

# Design of Neural Networks on microcontrollers, applied in functional modules for an eco-park

David J. Steckler V.<sup>a</sup>, Cesar Nava<sup>a</sup>, Jonattan Duarte<sup>a</sup>, José Jhoan Zambrano<sup>a</sup> y Cecilia E. Sandoval-Ruiz<sup>\*,b</sup>

<sup>a</sup>*Dirección de Postgrado, Facultad de Ingeniería, Universidad de Carabobo, Venezuela.*

<sup>b</sup>*Instituto de Matemática y Cálculo Aplicado, Facultad de Ingeniería, Universidad de Carabobo, Venezuela.*

**Abstract.-** This paper presents a technique for implementing neural networks -RNA in hardware, from a prototype to scale to improve the management and production of self-generated electrical energy. The method consisted in the modular design of applications, using a micro-controller, for the control of the subsystems, which interact with each other, with the aim of providing efficient service to an ecological park, incorporating a pet food dispenser, conversion of hybrid electric power (solar and turbine for energy production from rainwater), a predictor of electricity consumption. This research consists of the configuration of the neural network models for each module, their training, tests of network behavior and validation, through the tool designed for the management and programming of subroutines in the micro-controller. The result is an integrated neural system, with monitoring of the target for each function and a didactic platform for modeling adaptive and multilayer neural networks. Its contribution is given by the design of neuronal applications, ideal for its flexibility, oriented to sustainable technology and intelligent control.

**Keywords:** neuro-control; adaptive network; multilayer network; microcontroller; ecological park; renewable sources.

## Diseño de Redes Neuronales sobre microcontroladores, aplicadas en módulos funcionales para un eco-parque

**Resumen.-** Este trabajo presenta una técnica de implementación de redes neuronales -RNA en hardware, a partir de un prototipo a escala para mejorar la gestión y producción de energía eléctrica autogenerada. El método consistió en el diseño modular de aplicaciones, usando un micro-controlador, para el control de los subsistemas, que interactúan entre sí, con el objetivo de proveer servicio eficiente a un parque ecológico, incorpora un dispensador de alimento para mascotas, sistemas de conversión de energía eléctrica híbrida (solar y turbina para producción de energía a partir de agua de lluvia), un predictor de consumo eléctrico. Esta investigación consta de la configuración de los modelos de red neuronal para cada módulo, su entrenamiento, pruebas de comportamiento de la red y validación, a través de la herramienta diseñada para el manejo y programación de subrutinas en el micro-controlador. Se obtiene como resultado un sistema neuronal integrado, con seguimiento del target para cada función y una plataforma didáctica para modelado de redes neuronales adaptativas y multicapa. Su aporte está dado por el diseño de aplicaciones neuronales, idóneo por su flexibilidad, orientado a tecnología sostenible y control inteligente.

**Palabras claves:** neuro-control; red adaptativa; red multicapa; microcontrolador; parque ecológico; fuentes renovables.

Recibido: agosto 2017

Aceptado: marzo 2018

### 1. Introducción

Actualmente se plantean objetivos de desarrollo sostenible, que apuntan a la producción de una energía sin emisiones contaminantes, lo que ha

definido un área de investigación en la materia, como estrategia de solución a los efectos propios de problemas ambientales. En tal sentido, el grupo de investigación en Redes Neuronales Aplicadas al procesamiento de señales, se ha propuesto el desarrollo de un sistema de control neuronal, orientado a un parque ecológico (en las inmediaciones de la Universidad), como solución didáctica para promover los diseños sostenibles. Esto por tratarse de un punto estratégico que tendrá impacto positivo en la comunidad universitaria,

\*Autor para correspondencia:

Correo-e:cecisandova@yahoo.com (Cecilia E. Sandoval-Ruiz)

sobre el cual se pueden formular nuevas propuestas. En trabajos previos se han desarrollado sistemas que integren módulos de conversión de energía renovable [1, 2], siendo relevante el diseño del sistema de control neuronal [3, 4, 5], para mejorar el tratamiento de las señales en sistemas de ingeniería sostenibles. Además, se ha considerado el parque como un sistema eco-adaptativo [6], incorporando un conjunto aplicaciones, como solución a la demanda de energía de éste, a la vez de presentar un desarrollo tecnológico.

Algunos conceptos como diseño sostenible, Smart Grid [7], aplicaciones inteligentes en urbotica, sistemas eco-adaptativos [6], neuro-control en microredes de conversión de energías alternativas, aportan los avances en tecnología de redes neuronales aplicadas a energías renovables. El propósito de esta investigación es desarrollar un sistemas de control neuronal, basado en programación de microcontroladores en lenguaje C (versión libre de prueba), partiendo de conceptos de inteligencia artificial, como lo son las redes neuronales artificiales. Esto bajo criterios de control orientado a eficiencia energética, dadas las tendencias en energía renovables [8]. Es así como se propone un conjunto de funciones para un parque ecológico, definidas como: seguimiento de máxima eficiencia [5, 9]; un gestor de fuentes de energía, incorporando energía hidroeléctrica a pequeña escala, energía solar [4], estimación del consumo de energía [10] y predicción adaptativa, así como la automatización de un módulo dispensador de alimentos para la fauna del parque [11]. Tales ideas agregan la independencia y autonomía necesarias para mayor optimización de un sistema integrado.

Una vez definido el sistema a diseñar, se seleccionó el hardware de soporte, tomando en consideración trabajos con implementación de redes neuronales en hardware [12, 13, 14], el estudio de las tecnologías disponibles, las ventajas de su diseño en aplicaciones de dispositivos electrónicos inteligentes [15] y estudiando la programación de las redes neuronales sobre micro-controladores [16], configuración la topología de la red, el método de entrenamiento en circuito y el diseño de una interface de control neuronal de los procesos, la cual permita seleccionar las acciones a

tomar y el monitoreo del comportamiento de la red. Se seleccionó esta tecnología por las características del diseño modular, con puertos de manejo de señales del parque ecológico.

