

Metodología para determinar la topología, patrones de parámetros eléctricos y entrenamiento de una Red Neuronal Artificial para un control inteligente.

Methodology to determine the topology, electrical parameters patterns, and training of an Artificial Neural Network for intelligent control.

José Ángel Zepeda Hernández (1).
Instituto Tecnológico de Tuxtla Gutiérrez.
jzpedah@ittg.edu.mx.

Aldo Esteban Aguilar Castillejos (2), I. T. de Tuxtla Gutiérrez, aguilarcastillejos@gmail.com.

Álvaro Hernández Sol (3), I. T. de Tuxtla Gutiérrez, obrindis@gmail.com.

María Catalina Salgado Gutiérrez (4), I. T. de Tuxtla Gutiérrez, ing_katy@hotmail.com.

Artículo recibido en junio 23, 2015; aceptado en noviembre 05, 2015.

Resumen.

El presente trabajo describe la metodología para determinar la topología y la implementación de una Red Neuronal Artificial tipo FeedForward, basada en reglas que se pueden utilizar como punto de partida en la construcción de la red. Se aborda también el proceso para la obtención de patrones de entrenamiento de parámetros eléctricos y su acondicionamiento para la determinación de los pesos de las interrelaciones de las neuronas y bias de ésta, a través de un proceso de simulación realizado en Simulink de MatLab. Dicha actividad es crítica ya que de ésta depende la capacidad de la RNA para responder a los procesos de control inteligente donde será implementada. Se presenta el caso específico de un control neuronal de un Restaurador Dinámico de Voltaje como ejemplo de aplicación de la metodología.

Palabras clave: Control inteligente, Control Neuronal, Patrones de Entrenamiento, Topología de una RNA, Neurona, Red Neuronal Artificial, Parámetros Eléctricos, Restaurador Dinámico de Voltaje.

Abstract.

This paper describes the methodology for determining the topology and implementation of an artificial neural network type FeedForward, based on rules that can be used as a starting point in building the network. Also it addresses the process for obtaining training patterns of electrical parameters and conditioning for determining the weights of the interrelationships of neurons and this bias, through a simulation process performed in Matlab Simulink. This activity is critical because this depends on the ability of the RNA to respond to intelligent control processes, which will be implemented. The specific case of a neural control of a Dynamic Voltage Restorer as an example of application of the methodology is presented.

Keywords: Intelligent Control, Neural Control, Training Patterns, Topology of RNA, Neuron, Artificial Neural Network, Electrical Parameters, Dynamic Voltage Restorer.

1. Introducción

Una Red Neuronal Artificial (RNA) tiene su inspiración en la organización de las neuronas del cerebro humano y pretende emular el proceso de aprendizaje y generalización que éste tiene. Las redes neuronales artificiales presentan un gran número de características similares a las del cerebro: son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, alta tolerancia a fallos, operación en tiempo real, entre otras mencionadas en (Hagan y Demut, 1996). Esto hace que ofrezcan numerosas ventajas concebidas para el control de sistemas complejos no lineales.

1.1. Arquitectura de la Red Neuronal Artificial.

Se denomina arquitectura a la topología, estructura o patrón de conexionado de una red neuronal. En una red neuronal artificial los nodos se conectan por medio de enlaces llamados sinapsis, estructura de conexiones que determinan el comportamiento de la red. Estas conexiones sinápticas son direccionales y la información solamente puede propagarse en un único sentido (desde la neurona pre-sináptica a la pos-sináptica). Las neuronas se agrupan en unidades estructurales denominadas capas. El conjunto de una o más capas constituye la red neuronal de acuerdo a (Hagan y Demut, 1996).

Existen tres tipos de capas en una red, de entrada, de salida y ocultas. Una capa de entrada, también denominada sensorial, está compuesta por neuronas que reciben datos o señales procedentes del entorno. Una capa de salida se compone de neuronas que proporcionan la respuesta de la red neuronal. Una capa oculta no tiene una conexión directa con el entorno, es decir, no se conecta directamente ni a dispositivos sensores ni de actuación. Este tipo de capa oculta proporciona grados de libertad a la red neuronal gracias a los cuales es capaz de representar más fehacientemente determinadas características del entorno que trata de modelar. Véase la Figura 1.

