

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Neuro-Fuzzy Digital Filter

José de Jesús Medel¹, Juan Carlos García² and Juan Carlos Sánchez²

¹Computing Research Centre, México D. F

²Superior School of Mechanical and Electrical Engineering,
Department of Micro Electronics Research
Mexico

1. Introduction

1.1 Neural net

An artificial neural net is a computational model which imitates natural biological system actions, through neurons that adapt their gains as occurs in the brain, and these are interconnected constructing a neural net system (Nikola, 1996) (Medel, García y Sánchez, 2008), shown in figure 1.



Fig. 1. Neural Network Interconnections (Source: Benedict Campbell 2008).

The Biological neuron is described illustratively in figure 2, taking into account a biological description.

In traditional concepts a neuron operates receiving signals from other neurons through bioelectrical connections, called *synapses*. The combination of these signals, in excess of a certain *threshold* or *activation* level, will result in the neuron *firing* that is sending a signal on to other interconnected neurons. Some signals act as *excitations* and others as *inhibitions* to a neuron firing.

These acts applied in a hundred billion interconnected neurons generate "thinking actions".

Each neuron has a body, called the *soma*. The soma is much like the body of any other cell, containing the cell nucleus, various bio-chemical factors and other components that support

ongoing activity, and surround the soma *dendrites*. The dendrites have the receptor functions with respect to signals generated by other neurons. These signals combined may determine whether or not that neuron will fire.

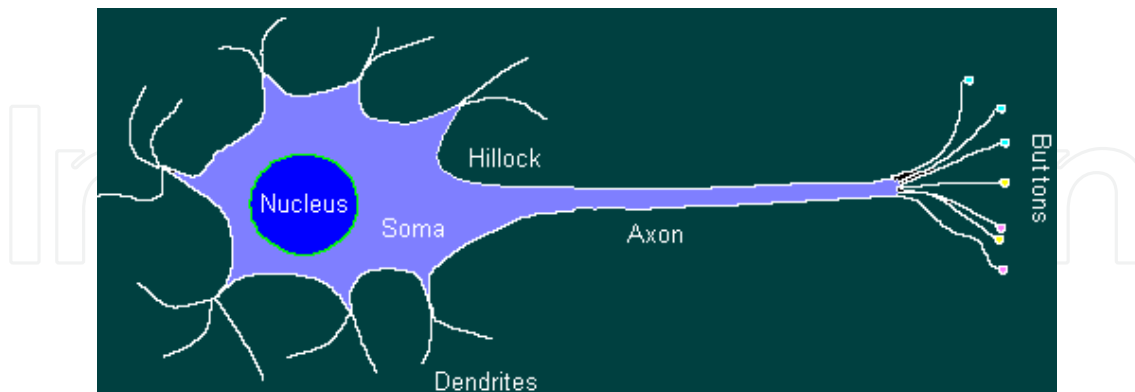


Fig. 2. Basic Biological Neuron with its elements.

If a neuron fires, an electrical impulse noise is generated. This impulse starts at the base, called the *hillock*, of a long cellular extension, called the *axon*, and proceeds down the axon to its ends. The end of the axon is split into multiple ends, called the *buttons*. The buttons are connected to the dendrites of other neurons and the resulting interconnections are the previously discussed synapses. (In figure 2, the buttons do not touch other dendrites having a small gap generating an electrical potential difference between them; i.e., if a neuron has fired, the electrical impulse noise that has been generated stimulates the buttons and results in electro-chemical activity which transmits the signal across the synapses dendrites actions).

Commonly, the neuron maintains an electrical interval potential $[35, 65]$ milli-volts; but when a neuron fires an electrical impulse noise it increases its chemical electric energy releasing an electrical potential $[90, 110]$ milli-volts. This impulse noise is transmitted with an interval velocity $[0.5, 100]$ in meters per second and is distributed on average in a 1 milli-second. The fast rate repetition on average corresponds to 10 milli-seconds per firing.

Considering an electronic computer whose signals travel on average at $2.0 \times 10^9 \frac{m}{sec}$ (speed of electrical energy in a wire is 0.7 of that in air), whose impulse noises last for ten nanoseconds and may repeat such an impulse noise in each succeeding 10 nano-seconds. Therefore, an electronic computer has at least a two thousand times advantage in signal transmission speed considering the biological basic neuron, and a thousand times advantage in signal fire repetition. This difference in velocity manifests itself in at least one important way; the human brain is not as fast as an arithmetic electronic computer, which is many times faster and hugely more capable of patterns recognition and perception relationships. The main advantage of the brain in respect to other electronic devices is it is capable of "self-programming" with changing external stimuli, known as "adaptability". In other words, it can learn dynamically and in all conditions.

Naturally, the brain has developed the neuron ways changing their response to new stimulus so that similar events may affect future neighborhood responses. The adaptability of a brain corresponds to survival actions.

1.2 Neural network structure

The computational Neural Network structures are based on biological neural configurations. The basic neural network is a model neuron, shown in figure 2, consisting of Multiple Inputs and a Single Output (MISO structure). Each input is modified by a *weight*, which multiplies the input value. The neuron combines these dendrite weight inputs and if the soma biological actions exceed a threshold value, then the nucleus in biological sense and activation function in computational actions, determines its output answer. In an electronic computational device as shown in Figure 3, a behavioral additional condition has the answer close to the real neuron actions.

