

# Desarrollo e implementación de redes neuronales para acondicionamiento de sensores LM35, SHARP y MG811 utilizando Matlab y PIC16F877A

J.C. Valdez-Muñoz<sup>1</sup>

**Resumen**—En este proyecto se muestra la metodología utilizada así como los resultados obtenidos correspondientes a la implementación de una red neuronal trabajando como respuesta de acondicionamientos para sensores tanto lineales como no lineales a través de un dispositivo microcontrolador y que permita visualizar los datos del sensor para ser aplicados en acciones de control de sistemas digitales. Para acondicionarlos, se sugiere la respuesta de sensores utilizando algoritmos de aprendizaje neuronal artificial, entrenando la red neuronal en el software Matlab, para posteriormente programar el comportamiento aprendido en el microcontrolador. En el desarrollo del proyecto se obtuvieron respuestas de acondicionamiento de tres sensores, el sensor de temperatura LM35, el sensor de distancia Sharp y un sensor de CO2 MG811 de forma precisa y sin utilizar métodos de linealización para el cálculo de los acondicionamientos y permitiendo visualizar los resultados en un display LCD para monitorearlos y desarrollar acciones de control.

**Palabras claves**— acondicionamiento de señales, microcontrolador, sensores, redes neuronales.

**Abstract**— This project shows the methodology used as well as the obtained results corresponding to the implementation of a multilayer perceptron type neuronal net, working as a response to conditioning either for linear and non-linear sensors through a microcontroller and that allows to visualize the sensor data to be applied in digital systems control actions. To prepare them, the response of sensors is suggested using artificial neural learning algorithms, programming and training the neuronal net in Matlab software, subsequently program the learned behavior in the microcontroller PIC16F877A. In the development of the project, preparation responses were obtained from three sensors, the LM35 temperature sensor, the Sharp distance sensor, and a carbon-dioxide MG811 CO2 accurately and without using linearization methods to calculate the conditioning and allow the results to be displayed on an LCD screen to monitor them and develop control actions.

**Keywords**— signal conditioning, microcontroller, sensors, neural networks.

## I. INTRODUCCIÓN

La investigación se desarrolla para cumplir con necesidades del campo de la instrumentación, control y monitoreo de variables físicas en procesos que hacen uso de sensores cuya respuesta permita acondicionarse, se desarrolla una tarjeta adquisidora y de respuesta de variables físicas a través de dispositivos sensores para el monitoreo y acondicionamiento del mismo, haciendo uso de redes neuronales implementadas en dispositivos programables como los microcontroladores. Las redes neuronales se han utilizado para desarrollar sistemas controlados aplicándose al control de riego usando instrumentación y análisis de imágenes para un micro-invernadero aplicado al cultivo de albahaca [1]. Las redes neuronales se han utilizado para el desarrollo de un control de dirección de un robot móvil Zumo 32U4, utilizando una red neuronal artificial B-Spline [2].

Los alcances que tienen las redes neuronales artificiales en la implementación en algún proceso o control son amplios, y podrán permitir aproximar el valor de un sensor en función de un parámetro de aprendizaje. En cuanto a su implementación en microcontroladores, se han implementado redes neuronales multicapa en microcontroladores PIC para la identificación de colores [3]. El presente artículo cumple con el objetivo de otorgar una opción más de procesos de acondicionamiento y monitoreo de sensores, brindando un prototipo de tarjeta adquisidora de datos, la cual otorga respuestas de acondicionamiento haciendo uso de algoritmos de redes neuronales.

## II. PLANTEAMIENTO DEL PROBLEMA Y ASUNCIONES

Los sistemas de acondicionamiento implementados actualmente en procesos de ingeniería de control y automatización, requieren de componentes físicos que puedan ejecutar las tareas necesarias, se emplean desde instrumentos, medidores, sensores y circuitos electrónicos los cuales se diseñan para poder aplicarse en dichos sistemas. En bastantes de los casos, los sensores utilizados para el control de un lazo, son sensores cuya respuesta no es lineal en función de la medición que realizan.

