An Adaptive Learning Algorithm for a Neo Fuzzy Neuron

Ye. Bodyanskiy **I. Kokshenev** **V. Kolodyazhniy**

Control Systems Research Laboratory,
Kharkiv National University of Radioelectronics,
14, Lenin Av., Kharkiv, 61166, Ukraine.
{bodyanskiy, ikokshenev, kolodyazhniy}@ieee.org

Abstract

In the paper, a new optimal learning algorithm for a neo-fuzzy neuron (NFN) is proposed. The algorithm is characteristic in that it provides online tuning of not only the synaptic weights, but also the membership functions parameters. The proposed algorithm has both the tracking and filtering properties, so the NFN can be effectively used for prediction, filtering, and restoration of non-stationary noisy stochastic and chaotic signals. A special feature of the proposed algorithm is its computational simplicity in comparison with the other learning procedures for neuro-fuzzy systems.

1 Introduction

Artificial neural networks (ANN) and fuzzy inference systems (FIS) have been widely used in recent years to solve a wide range of problems such as data mining and processing of signals of different nature under *a priori* and current uncertainty. Hybrid neuro-fuzzy systems emerged as a synergism of these two major directions in computational intelligence. The neuro-fuzzy systems possess the learning capabilities similar to those of neural networks, and provide the interpretability and "transparency" of results, inherent to the fuzzy approach. The disadvantages of the neuro-fuzzy systems are related to the slow convergence of the gradient-based learning procedures. The use of the second-order optimization methods is limited by their computational complexity and "the curse of dimensionality", which arises when the number of inputs is large. Certain problems may arise in the processing of non-stationary signals, since the second-order procedures with exponential forgetting can be numerically instable. To overcome these difficulties, a neuro-

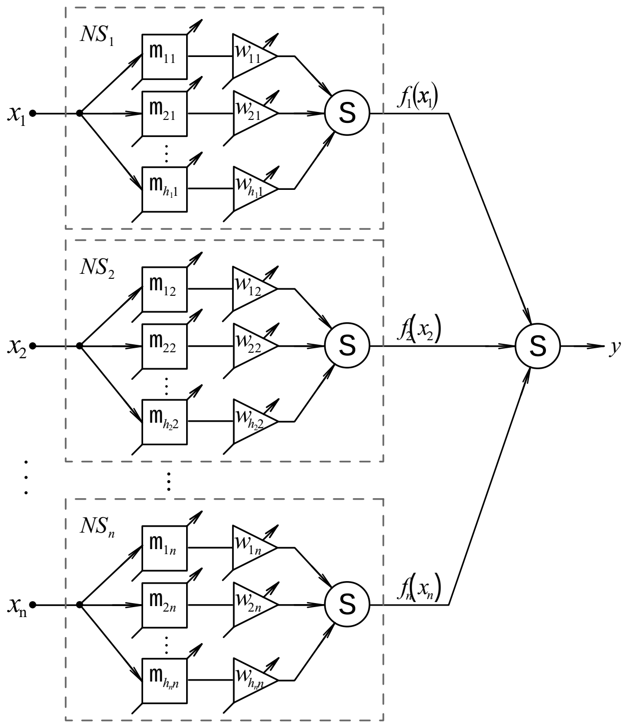

Figure 1: Neo-fuzzy neuron

fuzzy structure called "neo-fuzzy neuron" (Fig. 1) was proposed in [13, 11, 7]. The architecture of NFN is quite close to the conventional n-input artificial neuron, however, instead of usual synaptic weights it contains nonlinear synapses NS_i, $i = 1, 2, \ldots, n$.

When a vector signal $x(k) = (x_1(k), x_2(k), \ldots, x_n(k))^T$ (where $k = 1, 2, \ldots$ is discrete time) is fed to the input of the neo-fuzzy neuron, the output of that neuron is determined by both the membership functions μ_{ji} and the tunable weights $w_{ji}(k)$:

$$y(k)=\sum_{i=1}^{n}f_i(x_i(k))=\sum_{i=1}^{n}\sum_{j=1}^{h_i}\mu_{ji}(x_i(k))w_{ji}(k). \quad (1)$$

On the one hand, the NFN is similar to a 0-th order Sugeno fuzzy system [10], in which only one input is included in each fuzzy rule, and to a radial basis function network (RBFN) [4] with scalar arguments of basis functions, on the other hand. Taking into account the functional equivalence between FIS and RBFN [5], as well as their universal approximation properties [8, 12], we can hope that the neo-fuzzy neuron could be successfully applied to various problems. The authors of the NFN note [11], among its most important advantages, the high rate of learning, computational simplicity, the possibility of finding the global minimum of the learning criterion in real time.