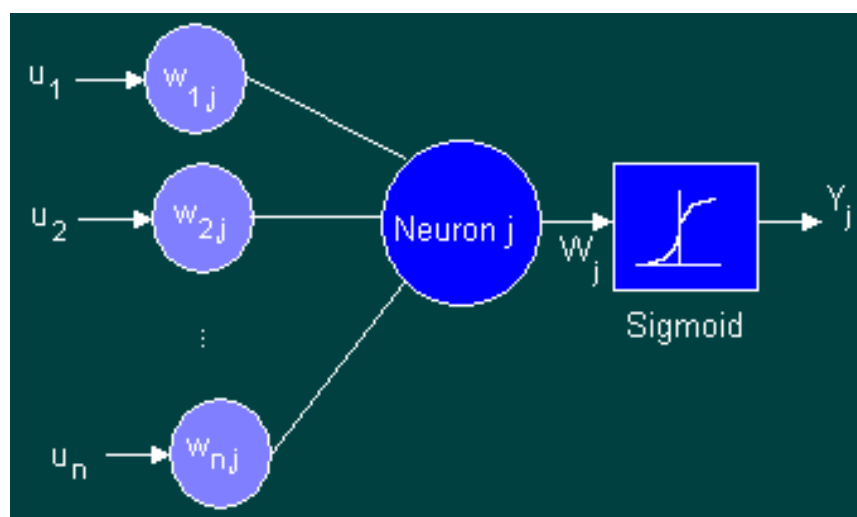


Fig. 3. Neuron device computational model

Meanwhile understanding how an individual neuron operates many researches generate the way neurons organize themselves and the mechanisms used by arrays of neurons to adapt their behavior to external bounded stimuli. There are a huge number of experimental neural network computational structures, and actually laboratories and researchers continue building new neural net configurations.

The common computational neural net used, is called *back-propagation network* and is characterized with a mathematical structure model, which knows its behavioral stability conditions (bounded inputs and bounded output, BIBO conditions).

Intuitively it is built taking a number of neurons and arrays them forming a *layer*. A layer is formed having all inputs and nodes interconnected with others nodes, but not both within the same node. A layer finishes with a node set connected with a succeeding layer or outputs giving the answer. The multiple layers are arrayed as an input layer, multiple intermediate layers and an output layer as shown in Figure 4; where the intermediate layers do not have inputs or outputs to the external world and are called *hidden layers*.

Back-propagation neural networks are usually *fully connected*. This means that each neuron is connected to every output from the preceding layer.

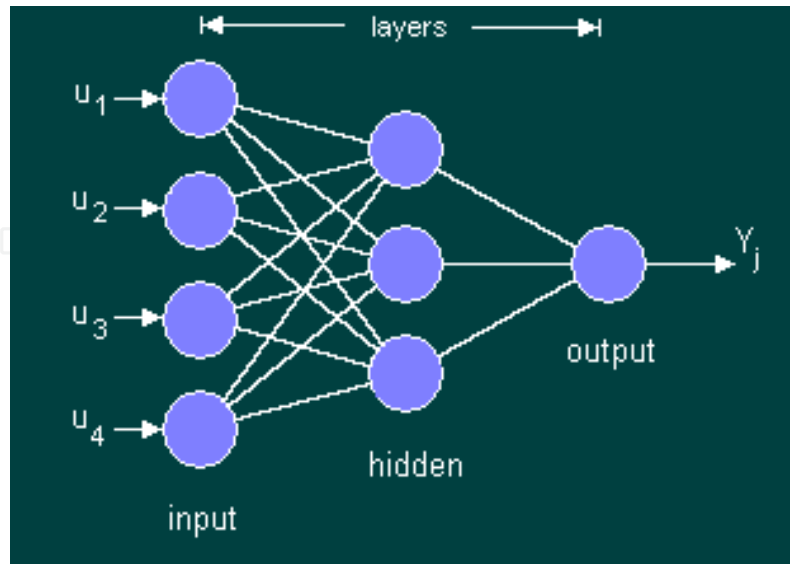


Fig. 4. MISO Back-propagation Network with three layers.

The layers are described as: input, distributing the signals from the external world; hidden, categorizing the signals; and the output collecting all features detected and producing a response. However the problem of the layers has many descriptions considering the set of optimal weights.

1.3 Neural network operation

The output of each neuron is a function of its inputs and weights, with a layer as described recursively in (1).

$$W_j(k) = u_n(k)w_{nj}(k) + W_{j-1}(k). \quad (1)$$

In where the basic function has the form $W_j(k) = \sum_{i=1}^n u_n(k)w_{ij}(k)$.

The output neural net answer is a convolution operation shown in (2).

$$y_j(k) = (F(k) \circ W(k))_j. \quad (2)$$

The $W_j(k)$ value is convoluted with a threshold value giving an approximate biological neural net answer, but in a computational sense it is active considering a $t_j(k)$, known as an activation function. The activation function usually is the sigmoid function shown in Figure 5. The output answer $y_j(k)$, is the neural net response, observing that the threshold function corresponds to biological electrical potential [90, 110] mill-volts needed in synopsis operations.