El proyecto comprende el diseño e implementación de un conjunto de modelos de control electrónico en hardware, a través de diversas redes neuronales, seleccionadas de acuerdo a su función: conversión de energía eléctrica y manejo de accionamientos, así como un análisis del impacto ambiental del diseño, que se vería reflejado en un balance eficiente de las cargas con la red eléctrica pública, ayudando a bajar el consumo energético a partir de fuentes alternativas. Partiendo del diseño conceptual del sistema integrado de convertidores de energía renovable y los módulos inteligentes, para lo cual se establece una selección de tipos de redes neuronales, diseño, entrenamiento y validación.

## 2. Diseño del Sistema de Control Neuronal

Este diseño está basado en la arquitectura de redes neuronales artificiales, conformadas por un conjunto de neuronas, que intercambian información, a través de conexiones, cada una de éstas tendrá una ponderación relacionada con el proceso de aprendizaje. La información procede de las señales de entrada a la red  $p(n)$ , tendrá una estructura de capa oculta, capa de salida que procesan la información, a fin de obtener las señales de salida  $a(n)$  de la red RNA. Otro aspecto de interés es los métodos de optimización del algoritmo de aprendizaje, en este caso una vez identificada la estructura del ponderador, representado por:  $w_i \cdot x_i$  puede ser implementado en hardware como una neurona, de manera que al optimizar el entrenamiento, el valor de  $w_i$  se puede establecer por aproximaciones sucesivas, requiriendo  $b$  iteraciones, de acuerdo al número de bits del peso, de esta manera se puede optimizar, de manera más eficiente, empleando el método de proporcionalidad de los aportes por neurona basados en  $\Delta e(n)$ .

En este proyecto, se abordaron las redes con topología predictiva lineal simple, lineal adaptativa y redes multicapa con interacción hacia adelante –

*FeedForward* (usando el Neural Network Toolbox), las cuales son importantes para el tratamiento de las señales y reconocimiento de clases. Estas redes se reconocen por su arquitectura dada en función al número de capas y la definición del algoritmo de entrenamiento: *incremental*, para la adaptación de los parámetros desde el circuito, con las entradas y targets de la red o *BackPropagation* [5], [17] para redes multicapa – MPL, con interacción hacia adelante, con la propagación hacia atrás del error, se maneja un diferencial del error de la red, para corregir los pesos sinápticos de la capa oculta, que son proporcionales a la relación de la función de la neurona estudiada, estos algoritmos se presentan en la Figura 1, los cuales serán la base de la programación para su implementación sobre el microcontrolador.

```

x(n) = leer puerto; ←
patrón a entrenar (vector de R entradas)
e(n) = t(n) - w(n).x(n); error de la neurona para el patrón n
ea = e(n) + ea; error acumulado
-- Cálculo del error para el patrón x(n) → t(n)
For i=1 to R para cada entrada n patrones
wi(n+1) = wi(n) + α·e(n)·xi(n) -- Se actualizan los pesos next
-- Verificación de condición de parada
et(i) = ea / n ; error promedio de la iteración
if et(i) < 0.00999 then comprobación del error
Goto Entrenamiento; si el error es mayor ir nueva iteración
endif;
    
```

(a) Incremental.

```

Para cada patrón p hacer:
Asignación de las entradas i hacer:
xi ← pi
Para cada capa c+ hacer: ←
in_j ← Σ_{i=0}^n w_{j,i} · xi
yj ← g(in_j)
Para cada nodo k en la capa de salida hacer:
Δek ← g'(in_k) · (yk - tk) ←
Para cada capa c (M-1 a 1) hacer:
Para cada nodo j de la capa c hacer:
Δej ← g'(in_j) · Σ_{j=0}^n w_{k,j}(n) · Δek
Para cada neurona j de la capa c+1 hacer:
w_{j,i}(n+1) ← w_{j,i}(n) + α · Δej · x_j(n)
Hasta que se alcance el criterio de parada
    
```

(b) BackPropagation.

Figura 1: Algoritmos de Entrenamiento para las RNA.

Los módulos componentes han sido definidos como: (1) Sistema de reconocimiento, basado en características (codificadas en binario), para ofrecer una clasificación y salida proporcional, de

acuerdo a la definición establecida, para aplicación de un dispensador inteligente de alimentos para mascotas [11], éste se basa en una red lineal, la cual mediante sensores de detección de características comprueba la presencia, detecta la forma de dos tipos de animales, así como la información de peso y tamaño, donde la red lineal decide tipo y cantidad de alimento. (2) Sistema Predictor simple para carga de baterías con mini-turbinas accionadas por agua de lluvia, (3) Sistema Predictor adaptativo para consumo eléctrico, implementado mediante una red neuronal predictiva adaptativa, ésta permitirá la conexión y desconexión del suministro directo a la carga de la red, permitiendo así la extensión de la durabilidad de las baterías u otro dispositivo de almacenamiento de energía y el aporte extra de energía solar en horas no pico, siendo esta práctica más ecológica al alargar la vida útil del dispositivo, así como un (4) Prototipo de optimizador de un sistema foto-voltaico, con controlador de posicionamiento de paneles solares, aplicando una red neuronal MPL hacia adelante, la cual permitirá el seguimiento en tiempo real de la máxima irradiación solar y así el aprovechamiento máximo de la conversión de energía eléctrica, para máxima eficiencia, todo esto basado en redes neuronales.

De esta manera, se han establecido los sub-sistemas, para un parque, el cual busca el aprovechamiento máximo de energía del medio ambiente, para mantener un principio ecológico en sus actividades, las cuales provean un servicio a la comunidad visitante y a sus mascotas Figura 2.