<sup>1</sup> Tecnológico Nacional de México, Instituto Tecnológico Superior de Lerdo, Av. Tecnológico S/N, Col. Periférico, C.P. 35150, Cd. Lerdo Durango, México.  
[juancarlos99mx@gmail.com](mailto:juancarlos99mx@gmail.com)

La posibilidad de realizar acondicionamientos para respuestas tanto lineales como no lineales a través de un dispositivo microcontrolador y que a su vez permita una fácil implementación para el sensor facilitaría este proceso. Para llegar a ello se presentara la respuesta a sistemas de acondicionamiento de sensores utilizando algoritmos de aprendizaje neuronales artificiales. La finalidad de utilizar la red neuronal nos ayudara a determinar si esta aplicación es posible por medio de otras opciones y todo desde una misma tarjeta de desarrollo.

III. PARTE TÉCNICA DEL ARTÍCULO

A. Desarrollo de la red neuronal en software Matlab

El desarrollo de una red neuronal artificial consta de un algoritmo, el cual se describe en base ecuaciones matemáticas, en (1) se muestra la ecuación matemática para el perceptrón multicapa, esta ecuación corresponde a una sola capa, por lo que de tenerse más capas se deberán realizar en igual cantidad la misma ecuación.

$$y_k = f \left( \sum_{i=1}^n (X_i * W_i) - \theta \right) \tag{1}$$

Donde:  $y_k$  es la salida de la red neuronal,  $f$  es la función de activación,  $n$  es la cantidad de neuronas,  $X_i$  es la entrada de la  $i$ -ésima neurona,  $i$  es la neurona seleccionada,  $W_i$  es el peso sináptico de la  $i$ -ésima neurona y  $\theta$  es el umbral de la capa.

La red neuronal desarrollada está programada para dos capas, por lo que la ecuación (1), queda expresada como se muestra en (2) y (3).

$$y_1 = f(W_1 * X_i + \theta_1) \tag{2}$$

$$y_2 = f(W_2 * y_1 + \theta_2) \tag{3}$$

Inicialmente, los valores numéricos de  $W_n$  y de  $\theta_n$  deberán ser un valor entre -2 y 2 para asegurar una convergencia más rápida. El valor de  $W$  y de  $\theta$  para todas las capas implementadas deberá recalcularse por cada iteración de aprendizaje, siguiendo las ecuaciones de la (4)-(7).

$$s_2 = -2(1 - y_2[i]^2) * e \tag{4}$$

$$s_1 = MD(1 - y_1[i]^2) * W^t * s_2 \tag{5}$$

$$W_{new} = W_{old} - a * s_i * y_i^t \tag{6}$$

$$\theta_{new} = \theta_{old} - a * s_i \tag{7}$$

Donde:  $a$  es el alfa (entre 0.01 y 0.65),  $s_i$  es la sensibilidad de la  $i$ -ésima capa,  $e$  es el error de la salida  $y_n$ -la salida deseada de aprendizaje y, MD es la matriz diagonal de los valores indicados [4].

Todo el desarrollo tanto en Matlab como en el microcontrolador se desarrolla en base a la **¡Error! No se encuentra el origen de la referencia..**

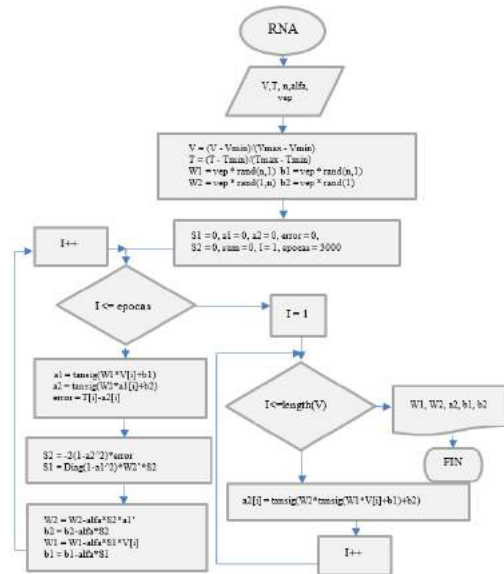


Figura 1. Diagrama de flujo de una RNA perceptrón multicapa de 2 capas.

Para el desarrollo de la red neuronal se declaran todas las variables vistas de las ecuaciones anteriores, y desarrollar un algoritmo que cumpla con dicha estructura. La función de activación depende de los tipos de datos a manejar por la red neuronal, en la Tabla I se muestran los tipos de funciones de activación existentes.