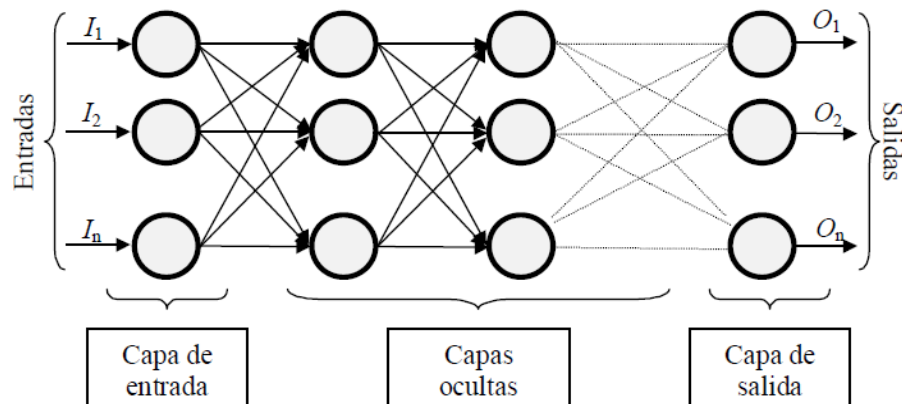


Figura 1. Red neuronal artificial.

1.2. Pesos y *bias* de una RNA tipo *FeedForward*.

Las estructuras de conexiones sinápticas de la RNA tienen un grado de relación entre ellas denominado peso y las neuronas con un valor llamado *bias* el cual proporciona el umbral que permite a la neurona en conjunto con el peso determinar la salida de ésta, los valores de los pesos y *bias* se obtienen como resultado de un entrenamiento a través de patrones que contienen ejemplos de cómo deberá responder la RNA a los datos proporcionados, por tanto es indispensable que estos patrones contengan la suficiente información o datos, que permitan a la Red utilizar su capacidad de generalización (interpolación y extrapolación) para representar de manera adecuada la función requerida. La cantidad de datos a elegir para integrar un patrón es un factor importante a considerar para llegar al resultado esperado, ya que una gran cantidad de datos sobreentrenará a la red impidiendo que se ajuste a condiciones

no establecidas en su operación, por el contrario una cantidad escasa de información no será suficiente para arrojar resultados adecuados.

2. Métodos.

Uno de los principales problemas de las redes multicapas, es la definición de su arquitectura, que incluye las funciones de transferencia y salida a utilizar y la topología de la red. Para garantizar el aprendizaje, las funciones de transferencia y de salida deben ser continuas y derivables en todos los puntos, lo que lleva a la selección en la mayoría de los casos a funciones de transferencia o de activación *sigmoidales* y funciones de salida lineales.

En cuanto a la topología de la Red, la determinación del número de capas que la integran y el número de neuronas ocultas a incluir en cada capa resulta una tarea mucho más compleja, que incide directamente en la capacidad de generalización del modelo. Dado que toda red neuronal dispone necesariamente de una capa de entrada y una capa de neuronas de salida, que recibe los estímulos externos y los entrega al exterior respectivamente, el problema se limita a establecer el número y extensión de las capas ocultas.

Se ha demostrado que la propiedad de aproximador universal de funciones de la red multicapa, requiere un máximo de dos capas ocultas, en la mayoría de los casos una única capa oculta resulta suficiente para conseguir óptimos resultados. (Lippmann, 1987) considera que redes con una única capa oculta resultan suficientes para resolver problemas potencialmente complejos, siempre y cuando la capa oculta incluya al menos tres veces el número de nodos de la entrada.

Por su parte, (Hecht-Nielsen, 1990) y (Lippmann, 1987), aplican una extensión del teorema de Kolmogorov para demostrar que una red de capa simple oculta integrada por $(2N+1)$ neuronas y con funciones de transferencia continua, no lineales y monótonamente crecientes, resulta suficiente para computar cualquier función continua de N variables de entrada.

Las redes neuronales más pequeñas son preferibles a las más grandes que realizan una misma tarea, por varias razones: tienen un menor número de parámetros, el entrenamiento es más rápido y suelen tener una mayor capacidad de generalización al utilizar nuevos patrones. Actualmente no hay un procedimiento sencillo ni general para determinar el tamaño óptimo de una red neuronal, sin embargo existen algunas reglas o procedimientos que se pueden utilizar como base o inicio en la construcción de la red neuronal.

Para determinar el número de neuronas ocultas de cada capa suelen utilizarse las reglas ad-hoc y los procedimientos de poda entre otros que, aunque no todas resultan matemáticamente justificables, han demostrado un buen comportamiento en diversas aplicaciones prácticas.

2.1 Reglas ad hoc.

Entre las reglas ad hoc más conocidas destacan:

a) *Regla de la pirámide geométrica.*

Se basa en la suposición de que la cantidad de neuronas de la capa oculta ha de ser inferior al total de variables de entrada pero superior a las variables de salida. Se considera que el número de neuronas en cada capa siguen una progresión geométrica, tal que, para una red con una capa única oculta, el total de neuronas intermedias debe ser próximo a $\sqrt{n * m}$ donde n = número de variables de entrada y m = número de neuronas de salida (Masters, 1993).

b) *Regla de la capa oculta-capas entrada.*

Según esta regla, el número de neuronas está relacionado con el número de neuronas de entrada. En particular, suele aplicarse la regla 2x1, de forma que el número de neuronas ocultas no puede ser superior al doble del número de variables de entrada. (Swingler, 1996); (Berry y Linoff, 1997).

c) *Regla de la capa oculta-numero de patrones.*

Esta regla relaciona el número de neuronas de la capa oculta con el total de patrones presentados al sistema, siendo habitual la relación 1/30 (una neurona oculta por cada 30 patrones).