*Descripción de los módulos inteligentes del parque ecológico*

En primer lugar la aplicación del dispensador de alimentos para la fauna del parque, el cual se diseñó a través de una red neuronal con función lineal, donde se entrenó la red a partir de 24 patrones que están formados por 13 bits cada uno representados por diferentes sensores, la salida de la red neuronal responderá con un numero dependiendo de sus respectivas entradas binarias y luego tomará una decisión para dosificar el tipo de alimento y la cantidad necesaria. Las 13 entradas binarias estarán conformadas de la

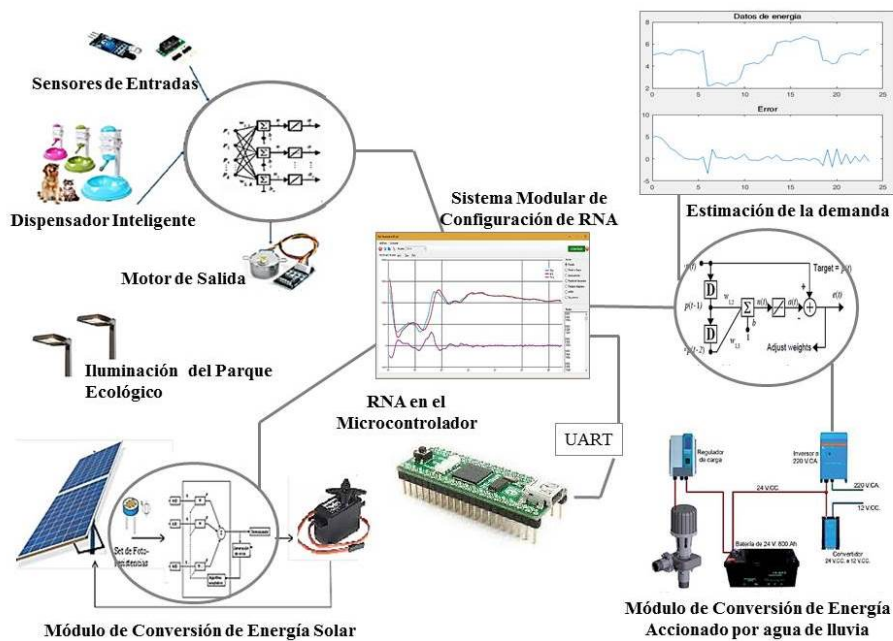


Figura 2: Esquema conceptual del sistema modular del parque ecológico.

siguiente manera: 2 bits corresponden a la salida de un sensor de peso digital, 2 bits dados por un sensor óptico para la altura del animal, 9 bits para clasificar tipo de animal (a partir de un pre-procesamiento de imagen capturada por una cámara digital), el accionamiento de salida estará dado por un motor de paso para la dosificación de alimento. Al encontrarse un animal en el dosificador de alimentos, la red neural estará en la capacidad de entregar la cantidad de alimento necesario según su clasificación (perros y gatos), peso (S, M, L, XL) y altura (P, M, G, EG), esto lo hará con un motor paso a paso que será programado para dar N vueltas según la salida de la RNA.

El siguiente módulo diseñado corresponde al predictor de la demanda para control para la configuración de los convertidores, donde se tiene la posibilidad de alimentar el parque ecológico, por una turbina accionada por un sistema de tanques que gestionan el agua de lluvia capturada y se emplea en los casos que la red tenga requerimientos adicionales. Este subsistema se encarga de monitorear la carga de las baterías, a través del voltaje en sus terminales. La condición para un voltaje superior a 26,6 VDC implica un cierre en la válvula que permite el flujo de agua

por la mini-Turbina, la cual aprovecha el flujo y la presión del agua para generar electricidad (24 VDC). El sistema predictor del consumo del parque es utilizado para determinar el consumo general del parque, durante las 24 horas del día, mediante un predictor adaptativo con una memoria o TDL de 5 retrasos. El sistema determina mediante datos de consumo promedio de potencia, la conexión eléctrica más favorable para mantener una reserva de energía importante almacenada.

Para el módulo de control de posición de la celda fotovoltaica, se seleccionó una red multicapa, mediante una red hacia adelante (*FeedForward*). Se diseñó el control de posición para un panel fotovoltaico, el cual permitirá el seguimiento de la trayectoria del sol, se realizará la lectura de las tensiones respectivas usando LDRs (*Light Dependent Resistor*) y se definió un conjunto de patrones, basados en una prueba experimental. Se toman valores de entrada a la red referentes a tensiones de intensidad de radiación y la variable salida como la magnitud y sentido de giro dando una posición al servomotor, se ejecuta el cálculo de la inclinación más apropiada, el sistema fotovoltaico detecta el cambio de intensidad solar recibida y ajusta la posición en tiempo real. Por

otra parte, se tiene la red lineal adaptativa, como predictor, de acuerdo a la salida del sistema en el tiempo, destacando que su implementación en circuito, aporta eficiencia al sistema híbrido de conversión de energía.

El modelo neuronal para el predictor viene dado por la Ecuación (1).

$$a(k) = \sum_{i=1}^R \omega_i p(k-i) + b(k) \quad (1)$$

Donde  $a(k)$  corresponde a la salida de la red neuronal,  $w_i$  la matriz de pesos sinápticos,  $p(k)$  el patrón de entrada para el elemento  $i$  del TDL, es decir, la muestra retrasada y  $b(k)$  la polarización de la red. Para este caso, se realizó el desarrollo de la neurona, tomando  $R = 14$  registros del TDL, siendo este modelo matemático el que se utilizó para la programación del comportamiento de la red en el micro-controlador (ver ecuación (2)).

$$\begin{aligned} a(n) = & \omega_1 p(1) + \omega_2 p(2) + \omega_3 p(3) \\ & + \dots + \omega_{11} p(11) + \omega_{12} p(12) \\ & + \omega_{13} p(13) + \omega_{14} p(14) + b \end{aligned} \quad (2)$$

Se utilizó el algoritmo de aprendizaje regla delta LMS para actualización de pesos sinápticos, dado por la expresión de la ecuación (3).

$$\omega(n) = \omega(n-1) + (T(n) - a(n))p(n)^T \quad (3)$$

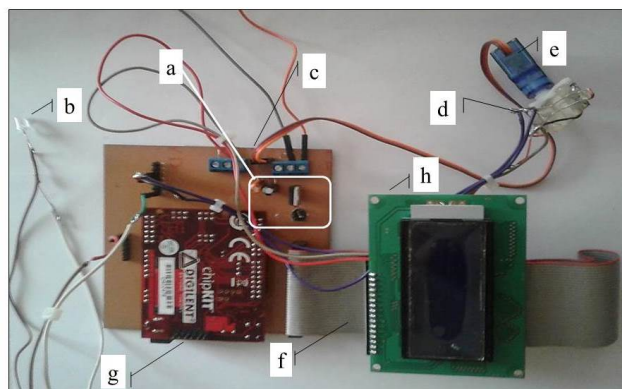
Donde  $\omega(n)$  será el valor del peso nuevo, a partir del peso anterior  $\omega(n-1)$ , corregido en función de un coeficiente de aprendizaje por el error dado por la diferencia entre el target  $T(n)$  y la salida de la red  $a(n)$ , en proporción de la entrada  $p(n)$ .