TABLA I  
FUNCIONES DE ACTIVACIÓN

Función	Formula	Rango
Identidad	$y = x$	$[-\infty, \infty]$
Escalón	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$	$[0,1]$
	$y = \begin{cases} +1 & \text{si } x \geq 0 \\ -1 & \text{si } x < 0 \end{cases}$	$[-1,1]$
Lineal a tramos	$y = \begin{cases} x & \text{si } -1 \leq x \leq 1 \\ +1 & \text{si } x > 1 \\ -1 & \text{si } x < -1 \end{cases}$	$[-1,1]$
Sigmoidea tangencial	$y = \frac{1}{1 + e^{-x}}$	$[0,1]$
	$y = \tanh(x), y = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$[-1,1]$ ó $[0,1]$
Sinusoidal	$y = \sin(\omega x + \varphi)$	$[-1,1]$

B. Identificación de sensores y entrenamiento

Para el desarrollo del proyecto, se decide utilizar un sensor de comportamiento lineal, y dos sensores de comportamiento no lineal, el de temperatura LM35, de distancia Sharp GP2Y0A21 y el sensor de CO2 MG811. Para desarrollar el aprendizaje de la red neuronal en función a estos sensores, solo se debe disponer de su hoja de datos técnicos, conocer

sus rangos de operación y el acondicionamiento necesario o existente en ellos.

El sensor de temperatura LM35, es un sensor que entrega una tensión de 10mV por °C. Por lo que la red neuronal es entrenada con los siguientes valores, **¡Error! No se encuentra el origen de la referencia.** muestra los datos de entrada, y **¡Error! No se encuentra el origen de la referencia.** muestra los datos de salida objetivo.

$$V[11] = [0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5] \quad (8)$$

$$T[11] = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100] \quad (9)$$

Para el sensor de distancia, es necesario basarse en la **¡Error! No se encuentra el origen de la referencia.**, la cual muestra los datos para poder obtener los valores de aprendizaje.

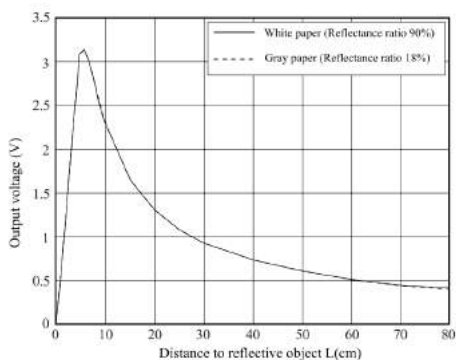


Figura 2. Curva de respuesta del sensor de distancia Sharp. La red neuronal es entrenada con los siguientes valores, **¡Error! No se encuentra el origen de la referencia.** muestra los datos de entrada y **¡Error! No se encuentra el origen de la referencia.** los datos de salida.

$$V[15] = [0 \ 0.25 \ 0.5 \ 0.75 \ 1 \ 1.25 \ 1.5 \ 1.75 \ 2 \ 2.25 \ 2.5 \ 2.75 \ 3 \ 3.15 \ 3.3] \quad (10)$$

$$D[15] = [85 \ 82 \ 61 \ 40 \ 27.5 \ 22 \ 17 \ 14 \ 12 \ 10 \ 9 \ 8 \ 6 \ 5 \ 5] \quad (11)$$

Para el sensor de CO2 MG811, es necesario basarse en la Figura 3, la cual muestra su comportamiento de medición en partes por millón en escala logarítmica.

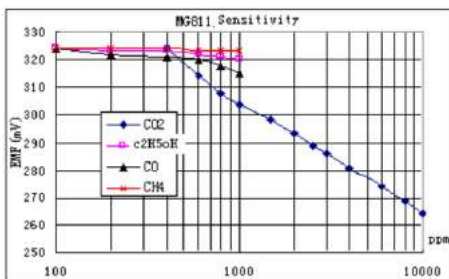


Figura 3. Respuesta del sensor de CO2 MG811.

C. Obtención y comprobación de pesos y umbrales

Se obtiene la comprobación visualizando gráficamente la respuesta de la red neuronal con una gráfica de aprendizajes deseados, este punto requiere de análisis y comprensión del comportamiento de la gráfica, desarrolla un código en Matlab para su visualización.