2.2 Procedimientos de poda.

- a) Partir de una red neuronal de gran tamaño y *podarla* eliminándole unidades de proceso y conexiones hasta conseguir un tamaño satisfactorio.
- b) Comenzar con una red neuronal muy pequeña e ir incrementando su tamaño añadiendo unidades de proceso, conexiones o capas hasta conseguir un tamaño satisfactorio.
- c) Partir de una red de tamaño suficiente y podar las conexiones y unidades de proceso que se consideren poco relevantes. A continuación se añaden nuevas unidades de proceso con pesos aleatorios y se vuelve a entrenar la red. Este proceso se continúa hasta que se consigue un tamaño aceptable y un comportamiento satisfactorio.

El método OBD (*Optimal Brain Damage*), de lesión cerebral óptima, propuesto por (Lecun, Denker y Solla, 1990), trata de identificar aquellos pesos sinápticos que pueden ser podados examinando las derivadas segundas de la función de error contenidas en la matriz Hessiana. La variación que produce en el error cuadrático medio una pequeña perturbación Δw_{ij} en el peso w_{ij} se aproxima por la ecuación 1, que considera los dos primeros términos del desarrollo de E en series de Taylor, mientras que el algoritmo de retropropagación (*backpropagation*) sólo considera el primer término.

$$\Delta E = \frac{\partial E}{\partial w_{ij}} \Delta w_{ij} + \frac{1}{2} \left(\frac{\partial^2 E}{\partial w_{ij}^2} \right) (\Delta w_{ij})^2 \quad (1)$$

Cuando la red finaliza el entrenamiento en un mínimo local de E , entonces $\frac{\partial E}{\partial w_{ij}} \approx 0$, y así podemos definir a ΔE en la ecuación 2

$$\Delta E \approx \frac{1}{2} \left(\frac{\partial^2 E}{\partial w_{ij}^2} \right) (\Delta w_{ij})^2 \quad (2)$$

Como la poda de una conexión supone pasar su peso sináptico del valor w_{ij} al valor 0, es decir, ocurre cuando $\Delta w_{ij} = -w_{ij}$ entonces la condición para realizar la poda de dicha conexión es que el cambio en el error resultantes sea insignificante, es decir, que la cantidad ΔE sea suficientemente pequeña.

3. Desarrollo.

En este trabajo se describe la metodología para la Construcción de una Red Neuronal Artificial y la obtención de patrones de parámetros eléctricos, tales como: voltajes, corrientes, frecuencias, entre otros que describen el funcionamiento de un sistema de control e implementarlo con un control neuronal, debido a la capacidad de estos sistemas en funcionar de manera adecuada con cargas lineales y no lineales.

El sistema a implementar es un control neuronal de un Restaurador Dinámico de Voltaje (RDV) y la configuración básica de éste se muestra en la figura 2, esta configuración citada es usada en otras investigaciones. (Rosli y Rahim, 2009).

El RDV es usado para mitigar los disturbios de voltaje tipo *SAG* y *SWELL* que se presentan en los sistemas de distribución, debido a la incorporación masiva de convertidores electrónicos implementados en las fuentes de alimentación primarias de equipos de uso cotidiano, condiciones de conexión y desconexión de altas cargas eléctricas no lineales, así como disturbios y fallas en líneas de transmisión y redes de distribución. Esta mitigación se logra compensando de manera vectorial el voltaje aplicado a la carga que el RDV alimenta, dicha compensación es aplicada en serie con la línea de suministro de voltaje principal. (El-Gammal, Abou-Ghazala y El-Shennawy, 2011).