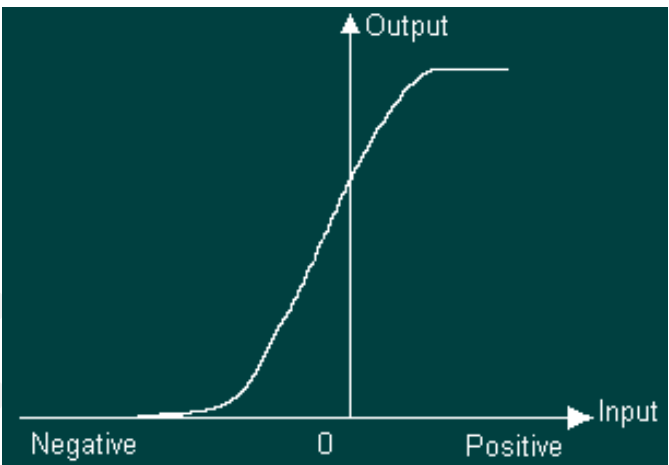


Fig. 5. Sigmoid Function as a neural net firing an electrical impulse noise.

The biological or computational fire answers correspond to threshold conditions that accomplish the excitation functions that permit generating an answer giving many inputs. Generally, the weights are selected intuitively in the first step; but with adaptive consideration can be adjusted to seek the desired answer.

2. Neural network adapting its weights usign fuzzy logic

The adaptation in a neural net means that it adapts its weights with a law action, seeking the convergence to the output desired answer. The difference between the desired ($d_j(k)$) and actual responses ($y_j(k)$) is known as convergence error ($e_j(k)$) and is defined as (3) and is shown in figure 6.

$$e_j(k) = d_j(k) - y_j(k).$$
 (3)

The law action could be a sliding mode, proportional gain in its weight and other non-linear models that allows the neural net system converging to the desired answer with respect to the input set.

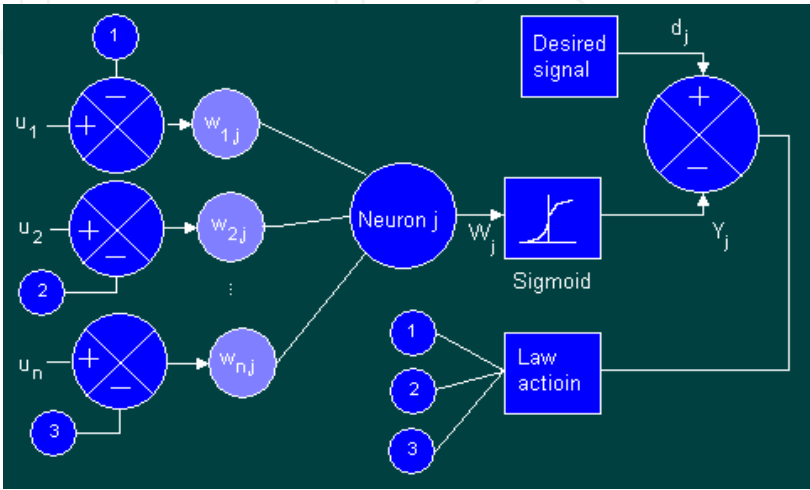


Fig. 6. Neuronal Weights Adjustment using a law action.

The adaptive back-*propagation* procedure is described in (4)

$$u'_{ij}(k) = u_{ij}(k) - L_j(k) \quad (4)$$

Where corresponds to law action considered by neural net designer.

Now applying the concept considered above with respect to neural net adjusting weights using fuzzy logic concepts gives a great advantage over traditional concepts such as the forgetting factor and sliding modes in other action laws.

The neural net that has adaptive weights is known as a digital identification filter, therefore, the neural net in where the adaptation process considered fuzzy inferences is known as a Neuro-Fuzzy Digital Filter.

A fuzzy neural net classifies, searches and associates information (Huang, Zhu and Siew, 2006) giving a specific answer value in accordance with a reference desired signal process, constructing a control volume described as $T_N = \{(y_j(k), \hat{y}_j(k))\} \subseteq R^2$ where a variant scheme has the form $T_N : (Y_N \times \hat{Y}_N) \times T \rightarrow \{(y_j(k), \hat{y}_j(k)), \tau(k)\}_{k=1}^N \subseteq R^2$ inside the membership intervals delimited by a Knowledge Base (KB) (Schneider y Kandel, 1996), with dynamical and bounded moments. The responses set into the KB represents all the possible correct filter responses (Gustafsson, 2000) (Margaliot and Langholz, 2000) (Zadeh, 1965) in accordance with an objective law, previously defined by the actual natural reference process in a distribution sense. The filtering mechanism adjusts the neural weights selecting the best answer from the KB when the state changes, to use fuzzy rules (*if-then*). The neuro-fuzzy filter is based on the back-propagation algorithm, because its weights have a dynamic actualization (Ali, 2003) (Amble, 1987) (Haykin, 1996) with different levels for each interval iteration (Huang, Zhu and Siew, 2006), using the error described as $e(k) \in R$ defined in (3) and considering its distribution function (García, Medel y Guevara, 2008) (Marcek, 2008)); this filter is shown in. Figure 7, integrating the fuzzy logic convenient actions into neural net structure using adaptive weights (Passino, 1998, and Medel 2008).

