}(k) + \varepsilon(k) J_{ni}(k) \big/ \sigma^{2i}(k), \\
c_{ji}(k + 1) &= c_{ji}(k) + \varepsilon(k) J_{ni}(k) \big/ \sigma^{2i}(k), \\
\sigma_{ji}^{-1}(k + 1) &= \sigma_{ji}^{-1}(k) + \varepsilon(k) J_{ni}(k) \big/ \sigma^{2i}(k), \\
\alpha_{ji}(k + 1) &= \alpha_{ji}(k) + \varepsilon(k) J_{ni}(k) \big/ \sigma^{2i}(k).
\end{align*}
\]
The learning of wavelet-neuron using a least squares learning algorithm does not complicate numerical realization. For taking into account constraints, let us introduce the Lagrange function

\[ L(w,e(k),\lambda(k)) = E(k) + \sum_{k=1}^{N} \lambda(k) (y(k) - w^{T} \phi(x(k)) - e(k)) \]

(12)

where \( \lambda(k) \) – \( N \) indefinite Lagrange multipliers and system of Karush-Kuhn-Tucker equations

\[ \begin{align*}
\n\nabla_{w} L(w,e(k),\lambda(k)) &= w - \sum_{k=1}^{N} \lambda(k) \phi(x(k)) = \tilde{0}_{N}, \\
\n\frac{\partial L(w,e(k),\lambda(k))}{\partial e(k)} &= y(k) - \lambda(k) = 0, \\
\n\frac{\partial L(w,e(k),\lambda(k))}{\partial \lambda(k)} &= y(k) - w^{T} \phi(x(k)) - e(k) = 0 \\
\end{align*} \]

where \( \tilde{0}_{N} - (N \times 1) \) is a vector that consists of zero elements. Solution of system (13) can be written in the form

\[ \begin{align*}
\n\n w(N) &= \sum_{k=1}^{N} \lambda(k) \phi(x(k)), \\
\n\lambda(k) &= ye(k), \\
\n y(k) &= w^{T} (N) \phi(x(k)) + e(k) \\
\end{align*} \]

(14)

or in the matrix form

\[ \left( y^{-1} I_{NN} + \Omega_{NN} \right) \begin{pmatrix}
\lambda(1) \\
\vdots \\
\lambda(N)
\end{pmatrix} = 
\begin{pmatrix}
y(1) \\
\vdots \\
y(N)
\end{pmatrix} \]

(15)

(here \( I_{NN} - (N \times N) \) is an identity matrix) or
were taken as prehistory for.

\[(\gamma^{-1}I_{NN} + \Omega_{NN})A_N = Y_N\]  

(16)

\[\text{(here } \Omega_{NN} = \{\Omega_{pq} = \varphi^T(x(p))\varphi(x(q))\}, p=1,2,\ldots,N; q=1,2,\ldots,N \text{ from it follows)}\]

\[A_N = (\gamma^{-1}I_{NN} + \Omega_{NN})^{-1}Y_N.\]  

(17)

Then an output signal of wavelet-neuron can be written in the form

\[\hat{y}(x) = \left(\sum_{k=1}^{N} \lambda(k)\varphi(x(k))\right)^T \varphi(x).\]  

(18)

As can be seen from neuromathematical point of view (the neural network learning theory) and theory of support vector machines (empirical risk minimization), the proposed wavelet least squares support vector machine based on wavelet-neuron is more simple in the implementation, has high speed of operation and requires short volume of a training data set.

If the data are fed sequentially, the process of wavelet least squares support vector machine learning should be organized in on-line mode. When a pair \(x(N+1), y(N+1)\) is fed to the input of wavelet-neuron, expression (18) can be written in the form

\[\hat{y}(x) = \left(\sum_{k=1}^{N} \lambda(k)\varphi(x(k)) + \lambda(N+1)\varphi(x(N+1))\right)^T \varphi(x).\]  

(19)

or in the matrix form

\[(\gamma^{-1}I_{N+1,N+1} + \Omega_{N+1,N+1})\lambda = \begin{pmatrix} Y(1) \\ \vdots \\ Y(N) \\ Y(N+1) \end{pmatrix}\]  

(20)

or

\[\begin{pmatrix} \Omega_{NN} & \omega_{N+1} \\ \omega_{N+1}^T & 1 \end{pmatrix} \begin{pmatrix} \lambda(N) \\ \lambda(N+1) \end{pmatrix} = \begin{pmatrix} Y_N \\ Y(N+1) \end{pmatrix}\]  

(21)

where

\[\omega_{N+1} = (\mu^T(x(1))\mu(x(N+1)), \mu^T(x(2))\mu(x(N+1)), \ldots, \mu^T(x(N))\mu(x(N+1)))^T.\]