The learning criterion is the standard quadratic error function

$$E(k)=\frac{1}{2}(d(k)-y(k))^2=\frac{1}{2}e^2(k) \quad (2)$$

minimized via the conventional gradient descent procedure, resulting in the following weight update algorithm

$$w_{ji}(k+1)=w_{ji}(k)+\eta e(k)\mu_{ji}(x_i(k)), \quad (3)$$

where $d(k)$ is the target value of the output, η is the scalar learning rate which determines the rate of convergence and is chosen empirically.

Since the membership functions are triangular (as shown in Fig. 2) and the sum of activation levels of the membership functions in each NFN is unity $\sum_{j=1}^{h_i}\mu_{ji}(x_i(k))=1$, $i=1,2,\dots,n$, the standard NFN does not require a special normalization layer. Our goal is to improve the learning speed and the approximation properties of the standard NFN via the optimal choice of the learning rate and tuning the membership functions themselves in the learning process.

2 Optimal learning algorithm

Let us introduce $(h_i\times 1)$ vectors $\mu_i(x_i(k)) = (\mu_{1i}(x_i(k)),\mu_{2i}(x_i(k)),\dots,\mu_{h_i i}(x_i(k)))^T$, $w_i(k) = (w_{1i}(k),w_{2i}(k),\dots,w_{h_i i}(k))^T$, $E_i=(1,1,\dots,1)^T$ and re-write the expression (1) as

$$y(k)=\sum_{i=1}^{n}w_i^T(k)\mu_i(x_i(k)), \quad (4)$$

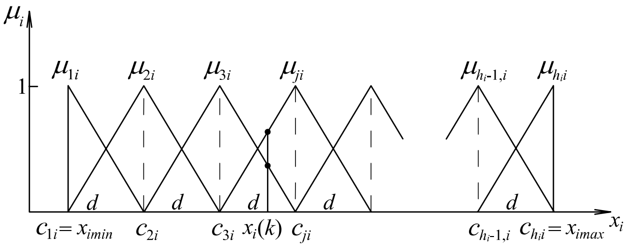

Figure 2: Triangular membership functions of a standart neo-fuzzy neuron

which holds only if $E_i^T\mu_i(x_i(k))=1$. If this is not the case, the output signal must be defuzzified as

$$y(k)=\sum_{i=1}^{n}\frac{w_i^T(k)\mu_i(x_i(k))}{E_i^T\mu_i(x_i(k))}. \quad (5)$$

By the substitution of variables

$$\widetilde{\mu}_i(x_i(k))=\frac{\mu_i(x_i(k))}{E_i^T\mu_i(x_i(k))}, \quad (6)$$

we obtain the following expression

$$y(k)=\sum_{i=1}^{n}w_i^T(k)\widetilde{\mu}_i(x_i(k)), \quad (7)$$

similar to (4).

Minimization of the error function (2) via the gradient descent-based procedure yield the following learning algorithm for the i-th nonlinear synapse NS_i:

$$\begin{aligned} w_i(k+1) &= w_i(k)-\eta(k)\nabla_{w_i}E(k)= \\ &= w_i(k)+\eta(k)e(k)\widetilde{\mu}_i(x_i(k)). \end{aligned} \quad (8)$$

Introducing the extended $(h\times 1)$ vectors $\widetilde{\mu}_i^T(x(k)) = \left(\widetilde{\mu}_1^T(x_1(k)),\widetilde{\mu}_2^T(x_2(k)),\dots,\widetilde{\mu}_n^T(x_n(k))\right)$, and $w^T(k) = \left(w_1^T(k),w_2^T(k),\dots,w_n^T(k)\right)$, we obtain the learning algorithm for all the weights of the neo-fuzzy neuron

$$w(k+1)=w(k)+\eta(k)e(k)\widetilde{\mu}(x(k)), \quad (9)$$

where $h=\sum_{i=1}^{n}h_i$, and $y(k)=w^T(k)\widetilde{\mu}(x(k))$.