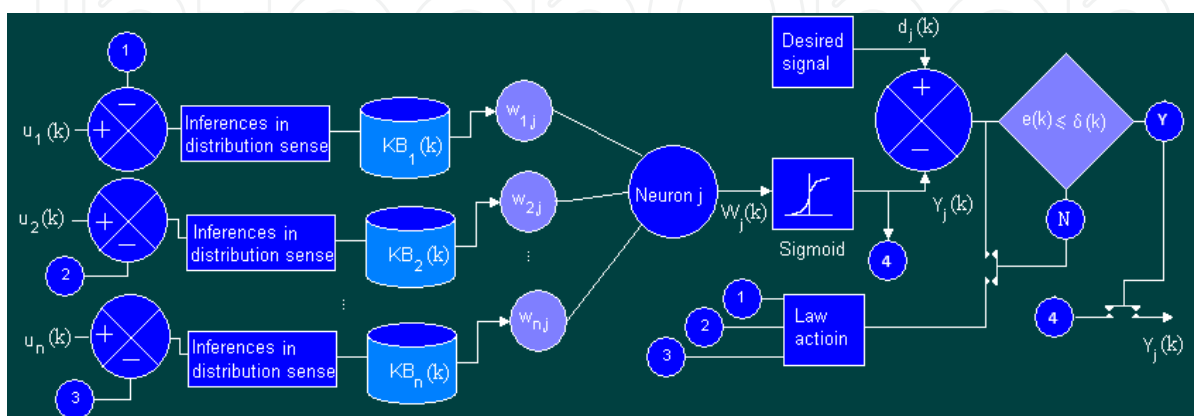


Fig. 7. Neuro-fuzzy Digital Filter Process

The error $\left(\left|e_j(k)\right|\right)$ has a interval limit $[0, \varepsilon]$ and ε is described as a positive value in accordance with $\inf\left\{\left|e_j(k)\right| \rightarrow \underline{\delta} > 0, \sup\left\{\left|e_j(k)\right| \rightarrow \bar{\delta} > \underline{\delta}\right\}\right\}$ in where $\varepsilon \in [\underline{\delta}, \bar{\delta}]$ (k is index interval) (Margaliot and Langholz, 2000) (Morales, 2002).

The Neuro-Fuzzy Digital Filter considers the concepts described in (Ash, 1970) (Abraham, 1991) (Ali, 2003) (Gustafsson, 2000), (Mamdani, 1974) (Morales, 2002) (Medel et.al 2008), having the elements needed in its basic description:

- Back propagation neural net scheme.
- Adaptive weights considering the law action and Fuzzy Logic inferences
- Convergence answer considering the stochastic error $e(k)$ and the its probability bounded moments
- In a metric sense, *weights distribution* is transformed in Fuzzy Inferences after the law action applied in the dendrites stage inputs.
- Rule Base* allows the interpretation of the stochastic weights bounded by distribution function accomplishing the actions using the logic *IF* connector.
- Inference Mechanism* as an expert consequence of the rule base, selects the membership function described as an adaptive weight $\left\{w_{ij}(k): i = \overline{1, n} \in Z_+\right\}_{j=\overline{1, m} \in Z_+}$ using logic *THEN* connector selecting the dendrite value corresponding to the knowledge base (Yamakawa, 1989).
- Activation Function* is the stage where the answer filter is transformed into a natural answer approximating to minimal convergence error region.
- Neuro-Fuzzy Digital Filter process has a *Natural Actualization* obtaining the linguistic values and actualizes its weights dynamically based on a distribution error and observing the second probability moment basic law action (5).

$$J_j(k) = \frac{1}{k^2} \left[e_j(k)^2 + (k-1)^2 J_j(k-1) \right] \in R_{[0,1)} \quad k \in Z_+. \quad (5)$$

The functional error $J(k)$ has an exponential convergence if the weights set into the Neuro-Fuzzy Filter allow that $\lim_{k \rightarrow \infty} |J_j(k)| \rightarrow m$, $m \in R_{[0,1)}$, if $0 < \left\{\left|e_j(k)\right|\right\} < 1$ and (6).

$$J_{\min} = \inf \left\{ J_{\min, j} := \min J(d_j(k), \hat{y}_j(k)), j = \overline{1, s}, s \in Z_+ \right\} \quad (6)$$

The Neuro-Fuzzy Digital Filter needs the knowledge base in order to select the corresponding answer weights in accordance with the desired signal and law action. Firstly, the filtering process uses the neuronal net and requires the adaptation process for non-stationary answer conditions, and the fuzzy rules adjust to the adaptation process guarantying the convergence rate (Takagi and Sugeno, 1986). The filter mechanism makes a selection of the correct variable weights into its data form and selects the correct level into three fuzzy regions. Then, the rules mechanism selects the weights $\left\{w_{ij}(k): i = \overline{1, n}\right\}_{j=\overline{1, m}}$ adjusting the filter gain, giving the correct answer $\hat{y}_j(k)$ as the filter output (Rajem y Gopal, 2006) (Medel et.al 2008), with MISO (Multi Inputs Single Output) properties.

