

Inteligencia artificial: implementación de una neurona artificial en un circuito eléctrico

SANDRA BELÉN BEJARANO JIMÉNEZ, SOFÍA MONTSERRAT PÉREZ PÉREZ, MARÍA FERNANDA SILVA HERNÁNDEZ, ESTEPHANIA VILLAVICENCIO RÍOS

Resumen— El desarrollo de redes neuronales artificiales tiene una amplia gama de aplicaciones dentro del campo de la Ingeniería Biomédica, por lo que se decide crear una neurona artificial como base para futuros proyectos de índole didáctico y académico. En este artículo se da a conocer el procedimiento realizado para crear una neurona artificial que utiliza entradas binarias y configuraciones sencillas de amplificadores operacionales para cumplir con el funcionamiento lógico de las compuertas “OR” y “AND”. El proyecto se realiza de manera teórica, es decir, con una justificación matemática, en simulación y físicamente sobre un protoboard.

I. INTRODUCCIÓN

El sistema nervioso es donde se llevan a cabo los procesos de pensamiento y control del cuerpo, de forma natural son las neuronas biológicas quienes se encargan de recibir y transmitir estímulos que llegan al cuerpo de forma tanto interna como externa [1].

Para que estas neuronas conduzcan la información a través de ellas, es necesaria la existencia de un estímulo de entrada superior al umbral electronegativo de la membrana, una vez que éste es superado, la neurona entera es despolarizada comenzando el proceso de sinapsis, proceso por el cual las neuronas se comunican entre sí y con las demás células del cuerpo, ver Figura 1 [1].

Por otra parte, en inteligencia artificial, una neurona artificial es la unidad equivalente a la neurona biológica, que intenta simular su comportamiento y funcionamiento; de modo que disponen de entradas y salidas. En el modelo artificial se plantean valores, llamados pesos, a las conexiones de entrada los cuales definen el estado de activación de la célula, ver Figura 2 [2].

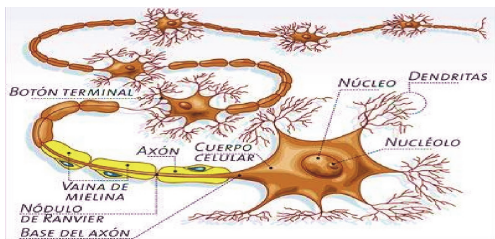


Figura 1. Neurona biológica en sinapsis

En este trabajo se pretende emular el comportamiento de compuertas lógicas OR y AND a través de una neurona

BEJARANO JIMÉNEZ SANDRA, PEREZ PEREZ SOFIA, SILVA HERNÁNDEZ MARÍA FERNAND, VILLAVICENCIO RIOS ESTEPHANIA pertenecen a la carrera INGENIERÍA BIOMÉDICA de la Facultad de Ingeniería y realizaron el proyecto dentro del curso DISPOSITIVOS ELECTRÓNICOS (Email: sbejaranojimenez@gmail.com).

El proyecto fue asesorado por JOSÉ AMBROSIO BASTIÁN.

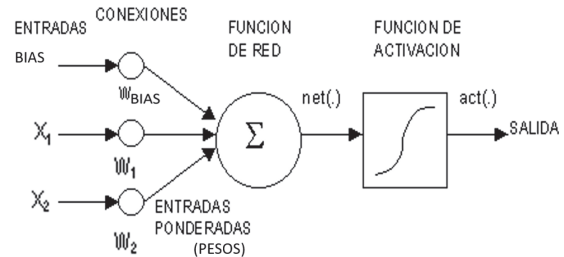


Figura 2. Modelo de la neurona artificial

artificial implementada en un circuito electrónico, utilizando para ello en las entradas valores de tensión que correspondan al “uno lógico” o al “cero lógico”, dispositivos electrónicos y eléctricos tales como transistores, amplificadores operacionales (opamps), resistencias, entre otros [3].

En la Figura 3 se muestran las tablas de verdad de las compuertas lógicas referidas OR y AND, en resumen tenemos que para la compuerta OR se requiere que al menos uno de los dos valores de entrada equivalga al 1 lógico, de lo contrario la salida será cero, para la compuerta AND se requiere que los dos valores lógicos de entrada sean 1 para poder tener un valor de salida igual a 1.

OR			AND		
A	B	A+B	A	B	AB
0	0	0	0	0	0
0	1	1	0	1	0
1	0	1	1	0	0
1	1	1	1	1	1

Figura 3. Tablas de verdad de las compuertas OR y AND

II. DESCRIPCIÓN DE ETAPAS Y SU DESARROLLO

A continuación se describen las etapas principales del proyecto y la relación entre ellas.