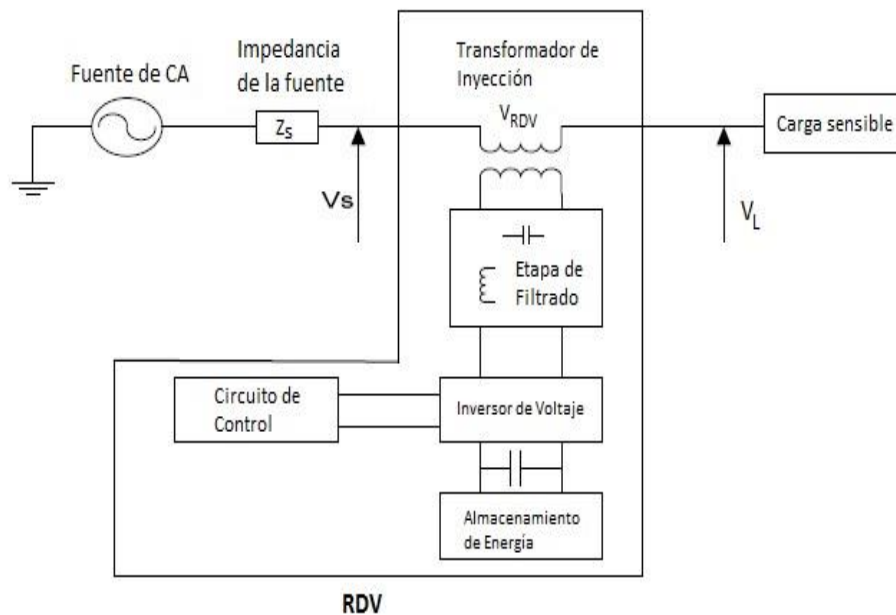


Figura 2. Configuración básica del restaurador dinámico de voltaje.

Una vez que se ha determinado el tipo de sistema a implementar, en este caso el circuito de control mostrado en la figura 2 se sustituye por un control basado en una Red Neuronal. Hasta este momento solo se conoce el número de entradas y salidas de la red tal como se muestra en la figura 3.

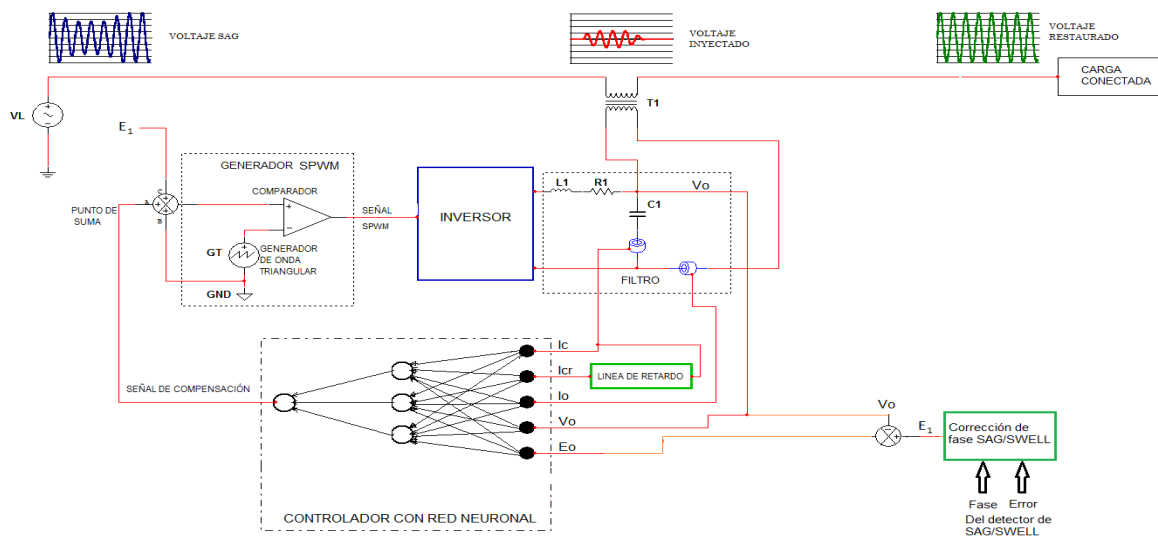


Figura 3. Control neuronal en un restaurador dinámico de voltaje.

Se continúa el proceso definiendo el tipo de arquitectura de la RNA (Red Neuronal Artificial), tomando en consideración el tipo de datos, en este caso variables continuas, y la complejidad de estos, por lo que se propone la utilización de una red tipo hacia adelante (*FeedForward*) multicapa, con un algoritmo de aprendizaje de tipo retropropagación (*backpropagation*).

Se construye la red aplicando los criterios de la reglas *ad-hoc* y *procedimientos de poda*, en relación al número de capas y neuronas de la red, este criterio solo sirve para iniciar el proceso, el número de capas y neuronas de la red se consolidarán de acuerdo con el desempeño de la red basado en el *error cuadrático medio* que se obtenga en el entrenamiento.