### 3. Diseño modular de las redes neuronales sobre micro-controlador

Inicialmente, se realizó el diseño de las redes según los requerimientos de cada función del parque ecológico (partiendo de las señales definidas en la descripción previa). Para cada módulo diseñado se consideró el tipo de red, partiendo de la configuración más sencilla, a fin de optimizar los recursos en el hardware de implementación.

### Métodos y materiales

La implementación del hardware se llevó a cabo mediante un microcontrolador PIC de 32 bits a 40 MHz (20 MIPS). Los componentes: celda solar de calculadora, servo motor, regulador 7805, Capacitor electrolítico 0,1  $\mu$ F, capacitor 0,3  $\mu$ F, resistencia 20 $\Omega$ , led indicador, pantalla LCD, etc. Se diseñó un prototipo de prueba para el control de posición de una celda fotovoltaica, tal como se presenta en la Figura 3.



Componentes Circuitales: (a) Circuito Regulador: Regulador 7805, Capacitor electrolítico 0,1 $\mu$ F, Capacitor electrolítico 0,33 $\mu$ F, Resistencia de precisión 20k $\Omega$  $\pm$ 2%. (b) Led Indicador. (c) Placa Baquelita 10x10. (d) Arreglo de Conversión Solar: Celda Solar 2,8 x 1,5, Sensores LDR. (e) ServoMotor. (f) Cable Plano. (g) Microcontrolador. (h) Pantalla LCD

Figura 3: Circuito de implementación de la red neuronal.

En la Tabla 1, se presentan los datos recolectados, realizando un experimento con el prototipo diseñado, se realizaron mediciones para el análisis del predictor obteniendo la conversión, a partir de la intensidad de luz solar.

Se observa el comportamiento de la irradiación solar tomada durante el día. La red del predictor dará como salida el valor de tensión de la próxima media hora en adelante, así tomando acción de conectarse directamente a la máquina expendedora de alimento o mantenerse conectada al banco de baterías. La data resaltada denota la conexión directa a la máquina expendedora de alimentos, en las otras horas se establece la conexión del sistema fotovoltaico al banco de baterías. La conexión y desconexión se indica mediante el *led* indicador del prototipo diseñado.

Una vez diseñado la plataforma de diseño para la red neuronal, se procedió a definir la

Tabla 1: Datos experimentales tomados con el prototipo.

Horas	Voltaje (mV)	Corriente (mA)	Potencia (mW)
07:30 am,	1604	0,160	0,256
08:00 am,	1664	0,166	0,276
08:30 am,	1695	0,169	0,286
09:30 am,	1685	0,168	0,283
10:00 am,	1776	0,177	0,314
11:00 am,	1932	0,193	0,373
11:30 am,	1928	0,193	0,372
12:00 m,	1910	0,191	0,364
1:00 pm,	1898	0,189	0,358
1:30 pm,	1892	0,189	0,357
2:30 pm,	1889	0,188	0,355
3:00 pm,	1888	0,188	0,354
3:30 pm,	1861	0,186	0,346
4:00 pm,	1753	0,175	0,306
6:00 pm,	1651	0,165	0,272

data de entrenamiento (a partir de los datos experimentales), siendo éste un aspecto importante para analizar la respuesta de la red y su capacidad de generalización. Así, se realizó el entrenamiento, que permitió obtener una aproximación de los parámetros de la red, logrando resultados en los que el error converge a cero, es decir, que la salida de la red coincide con el target para la data de validación, se realizó la simulación y se establecieron así las condiciones iniciales para la implementación en hardware, con lo que se programaron las neuronas componentes en el micro-controlador, para las pruebas de funcionamiento. El método empleado para el diseño modular consistió en la descripción generalizada de los tipos de redes (en lenguaje C para su implementación sobre microcontroladores), considerando su arquitectura, características de las capas, función de salida de las neuronas en cada capa y modo de entrenamiento.

#### 4. Aplicaciones neuronales desarrolladas

Se obtiene como resultado la matriz de parámetros de cada una de las redes diseñadas, las cuales han sido insumo para el proceso de programación en el micro-controlador, destacando que este procedimiento puede ser realizado en circuito, solo que por simplificación de la etapa de diseño, se consideró la opción de un pre-

entrenamiento, el cual resulta especialmente útil en las redes multicapa, los parámetros obtenidos como resultados de las simulaciones son presentados a continuación.

##### 4.1. Parámetros del entrenamiento del dispensador de alimentos

Se diseñó una red neuronal lineal, para la clasificación de las especies de fauna del parque que serán abastecidas con la aplicación, seleccionando el esquema de la Figura 4, para múltiples entradas y un número de salidas a seleccionar.

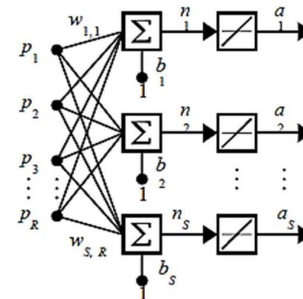


Figura 4: Esquema de la Red Neuronal Lineal.

Un método alternativo corresponde a una red convolucional previa a la red lineal, incorporando el concepto de aprendizaje profundo para clasificación, sin embargo el método de diseño que se ha empleado en esta oportunidad, simplifica la etapa de procesamiento en hardware, aplicando una composición de sensores de entrada, que constituyen el mapa de caracteres (características de identificación).