Multiplying (9) by $\widetilde{\mu}^T(x(k))$ from the left-hand side and subtracting both parts of the resulting expression from $d(k)$, we obtain $\widetilde{e}(k+1)=e(k)(1-\eta(k))\|\widetilde{\mu}(x(k))\|^2$, where $\widetilde{e}(k+1)$ is the *a posteriori* learning error after one step of parameter update. Squaring both parts of (**??**) and solving the differential equation $\partial\widetilde{e}^2(k+1)/\partial\eta=0$, we obtain the optimal value of the learning rate $\eta(k) = \|\widetilde{\mu}(x(k))\|^{-2}$, and the corresponding learning algorithm

$$w(k+1) = w(k) + \frac{e(k)\widetilde{\mu}(x(k))}{\|\widetilde{\mu}(x(k))\|^2}, \qquad (10)$$

which is a variety of the well-known Widrow-Hoff procedure [4].

For smoothing properties, let us re-write (10) as the following exponentially weighted modification

$$\begin{cases} w(k+1) = w(k) + e(k)\widetilde{\mu}(x(k))\alpha^{w^{-1}}(k), \\ \alpha^w(k+1) = \alpha\alpha^w(k) + \|\widetilde{\mu}(k+1)\|^2, \\ 0 \le \alpha \le 1 \end{cases} \qquad (11)$$

(where α is the forgetting factor), similar to the Goodwin-Ramadge-Caines stochastic approximation procedure of [3].

3 Neo-fuzzy neuron with tunable membership functions

Let us introduce a neo-fuzzy neuron with quadratic membership functions, shown in Fig. 3. These func-

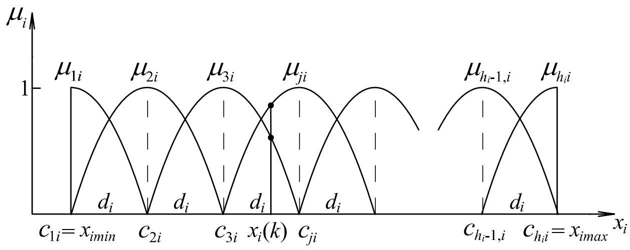

Figure 3: Quadratic membership functions

tions can be defined analytically as

$$\mu_{ji}(x_i) = (1 - (c_{ji} - x_i)^2 d_{ji}^{-2})\delta_{ji}. \qquad (12)$$

where

$$\delta_{ji} = \begin{cases} \text{if } |c_{ji} - x_i| < d_{ji}, 1 \\ \text{else } 0. \end{cases} \qquad (13)$$

After differentiating, we can derive a gradient descent-based learning procedure for the weights and membership functions as

$$\begin{cases} w_{ji}(k+1) = w_{ji}(k) + \eta^w(k)e(k)\mu_{ji}(k), \\ c_{ji}(k+1) = c_{ji}(k) - \eta^c(k)e(k)\delta_{ji}(k)\dfrac{c_{ji}(k) - x_i(k)}{d_{ji}^2(k)}, \\ d_{ji}^{-2}(k+1) = d_{ji}^{-2}(k) - \eta^c(k)e(k)\delta_{ji}(k)\cdot \\ \qquad \cdot (c_{ji}(k) - x_i(k))^2, \end{cases} \qquad (14)$$

where $\mu_{ji}(k) = (1 - (c_{ji}(k) - x_i(k))^2 d_{ji}(k)^{-2})\delta_{ji}(k)$,

$$\delta_{ji}(k) = \begin{cases} \text{if } |c_{ji}(k) - x_i(k)| < d_{ji}(k), 1 \\ \text{else } 0, \end{cases}$$

$\eta^w(k), \eta^c(k), \eta^d(k)$ are scalar learning rates for the respective variables. It should be noted that with the tunable membership functions the sum of memberships will not be unity, i.e. $\sum_{j=1}^{h_i} \mu_{ji}(x_i(k), k) \neq 1$, This does not affect the approximation properties of the NFN, which operates similarly to a radial basis function network. If the constraint on the sum of memberships should be observed, we can make the substitution according to (6).