Se obtienen los patrones para entrenar a la red, los cuales contienen las diferentes combinaciones de parámetros eléctricos, sus valores y rangos de operación dentro de las condiciones nominales, que se estima tendrán durante la operación del sistema, de tal manera que sean patrones representativos del funcionamiento tal como se muestra en la tabla 1

Tabla 1. Combinaciones de diferentes cargas para entrenamiento

RESISTENCIA (Ω)	INDUCTANCIA (mH)			CAPACITANCIA (μF)			V_L
	1	10	100	10	100	1000	
10	10 Ω 1mH	10 Ω 10mH	10 Ω 100mH	10 Ω 10 μF	10 Ω 100 μF	10 Ω 1000 μF	0.1 V_L
20	20 Ω 1mH	20 Ω 10mH	20 Ω 100mH	20 Ω 10 μF	20 Ω 100 μF	20 Ω 1000 μF	
30	30 Ω 1mH	30 Ω 10mH	30 Ω 100mH	30 Ω 10 μF	30 Ω 100 μF	30 Ω 1000 μF	
10	10 Ω 1mH	10 Ω 10mH	10 Ω 100mH	10 Ω 10 μF	10 Ω 100 μF	10 Ω 1000 μF	0.9 V_L
20	20 Ω 1mH	20 Ω 10mH	20 Ω 100mH	20 Ω 10 μF	20 Ω 100 μF	20 Ω 1000 μF	
30	30 Ω 1mH	30 Ω 10mH	30 Ω 100mH	30 Ω 10 μF	30 Ω 100 μF	30 Ω 1000 μF	
10	10 Ω 1mH	10 Ω 10mH	10 Ω 100mH	10 Ω 10 μF	10 Ω 100 μF	10 Ω 1000 μF	1.1 V_L
20	20 Ω 1mH	20 Ω 10mH	20 Ω 100mH	20 Ω 10 μF	20 Ω 100 μF	20 Ω 1000 μF	
30	30 Ω 1mH	30 Ω 10mH	30 Ω 100mH	30 Ω 10 μF	30 Ω 100 μF	30 Ω 1000 μF	
10	10 Ω 1mH	10 Ω 10mH	10 Ω 100mH	10 Ω 10 μF	10 Ω 100 μF	10 Ω 1000 μF	1.5 V_L
20	20 Ω 1mH	20 Ω 10mH	20 Ω 100mH	20 Ω 10 μF	20 Ω 100 μF	20 Ω 1000 μF	
30	30 Ω 1mH	30 Ω 10mH	30 Ω 100mH	30 Ω 10 μF	30 Ω 100 μF	30 Ω 1000 μF	

Esta tabla de combinaciones da como resultado la generación de 84 patrones que describen la operación del sistema, para el caso de estudio, para cargas lineales.

Con el fin de obtener buenos patrones de ejemplo para la red neuronal para entrenamiento fuera de línea, se implementa un modelo de simulación que pueda realizar la operación de control del RDV, bajo las diferentes condiciones de carga lineal y diversas fallas, por lo que se implementa el sistema con ayuda del Software Simulink de *Matlab* (MatLab, 2010) y utilizando la librería *SimPowerSystems*, de manera que se pueda capturar la información del comportamiento del sistema, el circuito implementado se muestra en la figura 4.

Se realiza la simulación del circuito para cada combinación mostrada en la tabla 1, identificando y guardando los diferentes archivos generados que contienen los datos de los parámetros eléctricos, se realiza esto para cada combinación modificando los valores y ajustando los parámetros de los controladores para obtener un buen comportamiento del sistema.

Los parámetros K_p y K_i del control PI, son diferentes para cada condición de carga, por lo que se ajustan para condiciones de óptimo funcionamiento, de tal manera que se obtenga una salida de voltaje con la mínima distorsión armónica y el menor error del estado estacionario en el voltaje RMS (Raíz Cuadrática Media).