3. Weight properties

A neutro-fuzzy filter has a weight set $\{w_{ij}(k): i = \overline{1, n}\}_{j=\overline{1, m}}$, where the knowledge base in each layer accomplishes the condition $\sum_{i=1}^n w_{i,j}(k) \leq 1$ without losing the transition function basic properties (Medel, 2008):

i. Each weight has a dynamic transition function with natural restrictions:

1. $\ln(\Phi_j(k)) < \infty$, 2) $\ln(\Phi_j(k)) > 0$, 3) $\ln(\Phi_j(k))k^{-1} < 1$.

ii. The weight is described using the transition function in (7).

$$w_j(k - k_0) = \ln \Phi_j(k) \left(\ln \phi_j(k_0)(k - k_0) \right)^{-1}, \quad (7)$$

iii. The velocity changes are limited inside the transition function (8).

$$\ln(\Phi_j(k)) \leq \ln \Phi_j(k_0)(k - k_0)^T, \quad \ln \Phi_j(k_i) \leq \ln(\Phi_j(k_i - 1)(k_i - (k_i - 1)))^T \quad (8)$$

The transition functions sum is bounded in each layer $0 \leq \sum_{i=1}^n \Phi_{ij}(k) / \leq 1$. In accordance with the value of $\ln \Phi_j(k_0)$, the weights are bounded considering (9).

$$w_j(k - k_0) \leq \ln \Phi_j(k_0) \quad (9)$$

The identifier described as (10) considered (6).

$$\hat{x}_j(k) = w_{ij}(k)(k - k_0)\hat{x}_j(k - 1) + K_j(k)\hat{w}_j(k) \quad (10)$$

Where $K_j(k)$ is the function gain and is a functional identification error, defined by the second probability moment (5), $\hat{w}_j(k)$ represents generalized perturbations with $\{\hat{w}_j(k)\} \subseteq N(\mu, \sigma^2 < \infty)$.

4. Results

The MISO neuro-fuzzy filter, considers the digital filter structure (Hayking, 1996) with the transition matrix bounded by the knowledge base in accordance with the functional error criterion (Ash, 1970). The soft system (statistic in variance sense) considers the evolution times bounded by PC with AMD Sempron 3100+ processor performance at k intervals, with an average evolution time of $0.004 \text{ sec} \pm 0.0002 \text{ sec}$.

This chapter uses the first order difference discrete ARMA model (11) representing a reference system with $j=1$.

$$x(k + 1) = a(k)x(k) + w(k) \quad (11)$$

And the output described as (12).

$$y(k) = x(k) + v(k)$$

(12)

$$x(k), y(k), w(k), v(k) \in R, a(k) \in R_{(-1,1)}$$

$x(k)$ is the internal states vector, $a(k)$ is the parameter, $w(k)$ is the vector noise into the system, $y(k)$ is the reference vector desired system signal and, $v(k)$ is the output vector noise.

The different operational levels are described in order to operate the distribution function error. The filter process establishes in the fuzzy region the linguistic descriptors adjusted in its ranges. Figure 8 describes the reference signal and its identification without knowing the internal parameter model $\hat{a}(k) \in R_{(-1,1)}$.

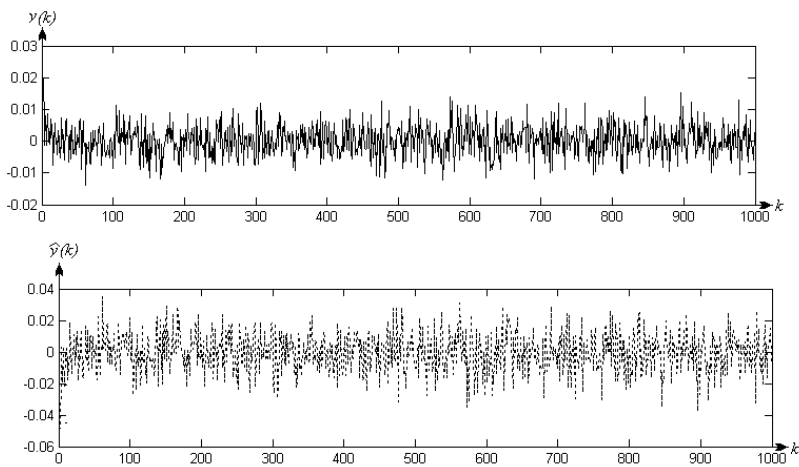


Fig. 8. Output signal $Y(k)$ and its identification $\hat{Y}(k)$ using the nero-fuzzy digital filter technique.

The fuzzy regions considered the distribution weights after applying the law action.

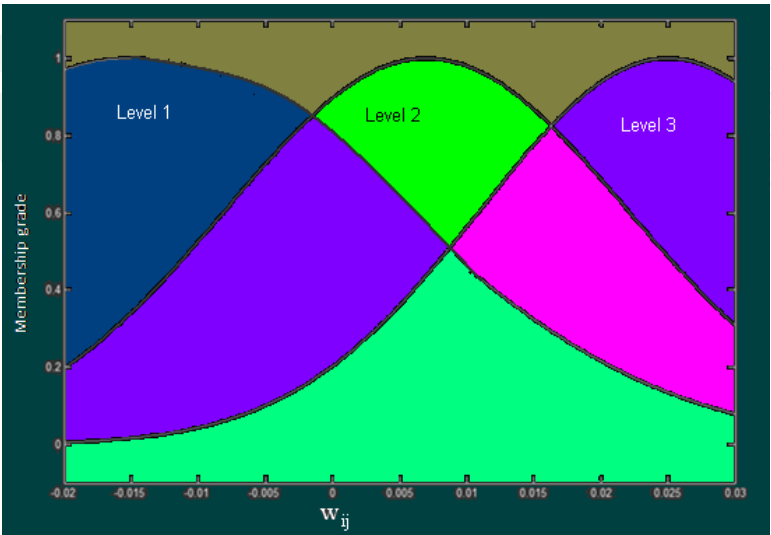


Fig. 9. A membership weights function.