Para este diseño se consideró una selección de señales de entrada, concatenadas en un vector  $p(x)$ , compuesto por las salidas de los sensores de identificación. En primer lugar pixeles de identificación, sensor de peso, un conjunto de sensores de presencia dispuestos a diferentes alturas, para la identificación de la talla y aspectos de reconocimiento de las mascotas.

En la Figura 5 se presentan los resultados del diseño y parámetros de entrenamiento de la red neuronal lineal (13 entradas 1 salida para accionamiento de los mecanismos del dispensador). Es importante señalar que se seleccionó una salida correspondiente a un comando numérico, para

establecer el accionamiento correspondiente, éste comprende el control de la electro-válvula de suministro de agua y el motor de paso de suministro de alimento, como funciones básicas.

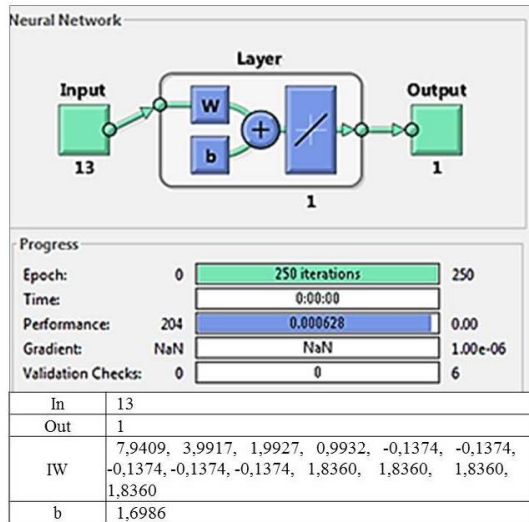


Figura 5: Parámetros de entrenamiento de la RNA Lineal.

#### 4.2. Aplicaciones de la red dinámica

Se diseñaron diversas aplicaciones de redes lineales dinámicas, partiendo del esquema general de la Figura 6.

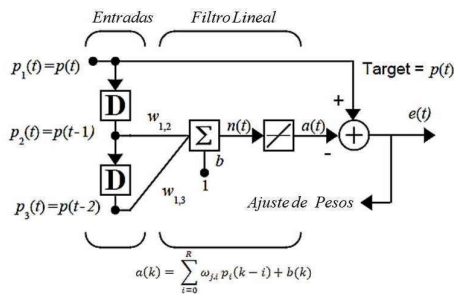


Figura 6: Esquema general de la red lineal dinámica.

Los resultados de los parámetros de entrenamiento obtenidos se presentan en la Tabla 2.

#### 4.3. Parámetros del entrenamiento del control fotovoltaico

Se diseñó una red multicapa, con entradas de los sensores LDR (IL, DL), y una salida que corresponde al accionamiento del servomotor,

Tabla 2: Resultados de entrenamiento red dinámica.

Cargador de baterías con mini-Turbinas	
In	12 (entradas retrasadas)
Out	1
IW	0,9628, -0,0013, -0,0006, -0,0075, 0,0067, -0,0003, -0,0003, -0,0077, 0,0068, -0,0003, -0,0004, -0,0046
b	0.8213
Sistema Inteligente de Gestión de Carga	
In	14 (entradas retrasadas)
Out	1
IW	0,0374, 0,0179,-0,0076, 0,0569, 0,0933, -0,0118, -0,0134, -0,0082, -0,0136, -0,0094, 0,0028, -0,0136, -0,0640, -0,0561
b	16,040
Cargador de baterías con mini-Turbinas	
In	5 (entradas retrasadas)
Out	1
IW	Adaptativo en tiempo real
b	Adaptativo en tiempo real

para posicionamiento del arreglo solar en máxima potencia, según el esquema de la Figura 7.

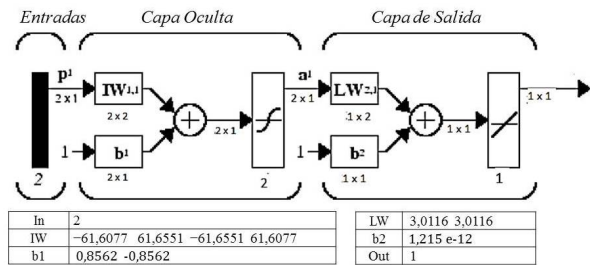


Figura 7: Esquema de la Red Neuronal Multicapa MPL.

Sustituyendo las entradas de los sensores y operando con la matriz de pesos de la capa oculta y la matriz de pesos de la capa de salida respectivamente, se obtiene:

$$a_3(k) = L\omega_{1,1}(\tanh(I\omega_{1,1}D_L(k) + I\omega_{1,2}I_L(k) + b_1(k)) + L\omega_{1,2}(\tanh(I\omega_{2,1}D_L(k) + I\omega_{2,2}I_L(k) + b_2(k))) + b_3(k)$$

Las pruebas de implementación en hardware de las redes neuronales estudiadas, permitió el procesamiento en tiempo real de los patrones adquiridos a través de la conversión analógica-digital, con una conversión de 10 bits por valor analógico en el rango de 0-3,3 V con una precisión

de 3,3 V/1024 bit = 3 mV/bit. Así mismo, se configuró la topología de la red neuronal, para el predictor se programó con función de activación lineal, asignándole 15 entradas, con una línea de retardo TDL de 14 registros, comunicando por hiper-terminal los valores de conversión en tiempo real de la entrada  $p(n)$ , la salida posterior a la función de activación lineal  $a(n)$  y el error  $e(n)$ , procesados en el micro-controlador. Se tomó en consideración para la red del predictor adaptativo, una actualización de pesos continua ejecutándose sin un límite de épocas de entrenamiento. De esta manera, se desarrolló la configuración de la red en el microcontrolador.