Introducing $(h_i \times 1)$ vectors of variables of the i-th nonlinear synapse

$$\begin{aligned} \mu_i(k) &= (\mu_{1i}(k), \mu_{2i}(k), \ldots, \mu_{h_i i}(k))^T, \\ c_i(k) &= (c_{1i}(k), c_{2i}(k), \ldots, c_{h_i i}(k))^T, \\ d_i^{(-2)}(k) &= \left(d_{1i}^{-2}(k), d_{2i}^{-2}(k), \ldots, d_{h_i i}^{-2}(k)\right)^T, \\ \delta_i(k) &= (\delta_{1i}(k), \delta_{2i}(k), \ldots, \delta_{h_i i}(k))^T, \end{aligned} \qquad (15)$$

and $(h \times 1)$ vectors of all the variables of neo-fuzzy neuron

$$\begin{aligned} \mu^T(k) &= \left(\mu_1^T(k), \mu_2^T(k), \ldots, \mu_n^T(k)\right), \\ c^T(k) &= \left(c_1^T(k), c_2^T(k), \ldots, c_n^T(k)\right), \\ d^{(-2)^T}(k) &= (d_1^{(-2)^T}(k), d_2^{(-2)^T}(k), \ldots, d_n^{(-2)^T}(k)), \\ x^T(k) &= (x_1(k)E_1^T, x_2(k)E_2^T, \ldots, x_n(k)E_n^T), \\ \delta^T(k) &= \left(\delta_1^T(k), \delta_2^T(k), \ldots, \delta_n^T(k)\right), \end{aligned} \qquad (16)$$

we obtain the learning algorithm

$$\begin{cases} w(k+1) = w(k) + \eta^w(k)e(k)\mu(k), \\ c(k+1) = c(k) - \eta^c(k)e(k)\delta(k)w(k)\odot \\ \qquad \odot\, d^{(-2)}(k) \odot (c(k) - x(k)), \\ d^{(-2)}(k+1) = d^{(-2)}(k) - \eta^d(k)e(k)\delta(k)w(k)\odot \\ \qquad \odot\, (c(k) - x(k)) \odot (c(k) - x(k)), \end{cases} \qquad (17)$$

where $\odot$ is the symbol of element-wise multiplication (direct product).

The convergence speed can be increased via the use of second order learning procedures, such as the well-known Levenberg-Marquardt algorithm [9]. Using the following notation

$$\begin{cases} \mu^c(k) = 2\delta(k)w(k) \odot d^{(-2)}(k) \odot (c(k) - x(k)), \\ \mu^d(k) = \delta(k)w(k) \odot (c(k) - x(k)) \odot (c(k) - x(k)), \end{cases} \qquad (18)$$

we can derive the learning procedure for the neo-fuzzy neuron based on the Levenberg-Marquardt algorithm as

$$\begin{cases} w(k+1)=w(k)+e(k)\left(\mu(k)\mu^T(k)+\alpha^w I\right)^{-1}\mu(k), \\ c(k+1)=c(k)-e(k)\left(\mu^c(k)\mu^{cT}(k)+\alpha^c I\right)^{-1}\mu^c(k), \\ d^{(-2)}(k+1)=d^{(-2)}(k)-e(k)\cdot \\ \qquad \cdot\left(\mu^d(k)\mu^{dT}(k)+\alpha^d I\right)^{-1}\mu^d(k), \end{cases} \quad (19)$$

where α^w, α^c, α^d are positive regularizing additional terms, I is the unity matrix $(h\times h)$.