The histogram identification evolution is shown illustratively in Figure 10, in where each weight is adjusted in neuro-fuzzy filter affecting the identification histogram convergence. The convergence in histogram is associated with the membership weights function, allowing that the identification system tends to the reference system.

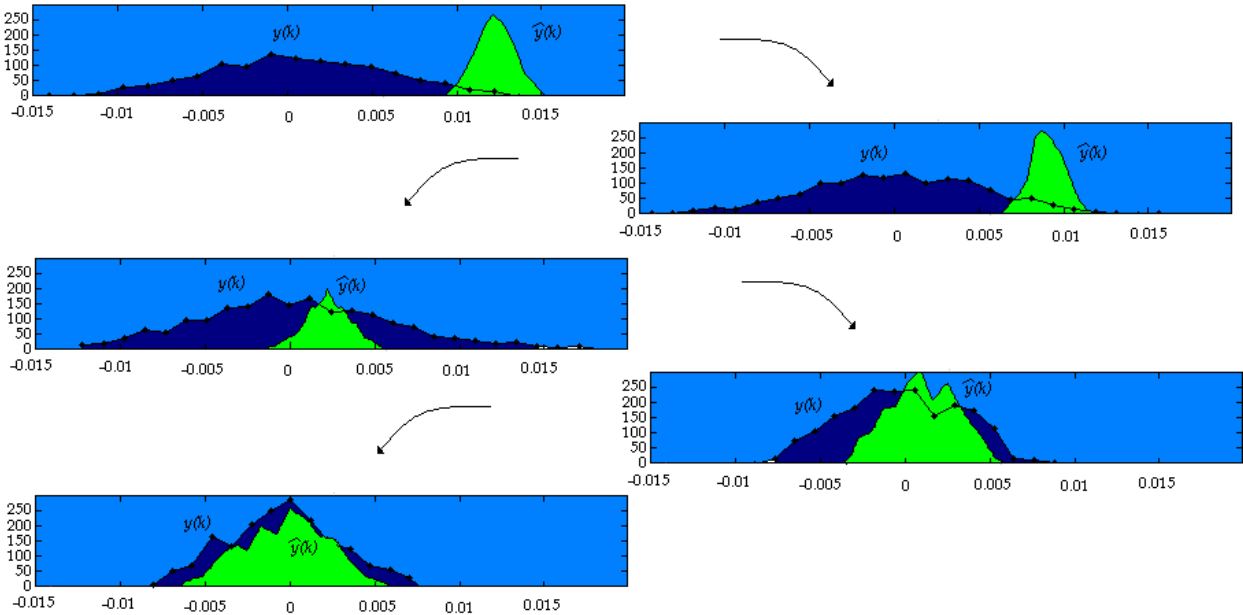


Fig. 10. The histogram convergence through the time evolution between the identification with respect to reference signal in base to adaptive weights.

Figures 11 and 12 show the $\hat{Y}(k)$ and desired signal $Y(k)$ final *histograms*, respectively.

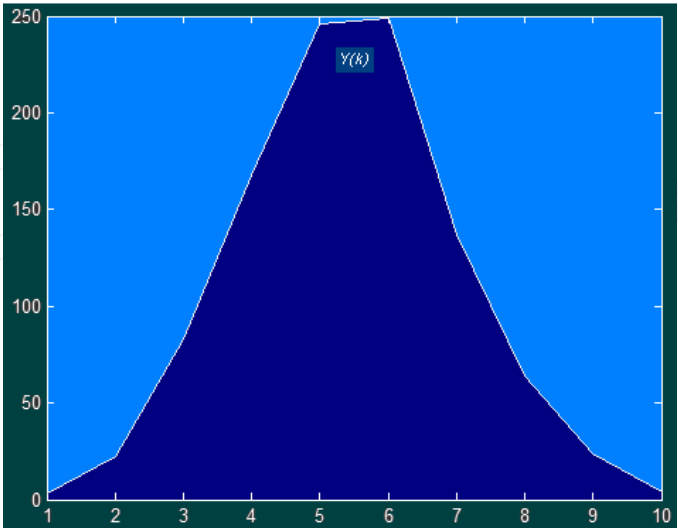


Fig. 11. Histogram of desired signal described as $Y(k)$

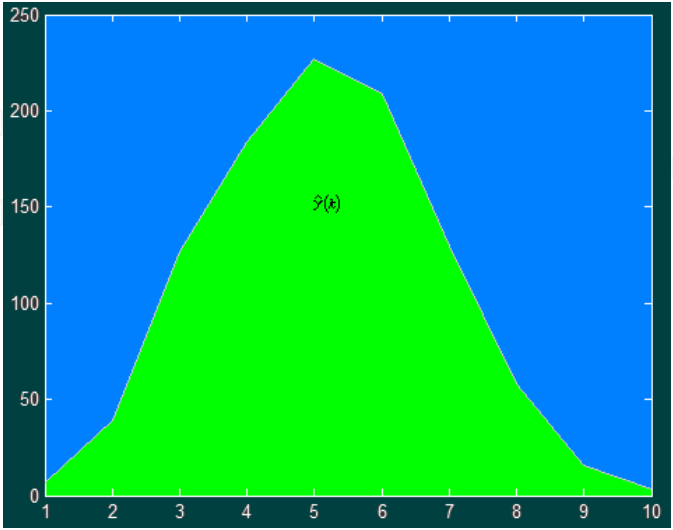


Fig. 12. Histogram of identification signal as $\hat{Y}(k)$

Figure 13, shows both overlapping final histogram considering the same time interval.