El error en todo momento llega a 0, cuando esto no sucede, se repite el entrenamiento, en el entorno de Matlab se obtuvieron los valores numéricos para cada sensor, a su vez, esto se debe hacer repetidas veces, para determinar cuál entrenamiento fue mejor e implementarlo en el microcontrolador. Las variables que determinan los pesos y umbrales son  $W_1, W_2, b_1$  y  $b_2$ . Por ello, se debe graficar y una vez comprobado, se registran esos valores, en la Figura 4, se muestra un ejemplo de los resultados obtenidos por la red neuronal.



Figura 4. Respuesta de la red neuronal para el sensor de CO2 MG811.

D. Implementación en microcontrolador y simulaciones

Para la implementación de la red neuronal en el microcontrolador, se programa el mismo código de Matlab adecuado para la lógica de programación en C. Los pesos y umbrales  $W$  y  $b$  están definidos gracias a los resultados otorgados por el software Matlab, en la Figura 5, se muestra el circuito electrónico diseñado para las simulaciones de la respuesta de la red neuronal en el microcontrolador PIC16F877A.

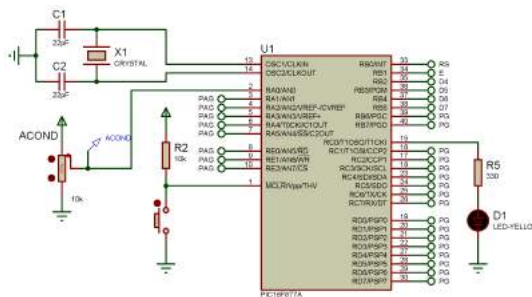


Figura 5. Diseño esquemático para simulación de RNA en PIC16F877A.

PG: Pin de propósito general, PAG: pin analógico de propósito general.

E. Implementación física del microcontrolador

En la Figura 6, se muestra el diseño esquemático para el desarrollo de la tarjeta encargada de comunicar el sistema de acondicionamiento con el microcontrolador, la RNA programada en él, toma este dato analógico y lo propaga a través de sus capas neuronales, finalmente el resultado es codificado para poder mostrarlo en un display LCD.

El circuito mostrado, es compatible con el PIC16F877A y el PIC18F4550. Además de ser compatible con comunicaciones para el protocolo USB.

El diseño esta adecuado para que se pueda disponer de todos aquellos puertos no utilizados, además de poder prescindir del LCD como comunicaciones utilizando el puerto B, y así enviar los datos por protocolo de Ethernet, utilizando el integrado ENC28J60.

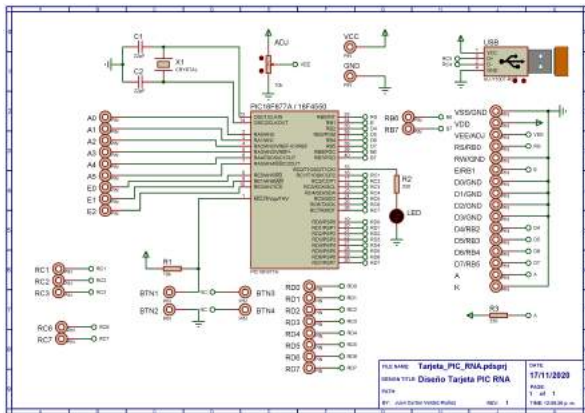


Figura 6. Diseño esquemático de la tarjeta para el microcontrolador.

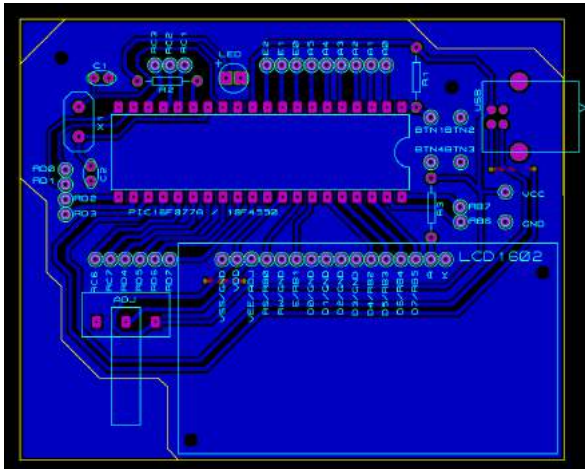


Figura 7. Diseño PCB de la tarjeta para el PIC16F877A.