Using the Sherman-Morrison formula and making obvious transformations [1, 2], the algorithm (19) can be re-written in the following simple form

$$\begin{cases} w(k+1)=w(k)+\dfrac{e(k)\mu(k)}{\alpha^w+\|\mu(k)\|^2}=w(k)+\dfrac{e(k)\mu(k)}{\alpha^w(k)}, \\ c(k+1)=c(k)-\dfrac{e(k)\mu^c(k)}{\alpha^c+\|\mu^c(k)\|^2}=c(k)-\dfrac{e(k)\mu^c(k)}{\alpha^c(k)}, \\ d^{(-2)}(k+1)=d^{(-2)}(k)-\dfrac{e(k)\mu^d(k)}{\alpha^d+\|\mu^d(k)\|^2}= \\ \qquad =d^{(-2)}(k)-\dfrac{e(k)\mu^d(k)}{\alpha^d(k)}, \end{cases} \quad (20)$$

which is a nonlinear extension of the Widrow-Hoff algorithm.

For filtering properties, the scalar parameters $\alpha^w(k)$, $\alpha^c(k)$, $\alpha^d(k)$ in (20) can be recursively re-computed as follows:

$$\begin{cases} \alpha^w(k+1)=\alpha\alpha^w(k)+\|\mu(k+1)\|^2, \\ \alpha^c(k+1)=\alpha\alpha^c(k)+\|\mu^c(k+1)\|^2, \\ \alpha^d(k+1)=\alpha\alpha^d(k)+\|\mu^d(k+1)\|^2, \\ 0\le\alpha\le 1. \end{cases} \quad (21)$$

The use of quadratic membership functions and the 2-nd order learning algorithms does not complicate the implementation of the neo-fuzzy neuron significantly, but the convergence of the learning process of all the parameters of the neo-fuzzy neuron is improved due to the optimal choice of the learning rate.

4 Experimental results

The efficiency of the proposed algorithms was tested in the problem of real-time forecasting with an adaptive predictor, trained in real time with different procedures.

The experiment consists in one-step forecasting of the chaotic signal [6]

$$x(k+1)=4(1-x(k))x(k),\ k=0,1,2,\ldots \quad (22)$$

The initial value of the chaotic process (22) was chosen as $x(0)=0.01$. The training data set contained 1500 values for $k=100,\ldots,1499$, and the checking data set contained 500 values for $k=1500,\ldots,1999$. The neo-fuzzy predictor had 1 nonlinear synapse with 5 membership functions, with parameters $\eta=0.6$ in (3), $\alpha=0.5$ in optimal procedure (11) and $\alpha=0.6$ in (20)– (21).

The absolute error on first 1000 steps of training process in Fig. 4 shows that the optimal learning procedure (11) has the fastest rate of convergence in comparison with the other two algorithms. Although the application of the algorithm (20)– (21) resulted in bigger error in the initial stage of learning, this algorithm provided considerable improvement of the forecast quality.

The normalized root mean squared error (NRMSE) was used to estimate the forecast accuracy on the checking data set. The numerical results are shown in Table 1.

Table 1: Performance of a neo-fuzzy network in the forecasting of the chaotic process (22)

Trainng procedure	$NRMSE_{chk}$
Constant learning rate (3)	0.0787
Optimal (11)	0.0779
(20)– (21)	0.0234

5 Conclusions

Optimal learning algorithms and tunable membership for the neo-fuzzy neuron were proposed. The algorithm is simple in implementation and provides high quality of signal processing in real time. Tuning of membership functions improves the accuracy of modeling of nonlinear processes. This was demonstrated in the experiment of chaotic time series forecasting. More experimental results will be presented at the conference.

References

[1] Ye. V. Bodyanskiy and V. V. Kolodyazhniy. About one algorithm of neuro-fuzzy prediction