Por otra parte, para la implementación de la red multicapa FeedForward, se tiene la estructura de la red programada usando lenguaje C, con dos neuronas en la capa oculta y una neurona en la capa de salida, destacando que el algoritmo de entrenamiento de retro-propagación, el cual se aplicó para definir la matriz de pesos sinápticos de la red neuronal, como se muestra en la Figura 8. Se puede observar en las dos neuronas de la

```
// Capa Oculta
// Neurona 1: a1(k)=tanh(ω1.1.p1(k)+ω1.2.p2(k)+b11(k))
n1f[0][0] += Seg_IW11[0][0]*V_LDR_Izf
+ Seg_IW11[0][1]*V_LDR_Derf + Seg_B1[0][0];
n1f[0][0] = tanh(n1f[0][0]);
// Neurona 2 a2(k)=tanh(ω1.2.p1(k)+ω1.2.p2(k)+b12(k))
n2f[0][0] += Seg_IW11[1][0]*V_LDR_Izf
+ Seg_IW11[1][1]*V_LDR_Derf + Seg_B1[1][0];
n2f[0][0] = tanh(n2f[0][0]);
// Capa de Salida
// Neurona de salida a3(k)=ω21.1(a1(k))+ω21.2(a2(k))+b2(k))
n3f[0][0] += Seg_LW21[0][0]*n1f[0][0]
+ Seg_LW21[0][1]* n2f[0][0] + Seg_B2[0][0];
```

Figura 8: Programación de la red MPL en el PIC.

capa oculta la interacción de las entradas (señales del sensor LDR), para los pesos definidos en el esquema matricial MPL. Cada una de las salidas obtenidas en las neuronas de la capa oculta n1f y n2f (función de salida tangh), pasan a ser las entradas de la neurona n3f (con función de salida lineal) de la cada de salida de la red neuronal.

Seguidamente, se programó un manejador para la comunicación de la red con una pc, la cual permite definir acciones, tomar las lecturas de la red, su respuesta y graficar el comportamiento de

la salida en relación a los patrones de entrada. La interface es presentada en la Figura 9.

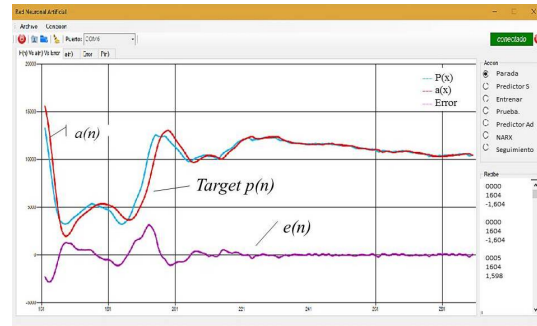


Figura 9: Manejador de entrenamiento y prueba de la red neuronal en el micro-controlador.

En la interface se presentan las funciones asociadas al radio-button, se describen las funciones diseñadas, que están soportadas por la herramienta para la configuración de las redes neuronales de la aplicación, sobre el hardware del micro-controlador. Estas funciones están definidas de la forma, el procesador ejecuta programa del Predictor Simple <A>, actualización de parámetros (pesos) durante el Entrenamiento <E>, Prueba de Secuencia <S>, Predictor Adaptativo <F>, Seguimiento de la red Red FeedForward-FF <B>, Entrenamiento de la red MPL, algoritmo retro-propagación conocido como BackPropagation <W>, Red Neuronal NARX <N>, Parada <Z> Detiene todo tipo de acción.

A través de la interface de realizó el procedimiento de diseño de las redes a estudiar, en primer lugar se tiene la función <Entrenamiento>, en la cual se realiza el cálculo de los parámetros de la red desde el microcontrolador, esto a partir de las entradas recolectadas desde el convertidor A/D, y los targets suministrados al sistema, destacando que en el caso del predictor las entradas están dadas por muestras retrasadas  $p(n-i)$ , y el target viene dado por  $p(n)$ . De esta manera, se realizó el entrenamiento de las redes neuronales implementadas en hardware, según la secuencia indicada: en la Figura 10.

Seguidamente, se diseñó un experimento de actualización de pesos sobre el hardware, donde se modificó la componente [0][10] con un valor arbitrario de 0,1, esto para poder realizar la validación de corrección de pesos cuando el



```
// RNA Lineal
-----
// Se define el coeficiente de aprendizaje de la red: Ap1=0.002;
// Se Actualizan los pesos sinápticos:
// W(n) = W(n-1) + coef*[f(n-a(n)]*p(n)
wf[0][0] = wf[0][0] + (Ap1)*(P_Inf-(af[0][0]))*TDL[0][0];
...
wf[0][13] = wf[0][13] + (Ap1)*(P_Inf-(af[0][0]))*TDL[0][13];
-----
// Red MPL con algoritmo de entrenamiento BackPropagation,
-----
// con función sigm.
// Matriz de pesos de inicialización:
// Se evalúa la salida en la red hacia adelante
net = n1*Seg_LW21[0][0] + n2*Seg_LW21[0][1] + Seg_B2[0][0];
// Se calcula el error asociado a cada neurona
Error = (TargetFeed[0][k]-(net)); // Cálculo del error de la red lineal
// actualización de Parámetros de la capa de salida:
Seg_LW21[0][0] += 0.01*Error*n1;
Seg_LW21[0][1] += 0.01*Error*n2;
Seg_B2[0][0] += 0.01*Error;
// actualización de Parámetros de la capa oculta:
// Relación de Diferencial del error según la función de la neurona
// Sigmoide : a1(k) .(1-a1(k)),
// Tangente Hiperbólica: (1-a1(k)2)
Error_1 = (1-pow(zol,2))*Error*Seg_LW21[0][0];
Error_2 = (1-pow(zol,2))*Error*Seg_LW21[0][1];
Seg_IW11[0][0] += 0.01*Error_1*P_LDR_1[0][k];
Seg_IW11[0][1] += 0.01*Error_1*P_LDR_2[0][k];
Seg_B1[0][0] += 0.01*Error_1;
Seg_IW11[1][0] += 0.01*Error_2*P_LDR_1[0][k];
Seg_IW11[1][1] += 0.01*Error_2*P_LDR_2[0][k];
Seg_B1[1][0] += 0.01*Error_2;
```

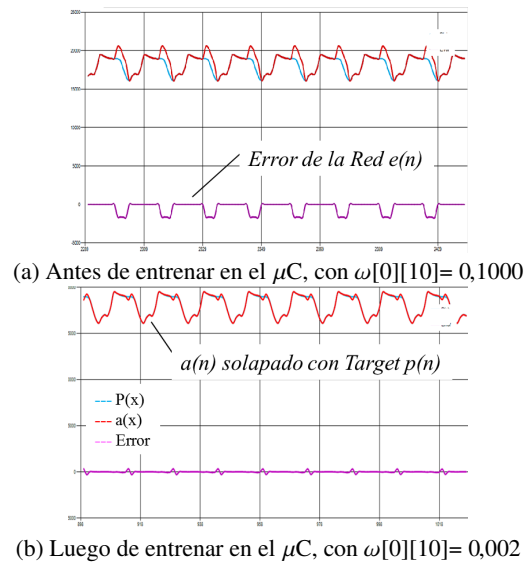


Figura 11: Entrenamiento de parámetros de la red neuronal predictiva.