A. Aprendizaje

Para la etapa de aprendizaje se utilizó una neurona tipo Adaline, la cual es considerada como la primera red neuronal lineal, fue desarrollada como circuito electrónico adaptivo para usos distintos a lo que hoy se conoce como “red neuronal”. El modelo de neurona artificial, por lo tanto, consta de los siguientes elementos:

1. Entradas x_i de la célula, estas toman los valores lógicos de 0 ó 1.
2. El bias es la entrada de la célula que se mantiene constante durante todo el proceso.

3. Pesos son los valores ponderados que se conectan a las entradas x_i .
4. $S = \{s_1, \dots, s_n\}$: estado interno de la neurona o nivel de activación de la célula.
5. Función de activación o de salida: permite cambiar el estado de la célula en base a las entradas y define las nuevas salidas.

B. Simulación

En esta etapa, se desarrolló un programa que simula una neurona artificial tipo Adaline (programado en Matlab), donde están consideradas las etapas de aprendizaje y de reconocimiento para el emular el funcionamiento de las compuertas AND y OR. Posteriormente y de acuerdo a los resultados del programa anterior se simuló en Multisim, arreglos de configuraciones básicas con amplificadores operacionales utilizando factores de ganancia acordes a los pesos entregados por la simulación de la neurona artificial. Para este paso fue necesario apoyarse en una tabla de Excel que entrega los valores de las resistencias idóneas para las ganancias indicadas.

C. Implementación

La implementación consiste en armar físicamente el circuito, para ello, primero se verifica mediante simulación del circuito en Proteus hasta que coincida con los resultados del programa de Matlab. Por ejemplo en la Figura 5 (lado izquierdo), se muestra la construcción del circuito que emula la compuerta OR, en donde se utilizaron resistencias de 3.3kΩ, 3.9kΩ, 1.8kΩ, 1kΩ, 330Ω, op-amps, leds y protoboard, en esta misma figura se observa el correspondiente circuito para la compuerta AND (lado derecho).

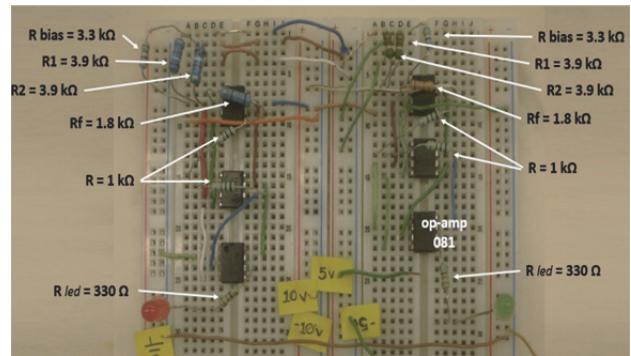


Figura 5. Implementación del circuito AND y OR

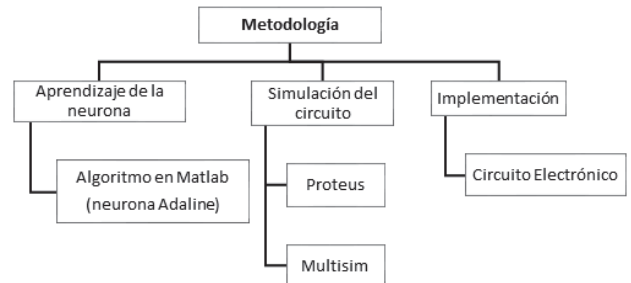


Figura 6. Diagrama de bloques del proceso metodológico

para que las entradas produzcan las salidas deseadas, se conoce como entrenamiento de la neurona. Finalmente cuando se tienen los pesos idóneos se almacenan estos valores ya que serán los valores de las ganancias de nuestros circuitos posteriormente.

Se debe considerar que los valores de los pesos de la neurona serán nuestras ganancias en las configuraciones de nuestros circuitos con OPAMPS, desafortunadamente no existen comercialmente todos los valores de resistencias eléctricas que permitan tener valores de ganancias iguales a las obtenidas en la simulación de Matlab para los pesos. Para evitar hacer varias pruebas con valores y reajustar los valores de resistencias, se desarrolló una herramienta extra apoyados en una hoja de Excel (véase Figura 8 y 9), donde fuera aún más fácil cambiar el valor de alguna resistencia y el ajuste del resto de valores, permitiendo un rápido acercamiento a valores comerciales de resistencias.

B. Simulación del circuito

Se diseñó un circuito eléctrico utilizando configuraciones

III. METODOLOGÍA

La Figura 6 muestra los pasos más importantes realizados. A continuación se describen cada uno de ellos.