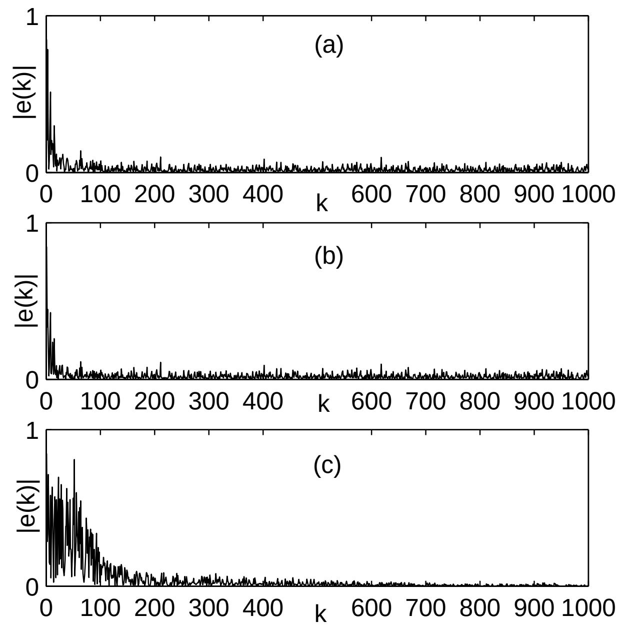

Figure 4: First 1000 steps of training on a chaotic process (22) using: (a) — algorithm with constant learning rate (3), (b) — optimal algorithm (11), (c) — algorithm (20)– (21)

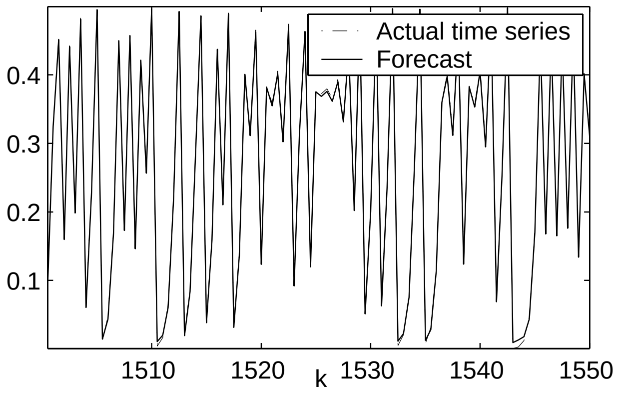

Figure 5: Forecast using (20)– (21) and actual time series

training. In *Adaptive systems of automatic control*, volume 23, pages 29–36, 2000 (in Russian).

[2] Ye. V. Bodyanskiy, V. V. Kolodyazhniy, and A. Stephan. An adaptive learning algorithm for a neuro-fuzzy network. In B. Reusch, editor, *Computational Intelligence. Theory and Applications*, pages 68–75, Berlin — Heidelberg — New York, 2001. Springer.

[3] G. C. Goodwin, P. J. Ramadge, and P. E. Caines. A globally convergent adaptive predictor. *Automatica*, 17(1):135–140, 1981.

[4] S. Haykin. *Neural Networks. A Comprehensive Foundation*. Prentice Hall, Inc., Upper Saddle River, N.J., 1999.

[5] J.-S. R. Jang and C.-T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Trans. on Neural Networks*, 4(1):156–159, 1993.

[6] B. B. Mandelbrot. *Die Fractale Geometrie der Natur*. Birkhaeuser Verlag, Basel, 1991.

[7] T. Miki and T. Yamakawa. Analog implementation of neo-fuzzy neuron and its on-board learning. In N.E. Mastorakis, editor, *Computational Intelligence and Applications*, pages 144–149, Piraeus, 1999. WSES Press.

[8] J. Park and I. W. Sandberg. Universal approximation using radial-basis-function network. In *Neural Computation*, volume 3, pages 246–257, 1991.

[9] A. J. Shepherd. *Second-Order Methods for Neural Networks*. Springer-Verlag, London, 1997.

[10] M. Sugeno. An introductory survey of fuzzy control. *Information Sciences*, 36:59–83, 1985.

[11] E. Uchino and T. Yamakawa. Soft computing bases signal prediction, restoration, and filtering. In Da Ruan, editor, *Intelligent Hybrid Systems: Fuzzy Logic, Neural Networks, and Genetic Algorithms*, pages 331–349, Boston, 1997. Kluwer Academic Publishers.

[12] L. X. Wang. Fuzzy systems are universal approximators. In *Proc. 1-st IEEE Conf. on Fuzzy Systems*, pages 1163–1169, San Diego, 1992.

[13] T. Yamakawa, E. Uchino, T. Miki, and H. Kusanagi. A neo fuzzy neuron and its applications to system identification and prediction of the system behavior. In *Proc. 2-dn Int. Conf. on Fuzzy Logic and Neural Networks — "IIZUKA—92"*, pages 477–483, Iizuka, Japan, 1992.