Figura 10: Código de entrenamiento de las redes neuronales diseñadas.

$\mu C$  ejecuta la rutina de entrenamiento, con lo cual se obtuvo que la red converge de modo estable, luego de 800 épocas de entrenamiento incremental se obtuvo nuevamente el valor original de  $\omega[0][10]=0,0028$ . De esta manera, se comprobó el correcto funcionamiento de las redes neuronales programadas, de forma independiente, funcionando al mismo tiempo. Se analizaron los resultados correspondientes al porcentaje de utilización de los recursos en hardware, el uso de la RAM estática en 1,08 KB y la memoria RAM dinámica en 15,34 KB, la memoria de programa utilizada es 33,40 KB, la cual corresponde el 25 % de la memoria total del dispositivo.

Al seleccionar la opción de *Prueba de secuencia* en la gráfica superior, podemos discernir el efecto que causó la variación introducida a propósito en una componente de la matriz de pesos. En la Figura 11, se presenta el resultado de la salida al modificar intencionalmente uno de los pesos sinápticos, a fin de observar su efecto. La gráfica de color rojo representa la salida de la neurona  $a(n)$ , la gráfica de color azul representa los datos de entrada  $p(n)$ ; los cuales serían datos a predecir, y la gráfica color violeta representa el error entre

el Target (valor esperado) y el valor de salida de la neurona.

Se selecciona la opción entrenamiento en la gráfica superior, comienza el proceso de corrección y adaptación mediante la actualización de pesos de una manera incremental, cada vez que ingresa un patrón de 14 datos se realiza el cálculo de pesos de cada componente de la matriz de pesos y se sustituye por el nuevo valor, al cabo de 67 épocas, se observa en la gráfica superior como la salida  $a(n)$  se va adaptando a los patrones de entrada  $p(n)$ . Al ejecutarse 800 épocas se obtiene una corrección total de la componente errónea en la matriz de pesos, luego de entrenar se logra un error que converge a cero, observándose que se solapan la salida de la red y el target. Este tipo de pruebas pueden ser implementadas por el usuario con una infinidad de variaciones tanto en el vector de entrada, matriz de pesos sinápticos y polarización, esto con la finalidad de afianzar la teoría de redes neuronales de manera práctica. La siguiente prueba consistió en realizar el entrenamiento de la red predictiva para el caso de diferentes coeficientes de aprendizaje y observar el efecto de estabilidad de convergencia de la red neuronal, en la Figura 12, se presentan los resultados del entrenamiento.

Se puede observar (Figura 12) que al seleccionar un coeficiente de aprendizaje mayor al límite

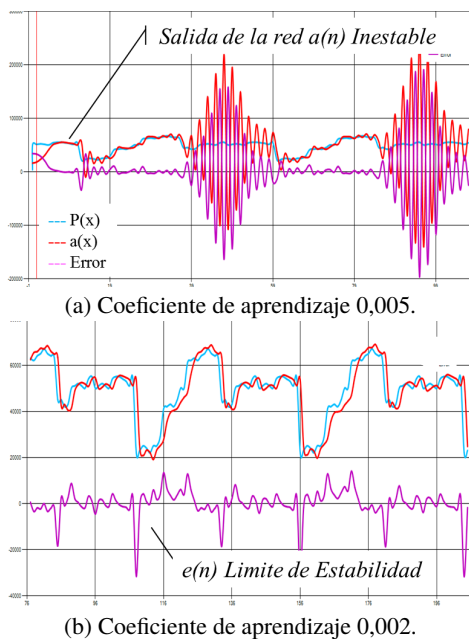


Figura 12: Respuesta de la red adaptativa como predictor.

de estabilidad (a) la respuesta del sistema se vuelve inestable, en tanto que el error oscila sin lograr converger a cero, se seleccionaron diversos coeficientes de aprendizaje obteniendo que el límite de estabilidad se obtiene con 0,002 (b), donde el error converge a cero, por tratarse de un predictor adaptativo al presentarse cambios en la señal de entrada  $p(n)$ , el sistema debe reentrenar sobre el circuito obteniendo los nuevos valores de los parámetros, lo que explica las oscilaciones en los puntos de cambio.

## 5. Conclusiones

Esta investigación aporta una solución en software/hardware, para la configuración y prueba del sistema de control neuronal para aplicaciones de un parque ecológico, de manera modular, que integra diversos elementos de conversión de energía renovable y un sistema inteligente para optimizar su configuración, con análisis de eficiencia energética, a partir de predicción de la demanda eléctrica y el seguimiento óptimo de la celda solar implementada en el prototipo de prueba. Destacando que la propuesta es respetuosa con el medio ambiente, considerando aplicaciones como el módulo inteligente del dispensador de alimento

para mascotas, considerado como la carga principal del sistema, por su importancia. En el diseño se analizó el impacto ambiental, siendo los recursos naturales tratados de manera responsable, como el caso del agua de lluvia para accionar la turbina de conversión de energía, y se utilizó componentes programables para que el proyecto no presente obsolescencia y pueda actualizarse en el tiempo, reduciendo así los residuos electrónicos, propios de los procesos de actualización tecnológica.