Using (21) we can write

\[A_{N+1} = \begin{pmatrix} \Lambda_N \\ \lambda(N+1) \end{pmatrix} = \begin{pmatrix} \Omega_{NN} & \omega_{N+1} \\ \omega_{N+1}^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} Y_N \\ Y(N+1) \end{pmatrix}.\]  

(22)

Using the Frobenius formula for a block matrix in the form [25]

\[K = A - BD^{-1}C,\]

we can write

\[K = \Omega_{NN} - \omega_{N+1}\gamma\omega_{N+1}^T, \quad K^{-1} = (\Omega_{NN} - \gamma\omega_{N+1}\omega_{N+1}^T)^{-1}.\]

Further, it is easy to compute \((N+1)\)-th Lagrange multiplier using an expression in the form

\[\lambda(N+1) = -\gamma\omega_{N+1}^TK^{-1}Y_N + \gamma(1 + \gamma\omega_{N+1}^TK^{-1}\omega_{N+1})y(N+1).\]

(23)

Further, using the Sherman-Morrison formula for matrix inversion we can rewrite the learning algorithm in the final form

\[\begin{pmatrix} K^{-1} = \Omega_{NN}^{-1} + \Omega_{NN}^{-1}\omega_{N+1}\omega_{N+1}^T\Omega_{NN}^{-1} \\ 1 - \omega_{N+1}\Omega_{NN}^{-1}\omega_{N+1} \end{pmatrix}, \]

\[\lambda(N+1) = 1 + \gamma\omega_{N+1}^TK^{-1}\omega_{N+1} - Y_N.\]  

(24)

V. RESULT OF SIMULATION

To demonstrate the effectiveness of the proposed wavelet least squares support vector machine, the practical problem of time series forecasting, which describes the average monthly temperature in Kharkiv, Ukraine, was solved [26]. Time series consisted of 24 points and, thus, a training sample contained 16 points and 8 points were taken as a testing sample.

Values \(x(k-2), x(k-1), x(k)\) were taken as prehistory for the forecasting \(x(k+1)\) value. Initial value of shape parameter of the adaptive wavelet activation function was taken as \(\alpha = 1\).

As forecasting quality criterion we used a mean squared error (MSE).

Fig. 3 shows the results of time series forecasting based on wavelet-neuron with different learning algorithms. As can be seen in Fig. 3a, the curves of the actual values (a dashed curve) and forecast ones (a solid curve) are close enough. Fig. 3b shows the results of forecasting using the wavelet-neuron and gradient learning method with a constant step, and Fig. 3c shows the results of wavelet-neuron and the proposed learning algorithm (4) and (5).
Table I provides the comparison of results of time series forecasting using the wavelet-neuron with different approaches to learning.

**TABLE I**

<table>
<thead>
<tr>
<th>Neural network / Learning algorithm</th>
<th>Num. of input / Num. of activation function</th>
<th>MSEtrn</th>
<th>MSEchk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wavelet-neuron / Proposed learning algorithm based on SVM-criterion</td>
<td>3/8 (on-line)</td>
<td>0.0063</td>
<td>0.0306</td>
</tr>
<tr>
<td>Wavelet-neuron / Gradient learning algorithm with a constant step</td>
<td>3/8 (10 epoch)</td>
<td>0.000093</td>
<td>0.3186</td>
</tr>
<tr>
<td>Wavelet-neuron / Proposed learning algorithm of all-parameters (4), (5)</td>
<td>3/8 (10 epoch)</td>
<td>0.0374</td>
<td>0.0572</td>
</tr>
</tbody>
</table>

As it can be seen, the wavelet neuron with a gradient learning algorithm shows the best result on a training set, but such a system has the worst prediction abilities. The wavelet-neuron with learning algorithm (4), (5) is not able to train all parameters of the system because of a small data set. Thus, as can be seen from experimental results, the proposed approach provides the best quality of forecasting in comparison with similar approaches due to a special learning algorithm that is able to process information in both off-line and on-line modes.

**VI. CONCLUSION**

The wavelet least squares support vector machine based on wavelet-neuron was introduced and investigated. The proposed wavelet least squares support machine has such advantages as computational simplicity due to the wavelet-neuron architecture, small number of tuning parameters, high speed operation thanks to the use of the second order learning algorithms and the possibility of on-line information processing.

**REFERENCES**


Вейгеня Бодяйский.

Является соучредителем Саратовского научно-исследовательского института сверхнизких частот и автором более 100 научных работ, включая 40 изобретений и 12 монографий. Некоторые ключевые работы включают:


Специалисты и эксперты отмечают, что вейгеняйский подход к решению задачи обеспечивает высокую точность и надежность результатов. Вейгеняйский также активно участвует в международных научных конференциях и семинарах, где он представляет свои исследования и доклады перед ведущими специалистами в области машинного обучения и интеллектуальных систем.