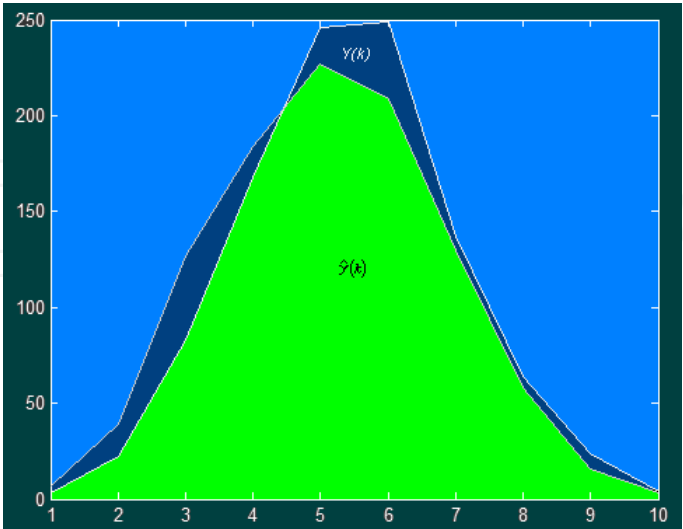


Fig. 13. Overlapping both final histograms with respect to $Y(k)$ and $\hat{Y}(k)$, respectively.

Figure 14, shows the evolution functional error described by (5).

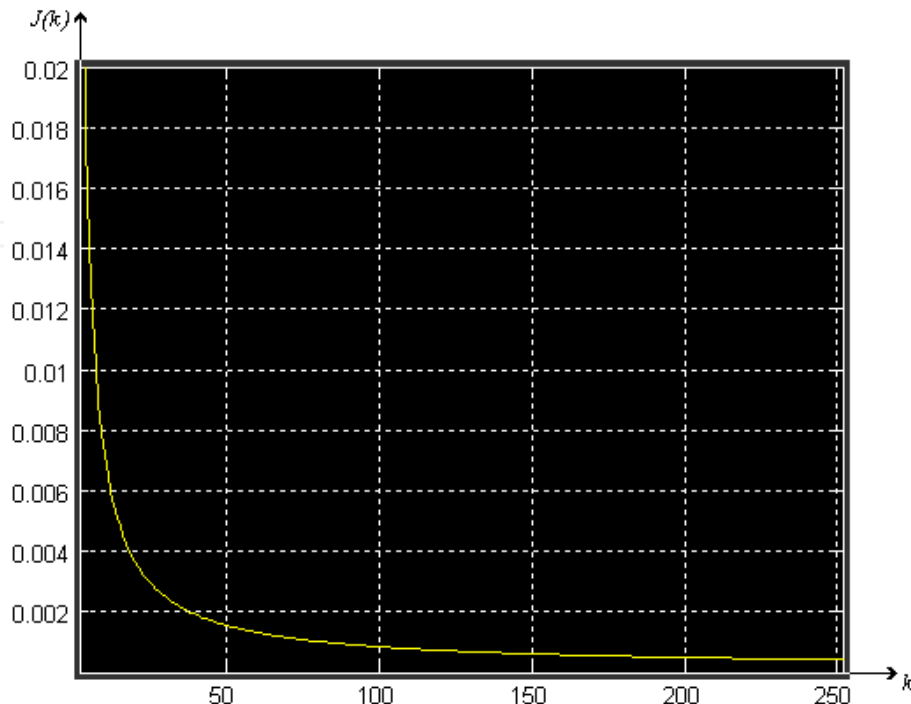


Fig. 14. Functional error considered in (5).

The Neuro-Fuzzy Digital Filter time evolution responses was less that the reference process time state change proposed with a value of 0.08 sec, and is delimited by the processor, considered in (10). The convergence time is 0.0862 sec, described in (Medel, 2008).

5. Conclusion

Neural net in identification sense, considered the adaptation process requiring adjust the weights dynamically using the common proportional condition. But in many cases, these applications generate convergence problems because the gain in all cases increase the neural net weights positive or negatively without converge to desired value. In the black box traditional scheme the internal weights are known, but in real conditions it is impossible and only has a desired or objective answer. But, the neural nets help jumping weights estimation adjusting dynamically their internal weights, needing adaptation process with smooth movements as a function of identification error (function generated by the difference between the filter answer with respect to desired answer.). An option considered was the fuzzy logic in where its natural actions based on distribution function error allowing built the adjustable membership functions and mobile inference limits. Therefore, the neural weights are adjusted dynamically considering the fussy logic adaptable properties applied in the law actions, shown in figure 7. Stable weights conditions were exposed in section 3, with movements bounded in (8). In the results section, the figure 13, illustrated the Neuro-Fuzzy Digital Filter advantages without lost the stability with respect to desired system answer observed in distribution sense, observing the Hausdorff condition approximating the filter to desired system answer in distribution sense.