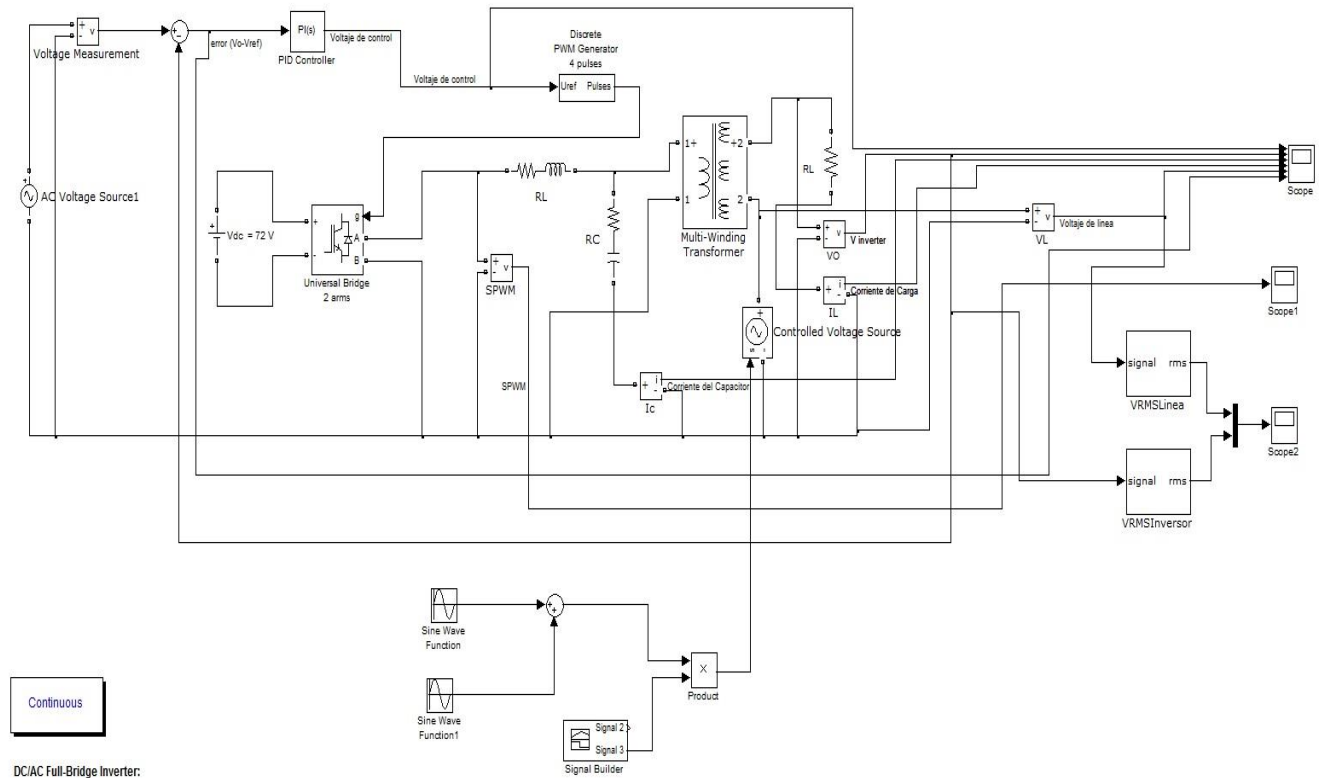


Figura 4. Controlador implementado en *Simulink* de *MatLab*, para obtener patrones de entrenamiento de la RNA.

La salida de voltaje, corriente de carga, corriente del capacitor, voltaje de línea y error respecto al voltaje de referencia, son recopiladas como entradas a la Red Neuronal y la señal de compensación como salida de la Red Neuronal, estos datos son guardados para el entrenamiento de la Red fuera de línea.

Después de correr la simulación para cada combinación propuesta por la tabla 1, se obtienen 84 patrones de entrenamiento, tomando en cuenta que el numero de datos representativos de cada patrón de los parámetros eléctricos cumpla con los criterios del Teorema de Muestreo (Nyquist) de acuerdo a (Irrázaval, 1999), donde establece que una función continua de ancho de banda limitado puede ser recuperada de sus muestras, siempre que hayan sido obtenidas con una frecuencia de muestreo superior o dos veces la máxima frecuencia contenida en la función original, esto debe aplicarse para la señal de más alta frecuencia que intervenga en el sistema.

Los 84 patrones para entrenamiento, se agrupan en un sólo arreglo separándolos en datos de entrada y datos de salida. Posteriormente con los dos arreglos ya consolidados y validados, se crea la red y se entrena, iniciando con una propuesta de arquitectura mínima, se comparan resultados y se efectúan las pruebas requeridas agregando neuronas o capas de acuerdo al resultado del entrenamiento, siempre buscando el máximo desempeño con el mínimo número de capas y neuronas.

Las diferentes pruebas realizadas con diferentes combinaciones de la arquitectura de la red neuronal artificial, se muestran en la tabla 2.

Tabla 2. Pruebas con diferentes estructuras de Red Neuronal

Red	funcion	Epocas	Tiempo	Desempeño	Gradiente	Momento	Pruebas de validacion	Regresion
3-2	TT	375	00:35:08	0.0194	0.00347	0.1	0	0.9654
4-2	TT	129	00:13:03	0.013	0.000575	1	0	0.96544
4-3	TT	191	00:19:44	0.0107	0.0181	1	0	0.9805
5-2	TT	246	00:26:15	0.0103	0.0154	10	0	0.9812
5-3	TT	194	00:20:44	0.0101	0.203	1	0	0.98174
4-3-2	TTT	274	00:29:18	0.0101	0.018	1	6	0.98169
4-3-2	TTT	304	00:32:36	0.0123	0.00806	10	0	0.97801
4-3-2	TLT	1000	01:43:27	0.0125	0.0115	1	0	0.97722
5-3-2	TTT	267	00:29:55	0.0116	0.854	10	6	0.97911
5-4-2	TTT	710	01:21:27	0.00792	0.00111	1	6	0.98566
5-4-2	TTT	1000	07:30:41	0.00888	1.53	10	0	0.9839
5-4-2	TLT	1000	01:53:06	0.0148	0.00239	10	0	0.97287
5-4-3	TTT	1000	01:57:22	0.00978	0.229	10	0	0.9822
5-4-3	TLT	1000	01:57:28	0.00962	0.422	1	0	0.98259
5-4-3	TTL	1000	01:59:38	0.00942	0.00686	1	0	0.98292
5-4-3	TLT	1000	07:38:18	0.00649	0.00695	10	0	0.98827

