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.

![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 \(2.0 \times 10^9 \text{ m sec}^{-1}\) (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.

![Neuron device computational model](image)

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.
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).

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

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).

\[
y_j(k) = (F(k) \circ W(k))_j .
\]

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.
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 using 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.

$$e_j(k) = d_j(k) - y_j(k).$$  \hspace{1cm} (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)

\[ u_{ij}(k) = u_{ij}(k) - L_j(k) \]  

(4)

Where \( L \) 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 \( T_N = \{(y(k), \hat{y}(k))\} \subseteq R^2 \) where a variant scheme has the form \( T_N : (Y \times \hat{Y}) \times T \rightarrow \{(y(k), \hat{y}(k), r(k))\} \subseteq R^2 \) 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 (Garcia, 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 \( |e_j(k)| \) has a interval limit \([0, \varepsilon]\) and \( \varepsilon \) is described as a positive value in accordance with \( \inf |e_j(k)| \to \delta > 0, \sup |e_j(k)| \to \bar{\delta} > \delta \) in where \( \varepsilon \in [\delta, \bar{\delta}] \) (k is index interval) (Margaliot and Langholz, 2000) (Morales, 2002).


- 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 \( \{w_{ij}(k) : i = \overline{1, m} \in \mathbb{Z}_+ \} \) 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).

\[
J_j(k) = \frac{1}{k^2} \left[ e_j(k)^2 + (k - 1)^2 J_j(k-1) \right] \in \mathbb{R}_{[0,1)} \quad k \in \mathbb{Z}_+.
\] (5)

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

\[
J_{\min} = \inf \left\{ J_{\min,j} := \min \{J(d_j(k), \hat{y}_j(k)) : j = \overline{1, s}, s \in \mathbb{Z}_+ \} \right\}
\] (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 \( \{w_{ij}(k) : i = \overline{1, m} \} \) 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 neuro-fuzzy filter has a weight set $\{w_{ij}(k) : i = 1, \ldots, n, j = 1, \ldots, m\}$, 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).

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

(7)

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

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

(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).

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

(9)

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

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

(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 sec ± 0.0002 sec.

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

$$x(k+1) = a(k)x(k) + w(k)$$

(11)
And the output described as (12).

\[ y(k) = x(k) + v(k) \]  

(12)

\[ x(k), y(k), w(k), v(k) \in \mathbb{R}, a(k) \in \mathbb{R}_{(-1, 1)} \]

\( x(k) \) 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 \( \hat{a}(k) \in \mathbb{R}_{(-1, 1)} \).

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

and National Security