A. Aprendizaje de la neurona y ajustes de ganancia

Primero se utilizó el programa de Matlab para que la neurona tipo Adaline pudiera llevar a cabo el aprendizaje de las funciones lógicas OR y AND, este programa realiza una serie de iteraciones hasta conseguir que la neurona artificial tenga en sus pesos los valores que permiten emular el comportamiento de la compuerta deseada. Este proceso iterativo donde se ajustan los pesos de la neurona artificial

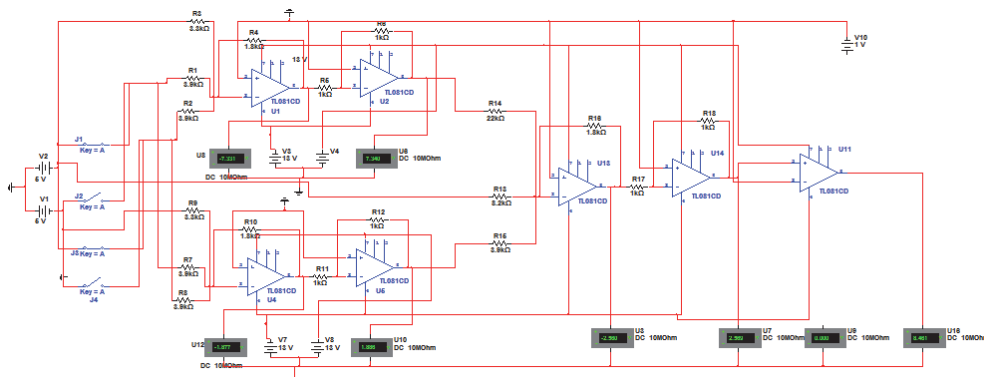


Figura 4. Simulación en Multisim

básicas de OPAMPS (sumador, inversor y sumador inversor) que permitieran incluir las ganancias que se obtuvieron en el aprendizaje de la neurona. En este paso es de gran utilidad simular un circuito previamente a ser armado, para ello se utilizaron los simuladores Proteus y Multisim, estas simulaciones nos permitieron hacer una comparación entre los resultados matemáticos de nuestro diseño y del simulador para las funciones lógicas OR y AND; de esta forma se aseguran los datos antes de ser implementados en el circuito de manera física. Cabe mencionar que en ocasiones al no obtener valores comerciales de resistencias que se apegaran a los valores de los pesos que entrega la simulación de la neurona artificial, se tenía que repetir todo el procedimiento. Finalmente si en la simulación del circuito se obtienen los resultados esperados de acuerdo a las tablas de verdad de las compuertas OR y AND, se procede a su implementación.

C. Implementación

Se procede a realizar físicamente en protoboard el circuito, para esto se consiguieron las resistencias con valores comerciales más cercanos a los de simulación y se colocaron LEDs al final de cada circuito para obtener una respuesta visual, de tal forma que se activara la luz cuando los valores lógicos fueran equivalentes a uno y se apagará cuando éstos fueran cero.

Como se muestra en la Figura 7; para las neuronas OR y AND, se usan 3 opamps realizando las operaciones de amplificador, inversor y comparador utilizando las ganancias ya descritas en sus respectivas configuraciones de acuerdo al diseño.

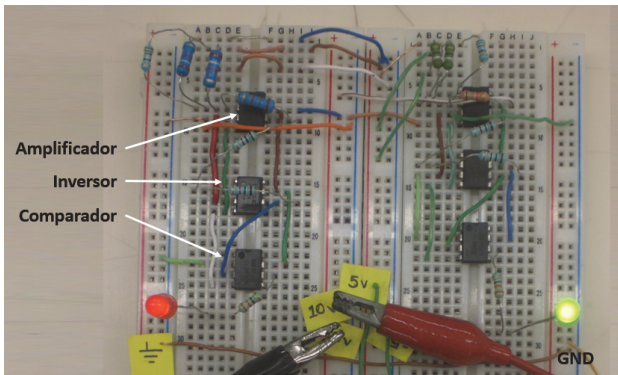


Figura 7. Circuito con entradas lógicas de 1. Izquierda AND. Derecha OR.

IV. RESULTADOS

Los resultados que se presentan provienen de las diferentes etapas del proyecto. Primeramente explicaremos los datos de las Figuras 8 y 9.

Las variables alpha y epsilon son usadas en la etapa de aprendizaje del algoritmo en Matlab y son valores que se utilizan para el aprendizaje y el criterio de paro del algoritmo respectivamente. “Dual” se refiere al voltaje con el que se alimenta los opamps. La letra ‘a’ hace referencia al voltaje usado con el valor lógico de 1, mientras que ‘b’ al valor lógico de 0. A partir de los pesos iniciales se obtienen los pesos finales y es a partir de éstos y del valor de R_f que se puede calcular las resistencias a implementar en el circuito.