Una vez obtenidos los resultados del entrenamiento, tal como se muestra en la tabla 2, se elige la arquitectura que nos proporciona la regresión más alta con **el mínimo error cuadrático**, en este caso se obtiene un valor de **0.00649**, por lo que se elige esta arquitectura de red, mostrada en la figura 4 la cual está formada por 5 entradas y una salida, 3 capas internas de 13 neuronas ordenadas en una estructura **5-4-3-1** y funciones de activación *tansig*, *logsig* y *tansig* en sus capas internas.

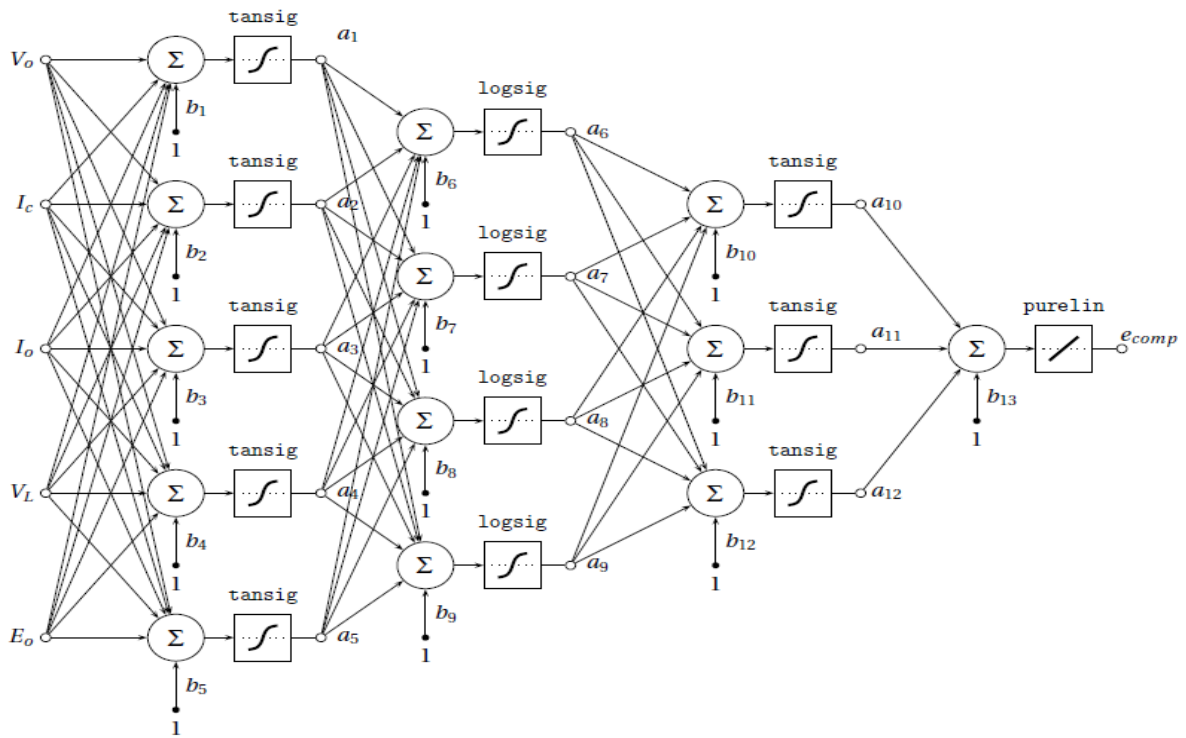


Figura 4. Red neuronal FeedForward 5-4-3-1.

Las operaciones que se llevan a cabo en la RNA tipo FeedForward están determinadas por las siguientes ecuaciones de la 3 a la 10, que contienen los pesos y bias, resultado del Algoritmo de asignación de estos durante el proceso de entrenamiento (Retropropagación).