6. References

- Abraham K. (1991). *Fuzzy Expert Systems, Florida*, Ed. CRC Press, ISBN: 9780849342974
- Ali H. S. (2003). *Fundamentals of Adaptive Filters*, Ed. John Wiley & Sons, New Jersey USA. ISBN.
- Amble T. (1987). *Logic Programming and Knowledge Engineering*. Addison Wesley, USA,. ISBN.
- Ash R. (1970). *Real Analysis and Probability*, Ed. Dover Publications. USA, ISBN: 0486472205
- García J.C., Medel J.J., Guevara P. (2007), RTFDF Description for ARMA Systems, *WSEAS Journal: Transactions on Systems and Control*, Vol. 2, Issue 10, pp. 491-496.
- García J.C., Medel J.J., Guevara P., (2008). Filtrado Digital Difuso en Tiempo Real, *Computación y Sistemas Journal*, Vol. 11, No. 4, pp. 390-401, ISSN: 1405-5546.
- García J.C., Medel J.J., Guevara P., (2008). Real-time neuro-fuzzy digital filtering: Approach, *Computer and Simulation in Modern Science Journal, WSEAS press selected papers*, Vol. 1, pp. 122-125.
- Haykin S. (1996). *Adaptive Filtering*, Prentice Hall, USA.
- Huang G., Zhu K., Siew C. (2006). Real-Time Learning Capability of Neural Networks, *IEEE Transactions on Neural Networks*, Vol. 17, pp. 863-878.
- Gustafsson F. (200). *Adaptive Filtering and Change Detection*, John Wiley and Sons, England.
- Mamdani E. (1974). Applications of Fuzzy Algorithms for Control of Simple Dynamic Plant, *Proc. IEEE*, Vol. 121, pp. 1585-1588.
- Manuel L. (2003). *Lebesgue Integral Measure Theory*, Ed. Del Rosario University, Buenos Aires.
- Margaliot M., Langholz and G. (2000). *New Approaches to Fuzzy Modeling and Control Design and Analysis*, Singapore, Ed. World Scientific.
- Medel J.J., Guevara P. (2004). Caracterización de Filtros Digitales en Tiempo-real para Computadoras Digitales, *Computación y Sistemas Journal*, Vol. VII, No. 3, ISSN:1405-5546
- Medel J.J., García J.C., Guevara P. (2008). Real-time Fuzzy Digital Filters (RTFDF) Properties for SISO Systems, *Automatic Control and Computer Sciences, AVT*, Vol. 42, No. 1, pp. 26-34.
- Medel J.J., García J.C., Sánchez J.C. (2008). Real-time Neuro-Fuzzy Digital Filtering: Basic Concepts, *WSEAS Transactions on Systems and Control*, 2008, Vol. 3, Issue 8, pp. 654-663.
- Marcek D. (2004). *Statistical, Classical and Fuzzy Neural Networks, Modeling Decisions for Artificial Intelligence*, Ed. Springer Verlag, pp. 41-48.
- Morales G. (2002), *Introducción a la Lógica Difusa*, Ed. Cinvestav, Mexico.
- Nikola K. (1996). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, Hong Kong, The MIT Press.
- Passino K. M. (1998). *Fuzzy Control*, USA, Addison Wesley.
- Rajen B., Gopal M. (2006), Neuro-Fuzzy Decision Trees. *International Journal of Neural Filters*, Vol. 16, pp. 63-68.
- Shannon M. (1948). A Mathematical Theory of Communication, *Bell Systems Technical Journal*, Vol.27, pp.379-423 and pp. 623-656.
- Schneider M., and Kandel A. (1996). *Fuzzy expert systems tools*, England, John Wiley & Sons.
- Takagi T., Sugeno M. (1986). Fuzzy Identification of Systems and its Applications to Modelling and control, *IEEE Transactions and Systems, Man, and Cybernetics*, Vol. 15, pp. 116-132.

Yamakawa F. (1989). *Fuzzy Neurons and Fuzzy Neural Networks*.

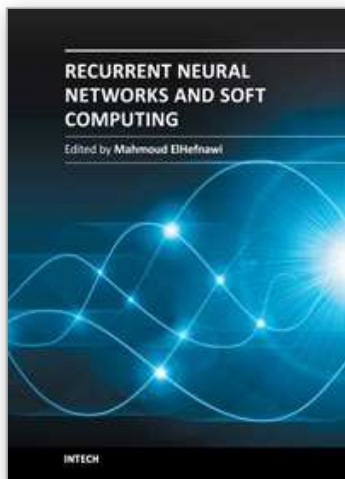
Zadeh L. (1965). Fuzzy Sets, *Information and control*, Vol. 8, pp. 338-353.

National security. (2008). Intelligence on the Brain, A New Dialogue on Neuro- Research and National Security

<http://www.scienceprogress.org/2008/11/intelligence-on-the-brain/>

IntechOpen

IntechOpen



Recurrent Neural Networks and Soft Computing

Edited by Dr. Mahmoud ElHefnawi

ISBN 978-953-51-0409-4

Hard cover, 290 pages

Publisher InTech

Published online 30, March, 2012

Published in print edition March, 2012

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

José de Jesús Medel, Juan Carlos García and Juan Carlos Sánchez (2012). Neuro-Fuzzy Digital Filter, Recurrent Neural Networks and Soft Computing, Dr. Mahmoud ElHefnawi (Ed.), ISBN: 978-953-51-0409-4, InTech, Available from: <http://www.intechopen.com/books/recurrent-neural-networks-and-soft-computing/neuro-fuzzy-digital-filter>



InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

intechOpen

© 2012 The Author(s). Licensee IntechOpen. This is an open access article distributed under the terms of the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

IntechOpen

IntechOpen