La Figura 7, muestra el diseño PCB del esquema de la Figura 6, el cual corresponde al diseño de la tarjeta para montar el microcontrolador.

IV. RESULTADOS

En la Figura 8, se muestra la gráfica de aprendizaje de la red neuronal utilizando Matlab, la cual corresponde al sensor LM35.

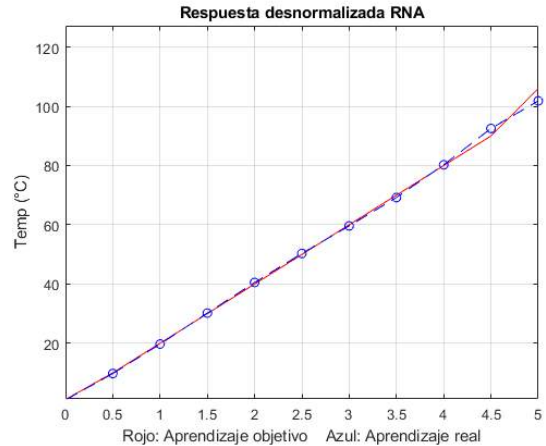


Figura 8. Curva de aprendizaje del sensor LM35 de 0 a 100 ° C.

La Figura 9, muestra la curva de aprendizaje para un sensor de distancia Sharp GP2Y0A21YK0F.

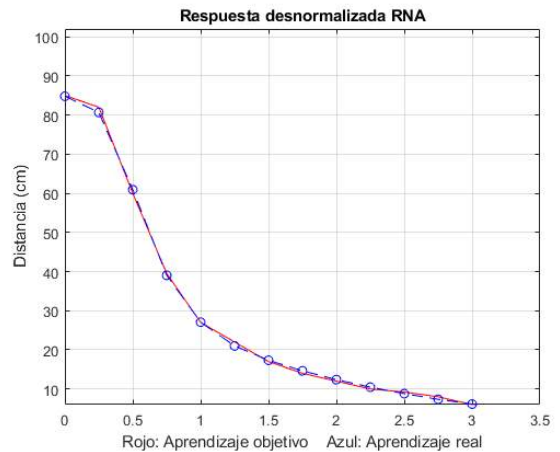


Figura 9. Curva de aprendizaje del sensor de distancia Sharp desde 5 cm hasta 85 cm.

La Figura 10, muestra la curva de aprendizaje para un sensor de concentración en partes por millón de CO2 MG811.



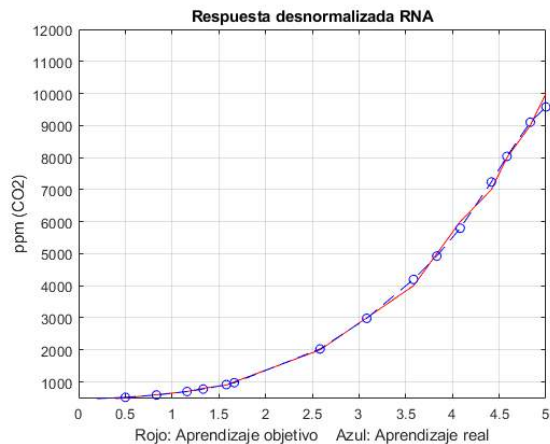


Figura 10. Curva de aprendizaje del sensor de CO2 en ppm MG811.

La Tabla II, registra los pesos y umbrales registrados para el sensor LM35 obtenidos a través del software Matlab.

TABLA II  
PESOS Y UMBRALES DEL SENSOR LM35

n	$W_1$	$W_2$	$b_1$	$b_2$
1	[2.29; 1.1241]	[1.96, 1.17]	[-2.59; 0.55]	1.35
2	[-2.8; -1.4]	[-2.12, -0.95]	[3.25; -0.37]	1.7662
3	[-1.201; -3.49]	[-0.84, -2.05]	[0.078; 4]	2.0965
4	[1.6; 3.6]	[0.62, 2.18]	[-0.38; -4.11]	2.4006
5	[3.48; -1.26]	[2.11, -0.79]	[-3.98; 0.16]	2.2252

La Tabla III, registra los pesos y umbrales registrados para el sensor Sharp GP2Y0A21 obtenidos a través del software Matlab.