$$a^1 = f^1 \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} \\ W_{51} & W_{52} & W_{53} & W_{54} & W_{55} \end{bmatrix} \begin{bmatrix} v_o \\ I_c \\ I_o \\ v_L \\ e_o \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix} \quad (3)$$

$$a^2 = f^2 \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} + \begin{bmatrix} b_6 \\ b_7 \\ b_8 \\ b_9 \end{bmatrix} \quad (4)$$

$$a^3 = f^3 \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \end{bmatrix} \begin{bmatrix} a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \end{bmatrix} + \begin{bmatrix} b_{10} \\ b_{11} \\ b_{12} \end{bmatrix} \quad (5)$$

$$e_o = f^4 [W_{11} \ W_{12} \ W_{13} \ W_{14}] \begin{bmatrix} a_{10} \\ a_{11} \\ a_{12} \end{bmatrix} + [b_{13}] \quad (6)$$

$$e_0 = f^4(W^4 f^3(W^3 f^2(W^2 f^1(W^1 p + b^1) + b^2) + b^3) + b^4) \quad (7)$$

$$f^1 = f^3 = \text{tansig} = \frac{e^n - e^{-n}}{e^n + e^{-n}} \quad (8)$$

$$f^2 = \text{logsig} = \frac{1}{1 + e^{-n}} \quad (9)$$

$$f^4 = \text{purelin} = 1 \quad (10)$$

Conclusiones.

La metodología para la determinación de la arquitectura y topología de la RNA, descrita en este artículo, está basada en la experimentación, para lograr un diseño óptimo este artículo pretende proporcionar el punto de partida y la secuencia de pasos a seguir para el diseño de una red neuronal y la generación de los patrones de entrenamiento de señales eléctricas, este método puede aplicarse a otro tipo de circuitos se deseen implementarse obteniendo resultados similares.

Los resultados de pesos y *bias* de la RNA pueden implementarse en tarjetas de desarrollo para realizar funciones de control de una manera eficiente.

Referencias bibliográficas.

Berry, M.J.A., Linoff, G. (1997). *Data Mining Techniques*. John Wiley & Sons. Nueva York.

El-Gammal, M. A., Abou-Ghazala, A. Y., y El-Shennawy, T. I. (2011). Dynamic Voltage Restorer (DVR) for Voltage Sag Mitigation. *International Journal on Electrical Engineering and Informatics*.

Hecht-Nielsen, R. (1990) *Neurocomputing*, Reading, MA: Addison-Wesley.

Irarrázaval, P.(1999). *Análisis de Señales*. Impresos Universitaria S.A., pp.110-111 McGraw Hill Interamericana.

Lippmann, R. (1987). An introduction to computing with neural nets. *ASSPMagazine, IEEE*.

Masters, T. (1993). *Practical neural network recipes in C++*. Academic Press Professional, Inc., San Diego.

M. T. Hagan, H. B. Demut (1996). *Neural Network Design*. PWS Publishing Company, pp. 1-8. Ed. Thompson.

MatLab (2010), *The MathWorks*.

Rosli, O., y Rahim, N. A. (2009). Mitigation Of Voltage Sags/Swells Using Dynamic Voltage Restorer (DVR). *ARPJ Journal of Engineering and Applied Sciences*.

Swingler, K. (1996). *Applying Neural Networks, A Practical Guide*. Press Limited Oval Road London NW1.

Información de los autores.



José Ángel Zepeda Hernández, es Ingeniero Industrial en Eléctrica y Maestro en Ciencias en Ingeniería Mecatrónica, egresado del Instituto Tecnológico de Tuxtla Gutiérrez ITTG. Es profesor de tiempo completo en el área de Ingeniería Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez e investigador desde 1999, Imparte cátedra en el área de Ingeniería Electrónica y la Maestría en Ciencias en Ingeniería Mecatrónica en el ITTG, donde imparte las asignaturas de Diseño con Transistores y Electrónica Básica respectivamente. Líder de la Línea de Investigación y Generación del Conocimiento “Instrumentación”.



Aldo Esteban Aguilar Castillejos, obtuvo el grado de Maestro en Ciencias en Ing. Mecatrónica en el Instituto Tecnológico de Tuxtla Gutiérrez (ITTG) en el 2011. Actualmente es parte del cuerpo académico del ITTG y universidades privadas, con desarrollos de proyectos financiados por CONACyT y empresas privadas. Con campos de interés en: dispositivos opto-mecatrónicos, manufactura, sensores de fibra óptica, instrumentación y control.



Álvaro Hernández Sol, es Ingeniero en electrónica, egresado del Instituto Tecnológico de Tuxtla Gutiérrez ITTG, en Tuxtla Gutiérrez, Chiapas. Es profesor de tiempo completo en el área de Ingeniería Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez y es investigador desde 1997. Siendo Jefe de proyectos de investigación de ingeniería electrónica desde el 2001. Fundó y asesora el “Club de robótica del ITTG”. Colabora en la línea de investigación de “Robótica” de Ingeniería electrónica. Y dirige el área de trabajo en “sistemas aéreos autónomos”.



María Catalina Salgado Gutiérrez, es Ingeniera en Electrónica y Maestra en Administración, egresada del Instituto Tecnológico de Tuxtla Gutiérrez ITTG. Es profesora de asignatura en el área de Ciencias Básicas. Imparte cátedra en las áreas de Ingeniería Electrónica y Ciencias Básicas en el ITTG.