Adicionalmente, se implementaron los módulos para el parque ecológico: predictor simple para las celdas fotovoltaicas, sistema de seguimiento de la irradiación máxima en un panel fotovoltaico, el predictor adaptativo conecta o desconecta los bancos de baterías dependiendo del consumo del parque, con esto se cuida la vida útil de las baterías, el módulo de seguimiento solar, en base a la irradiación, mueve el panel fotovoltaico hacia el lado de mayor incidencia, con lo cual se hace más eficiente el sistema de generación, gracias al control neuronal que permite aprender del modelo, desde su implementación en hardware y obtener resultados más eficientes.

A partir del análisis de recursos y eficiencia en épocas de entrenamiento se tiene la posibilidad de escalar el diseño hasta el límite de memoria del dispositivo, lo que hace factible la expansión de los módulos del sistema, siempre tomando en cuenta que se simuló en un procesador secuencial, incidiendo en la velocidad de procesamiento y considerando que la simplificación del diseño incide en la disminución del consumo de energía del modelo de control, quedando planteado para trabajos futuros su implementación sobre procesadores paralelos, a través de tecnología FPGA. Finalmente, el manejador de la red neuronal constituye una herramienta versátil para el modelado, entrenamiento y pruebas de funcionamiento, que permite afianzar conocimientos teóricos de manera didáctica [18], lo cual es un valioso aporte para el eco-parque, que tendrá la capacidad de ofrecer soporte en actividades de formación, investigación y desarrollo. El diseño modular, permite la incorporación de nuevas aplicaciones con implementación de redes neuronales, integrando elementos al sistema sin

alterar los objetivos alcanzados, siendo flexible para proponer alternativas que se adapten a las potencialidades del parque ecológico.

## 6. Referencias

- [1] Cecilia Sandoval Ruiz. Diseño conceptual de un módulo híbrido para generación eléctrica. In *VIII Congreso Nacional y 2do Congreso Internacional de Investigación Universidad de Carabobo*, Venezuela, 2013.
- [2] Cecilia Sandoval Rúiz. Plataforma de gestión, investigación y formación en tecnologías sostenibles para soporte de un laboratorio remoto. *Eduweb*, 10(1):79–92, 2016.
- [3] I. González García. Control neuronal de un generador de inducción para generación eólica. Tesis de Maestría, Instituto Politécnico Nacional, D.F., México, 2005.
- [4] D. Gandolfo, M. Molina, and H. Patiño. Estimación de la energía generada por sistemas fotovoltaicos mediante redes neuronales artificiales. In *Cuarto Congreso Nacional – Tercer Congreso Iberoamericano Hidrógeno y Fuentes Sustentables de Energía – HYFUSEN 2011*, 2011.
- [5] Santiago Fernández Posada. Diseño e implementación de una red neuronal para el seguimiento del punto máximo de poder de un panel solar. Trabajo Especial de Grado, Universidad Militar Nueva Granada, Bogotá, Colombia, 2015.
- [6] CECILIA Sandoval-Ruiz. Sistema eco-adaptativo integrado en elementos arquitectónicos con tecnología sostenible. *Revista Electrónica Científica Perspectiva*, 8(4):96–109, 2015.
- [7] María José Santofimia Romero, Xavier del Toro García, and Juan Carlos López López. Técnicas de inteligencia artificial aplicadas a la red eléctrica inteligente (smart grid). *Novática: Revista de la Asociación de Técnicos de Informática*, (213):29–34, 2011.
- [8] IRENA. Análisis del mercado de energías renovables américa latina. Reporte técnico, IRENA, 2016.
- [9] C Sandoval-Ruiz. Adaptive control in VHDL applied to a solar oven. *Revista Colombiana de Tecnologías de Avanzada*, 1(23):142–147, 2014.
- [10] Luis Gonzaga Baca Ruiz, Manuel Pegalajar Cuéllar, Miguel Delgado Calvo-Flores, and María Del Carmen Pegalajar Jiménez. An application of non-linear autoregressive neural networks to predict energy consumption in public buildings. *Energies*, 9(9):1–21, 2016.
- [11] Marco Antonio Reyes Clavijo and Álvaro Xavier Sánchez Santander. Análisis de factibilidad del proyecto de responsabilidad social “ECO CAN” en la ciudad de cuenca. Trabajo Especial de Grado, Universidad del Azuay, Ecuador, 2017.
- [12] Cecilia Sandoval-Ruiz. Modelo neuro-adaptativo en VHDL, basado en circuitos NLFSR, para control de un sistema inteligente de tecnología sostenible. *Universidad, Ciencia Y Tecnología*, 21(85):140–149, 2017.
- [13] Cecilia Sandoval Ruiz. FPGA prototyping of neuro-adaptive decoder. In *Proceedings of the 9th WSEAS international conference on computational intelligence, man-machine systems and cybernetics*, pages 99–104. World Scientific and Engineering Academy and Society (WSEAS), 2010.
- [14] Cecilia E. Sandoval-Ruiz. VHDL Model of configurable neural networks applied to decoding in cognitive radio. *Revista Ingeniería UC*, 24(3):290–301, 2017.
- [15] Thomas Strasser, Filip Andrén, Felix Lehfuss, Matthias Stifter, and Peter Palensky. Online reconfigurable control software for ieds. *IEEE Transactions on Industrial Informatics*, 9(3):1455–1465, 2013.
- [16] Granados-Rojas Benito, A Jiménez-Saucedo Mario, A Vallejo-Alarcón Manuel, E González-Navarro Yessenia, Villarreal-Cervantes Miguel, and Corona-Ramírez Leonel. Implementación de una red neuronal multicapa en un microcontrolador pic para la identificación de color. In *10<sup>mo</sup> Congreso Nacional de Mecatrónica*, pages 84–89, Puerto Vallarta, Jalisco, México, Noviembre 2011. Asociación Mexicana de Mecatrónica A.C.
- [17] Stuart Russell and Peter Norvig. *Inteligencia artificial: un enfoque moderno*. Prentice Hall Hispanoamericana, 2<sup>da</sup> edition, 1996.
- [18] Cecilia Sandoval Ruiz. Estrategias de aprendizaje colaborativo, a través de las TIC y grupo de investigación. *Revista de Tecnología de Información y Comunicación en Educación*, 11(2):99–112, 2017.