TABLA III  
PESOS Y UMBRALES DEL SENSOR SHARP

n	$W_1$	$W_2$	$b_1$	$b_2$
1	[7.20; -0.726]	[-2.89, 0.192]	[0.133; 0.387]	2.939
2	[-0.99; -6.68]	[0.108, -3.278]	[0.168; -0.248]	3.351
3	[-1.201; -3.49]	[2.991, 0.158]	[-0.169; 0.256]	3.055
4	[-6.97; -0.769]	[-1.72, 3.15]	[-1.87; -0.262]	1.555
5	[7.21; 2.4973]	[-2.816, -0.08]	[0.138; -0.877]	2.892

La Tabla IV, registra los pesos y umbrales registrados para el sensor de CO2 MG811 obtenidos a través del software Matlab.

TABLA IV  
PESOS Y UMBRALES DEL SENSOR MG811

n	$W_1$	$W_2$	$b_1$	$b_2$
1	[3.203; -6.027]	[0.325, -3.192]	[-2.15; 6.699]	3.511
2	[5.425; -2.505]	[2.89, -0.339]	[-6.03; 1.688]	3.188
3	[2.731; -5.6]	[0.331, -2.959]	[-6.107; 1.743]	3.262
4	[5.489; -2.592]	[2.917, -0.335]	[-1.87; -0.262]	3.216
5	[3; 5.841]	[0.328, 3.085]	[-2.018; -6.49]	3.397

En la Figura 11, se muestra la elaboración en físico del diseño en PCB, con un panel LCD montado.

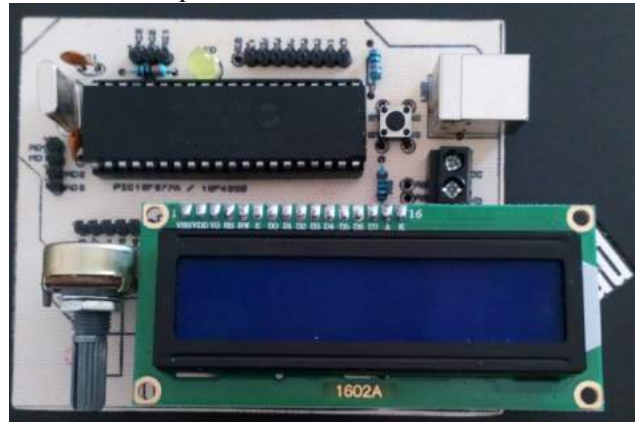


Figura 11. Montaje final del diseño.

En la Figura 12, se muestra la tarjeta en físico, mostrando el acondicionamiento otorgado por la RNA para el sensor LM35.



Figura 12. RNA mostrando el acondicionamiento del LM35.

En la Figura 13, se muestra la tarjeta con el acondicionamiento otorgado por la RNA para el sensor de distancia Sharp.



Figura 13. RNA mostrando el acondicionamiento Sharp.

Finalmente, la Figura 14 muestra el del MG811.



Figura 14. RNA mostrando el acondicionamiento del MG811.

## V. DISCUSIÓN, CONCLUSIÓN Y RECOMENDACIONES

La red neuronal implementada en el software Matlab, responde a los parámetros de aprendizaje de forma correcta y con una muy buena precisión, sin embargo, requiere de altos niveles de procesamiento en hardware y repetidas iteraciones de aprendizaje, con un promedio de 1000 iteraciones.

Los valores numéricos de los pesos y umbrales del software Matlab, presentan una muy buena precisión, sin embargo, al exportarlos a un lenguaje para microcontroladores, en este se pierde precisión en los cálculos, por lo que los resultados vistos gráficamente en Matlab, varían un poco a los resultados matemáticos otorgados por la misma red neuronal en el microcontrolador.

La red neuronal implementada en el microcontrolador, otorga respuestas más rápidas a los valores ingresados en el ADC, sin embargo, se sugiere que la RNA se utilice solo cuando se tenga que acondicionar sensores no lineales. Aun así, se demostró que la RNA responde correctamente a sensores lineales.