Las ganancias son representadas con s's, y son obtenidas a partir del algoritmo en Matlab que son los voltajes en el circuito a la salida del inversor. Las celdas que representan los voltajes reales son calculados con el valor exacto de las resistencias mientras que las celdas de valores aproximados se calculan con las resistencias adaptadas.

Como se puede observar en la Figura 8, la diferencia entre los valores de los voltajes es mínima y son causados por las adaptaciones que se tuvieron que hacer a las resistencias.

De tal manera que podemos afirmar que el funcionamiento lógico de la neurona artificial trabaja de la manera adecuada a la esperada.

V. CONCLUSIONES

En primer lugar, se destaca las herramientas, software y equipo que fue necesario conocer y emplear para la finalización de este proyecto. Con respecto a la etapa de aprendizaje es importante ser persistente en la variación de los valores iniciales usados en el código; ya que de éstos dependen los resultados obtenidos en las etapas posteriores.

OR									
alpha=	0.00001	epsilon=	0.01	dual	10V	a=	5	b=	-5
w's iniciales	valores aleatorios			Bias	V1	V2	s's		
w's finales	0.5	0.4994	0.4987		5	-5	-5	-2.4994	
Rf =	1.8				5	-5	5	2.4969	
R1=	3.6	R2= 3.604325	R3= 3.609384		5	5	-5	2.5031	
R1=	3.3	R2= 3.9	R3= 3.9		5	5	5	7.4906	
Reales					Aproximadas				
Vo1		Vo2		Vo1		Vo2			
2.4905		-2.4905		1.88811888		-1.88811888			
-2.4965		2.4965		-2.727272727		2.727272727			
-2.5035		2.5035		-2.727272727		2.727272727			
-7.4905		7.4905		-7.342657343		7.342657343			
Circuito					Proteus				
Vo1		Vo2		Vo3		Vo1		Vo3	
1.95		-1.95		-8.66		1.88		-1.89	
-2.74		2.7		6.14		-2.74		2.73	
-2.76		2.72		6.135		-2.74		2.73	
-7.44		7.47		6.13		-7.35		7.34	

Figura 8. Resultados OR

AND									
alpha=	0.00001	epsilon=	0.01	dual	10	a=	5	b=	-5
w's iniciales	valores aleatorios			Bias	V1	V2	s's		
w's finales	0.5	0.5006	0.5012		-5	-5	-5	-7.5093	
Rf =	1.8				-5	-5	5	-2.4969	
R1=	3.6	R2= 3.595685	R3= 3.591381		-5	5	-5	-2.5032	
R1=	3.3	R2= 3.9	R3= 3.9		-5	5	5	2.5092	
Reales					Aproximadas				
Vo1		Vo2		Vo1		Vo2			
7.509		-7.509		7.342657343		-7.342657343			
2.497		-2.497		2.727272727		-2.727272727			
2.503		-2.503		2.727272727		-2.727272727			
-2.509		2.509		-1.88811888		1.88811888			
Circuito					Proteus				
Vo1		Vo2		Vo3		Vo1		Vo3	
7.55		-6.83		-8.59		7.33		-7.34	
2.7		-2.76		-8.59		2.71		-2.72	
2.79		-2.78		-8.59		2.71		-2.72	
-1.92		1.91		8.5		-1.9		1.89	

Figura 9. Resultados AND

El uso de los simuladores de circuitos es un punto a destacar ya que nos permitió guiarnos y anticiparnos a posibles errores de diseño del circuito.

En la implementación, es importante considerar que el voltaje de alimentación es un factor determinante en el comportamiento de los opamps, ya que si no se alimentan debidamente no se cumple con la función de comparador, inversor o sumador, según sea el caso.

Finalmente, consideramos que los temas desarrollamos a lo largo de este proyecto, son relevantes en el campo de Ingeniería Biomédica y por tanto parte de nuestra formación.

AGRADECIMIENTOS

Agradecemos a la Universidad La Salle por haber brindado el acceso a los talleres y por el suministro del material para

ensamblar el circuito. Al profesor José Ambrosio Bastián por haber leído y corregido éste artículo incontables veces. Al profesor Josué Jiménez Rodríguez por su asesoría durante el proceso de implementación.

REFERENCIAS

- [1] A Guyton, J. Hall. "Tratado de Fisiología Médica". Elsevier Saunders, 2011.
- [2] Pedro Ponce Cruz, "Inteligencia Artificial con Aplicaciones a la Ingeniería", Alfaomega, 2010.
- [3] G. Ruiz "Electrónica Básica para Ingenieros" Universidad de Cantabria, 2009.
- [4] P. Viñuela, I. Galván "Redes de Neuronas Artificiales. Un enfoque Práctico". Pearson Prentice Hall, 2003.
- [5] J. Hílera, V. Martínez. "Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones". Alfaomega, 2000.