La red neuronal no es capaz de llegar de forma exacta a los valores límites de aprendizaje, debido a que la función sigmoide tangencial no puede llegar a límites establecidos.

La red neuronal implementada, fue de tan solo 2 neuronas, obteniendo excelentes resultados, sin embargo, de requerir más neuronas, el microcontrolador está limitado a soportar solamente 8 neuronas, esto se debe a que la memoria del microcontrolador termina llenándose durante el procesamiento de la señal.

La red neuronal es capaz de predecir todos aquellos valores a los cuales no fue entrenada, con tan solo 10 puntos de aprendizaje, la red neuronal es capaz de estimar más de 1000 puntos, para una resolución del ADC de 10 bits.

Se recomienda utilizar únicamente sensores no lineales, cuya respuesta pueda acondicionarse, puede entrenarse cualquier sensor siempre y cuando el acondicionamiento de este otorgue niveles en tensión entre 0V y 5V.

Como trabajos a futuro, se pretende mejorar el algoritmo, para que este sea capaz de dar respuesta a múltiples sensores a la vez, y a su vez pueda desarrollar acciones de control utilizando una interfaz Ethernet con el protocolo TCP/IP. Posterior a esto, se investigara para llevar a la red neuronal desarrollada a una librería útil para microcontroladores.

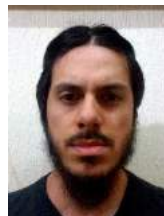
## VI. AGRADECIMIENTOS

Agradecimiento a mis padres, el Sr. Juan Lorenzo Valdez Galván, y la Sra. María Magdalena Muñoz Rosales, por los apoyos brindados, tanto de afecto como económicos, para el desarrollo de este proyecto en el transcurso de mis estudios de ingeniería.

## VII. REFERENCIAS

- [1] M. G. V. Rueda, M. I. Reyes, F. G. F. García y H. A. Casillas, «Redes neuronales aplicadas al control de riego usando instrumentación y analisis de imagenes para un micro invernadero aplicado al cultivo de albahaca.» 2018. [En línea]. Available: [https://www.rcs.cic.ipn.mx/2018\\_147\\_5/Redes%20neuronales%20aplicadas%20al%20control%20de%20riego%20usando%20instrumentacion%20y%20 analisis%20de%20imagenes.pdf](https://www.rcs.cic.ipn.mx/2018_147_5/Redes%20neuronales%20aplicadas%20al%20control%20de%20riego%20usando%20instrumentacion%20y%20 analisis%20de%20imagenes.pdf). [Último acceso: 17 Septiembre 2020].
- [2] M. Padilla Baeza, «Implementación de una red neuronal artificial en un microcontrolador de 8 bits para control de dirección de un robot móvil.» Agosto 2018. [En línea]. Available: [http://isisbi.uqroo.mx/bitstream/handle/20.500.12249/1611/QA76\\_87.2018-1611.pdf?sequence=3&isAllowed=y](http://isisbi.uqroo.mx/bitstream/handle/20.500.12249/1611/QA76_87.2018-1611.pdf?sequence=3&isAllowed=y). [Último acceso: 12 Noviembre 2020].
- [3] B. Granados Rojas, M. A. Jimenez Saucedo, M. A. Vallejo Alarcón, Y. E. Gonzales Navaro, M. Villareal Cervantes y L. Corona Ramirez, «Implementación de una Red Neuronal Multicapa en un microcontrolador PIC para la identificación de color.» 4 Noviembre 2011. [En línea]. Available: <http://www.mecamex.net/anterior/cong10/trabajos/art19.pdf>. [Último acceso: 12 Noviembre 2020].
- [4] A. Zilochian y M. Jamshidi, Intelligent Control Systems using soft computing methodologies, Boca Raton: CRC Press, 2001.

## VIII. BIOGRAFÍA



**Valdez Muñoz, Juan Carlos.** Nació el 18 de marzo de 1999 en la ciudad de Torreón, Coahuila, México. Inicio sus estudios de Ingeniería en Electrónica en el año 2017 en el Instituto Tecnológico Superior de Lerdo. Actualmente estudiante de octavo semestre de la carrera mencionada. Sus áreas de interés son relacionadas al área de instrumentación, control digital, programación y desarrollo de interfaces e inteligencia artificial.