

Compuertas lógicas con neuronas artificiales

M. Orozco García, *Estudiante de Ingeniería Biónica, UPIITA*

Resumen—Se han desarrollado análogos artificiales de las neuronas, los cuales emulan el comportamiento general de una neurona biológica. Estas son utilizadas en la creación de sistemas adaptativos que generen comportamientos que respondan a un set de entrenamiento y generen respuestas a entradas fuera y dentro de dicho set. Para el presente trabajo se emularán las compuertas “AND”, “OR” y “XOR” utilizando neuronas tipo Perceptrón y tomando como set de entrenamiento sus tablas de verdad. Creamos la neurona utilizando componentes electrónicos como OPAMS y resistencias para que esta reaccione a los valores lógicos de dos entradas y nos genere el valor lógico de salida. Dado que la compuerta “XOR” no es separable de manera lineal se utilizará un arreglo de 3 neuronas, dos en la primera capa y una en la segunda para generar esta compuerta.

Índice de términos—Neuronas artificiales, neurona Perceptrón, compuertas lógicas, set de entrenamiento.

I. INTRODUCCIÓN

“EN los últimos años han surgido varias técnicas inteligentes, tales como la Lógica Borrosa, Redes Neuronales, Algoritmos Genéticos y Razonamiento Probabilístico, las cuales se han agrupado bajo el nombre de *Computación Suave* o *Soft Computing*. Este nombre es usado para diferenciarla de la computación *dura*, *hard* o *tradicional*.” [1]

Las redes neuronales es que no hacen diseños de alto nivel, generan sistemas a partir de modelos genéricos que se van adaptando e imitan mecanismos que nos caracterizan como seres inteligentes. Se emplean generalmente para aproximaciones, predicciones, clasificaciones o agrupaciones.

Las neuronas cuentan con las características de robustez y tolerancia a fallos, adaptación y plasticidad, manejo de información inconsistente, difusa y con ruido, manejo de grandes cantidades de datos por medio de paralelismo y aprendizaje continuo, diversidad en áreas de aplicación, tiempo de aprendizaje no determinado y diseño poco sistemático. [2]

Las neuronas cuentan con potenciadores e inhibidores que corresponden a las entradas, pesos, umbral de activación, función de activación y salida. [3]

En el caso de una red neuronal multicapa backpropagation como la que utilizaremos las salidas de la primera capa corresponden a las entradas de la segunda capa y así sucesivamente hasta conseguir las salidas finales. [3]

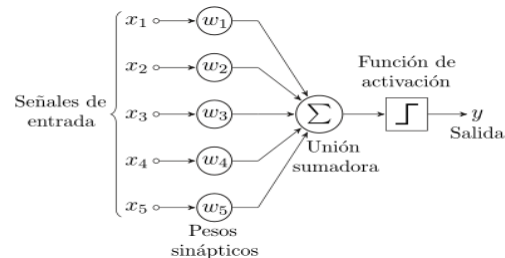


Fig. 1. Partes de la neurona artificial.

II. COMPUERTAS LÓGICAS

Se emularán las compuertas lógicas AND, OR y XOR, por medio de neuronas perceptrón cuyas representaciones gráficas en el plano se muestran a continuación:

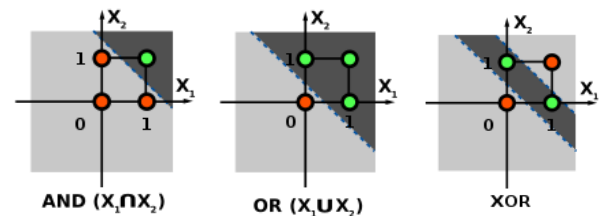


Fig. 2. Representación gráfica de compuertas lógicas AND, OR y XOR.

En estas gráficas podemos apreciar que la compuerta AND se puede separar con una línea recta entre los valores (0,1), (1,0) y el valor (1,1) esto es posible de realizar con una sola neurona perceptrón. La compuerta OR tiene una tendencia similar pero con el valor (0,0) como podemos apreciar. La compuerta XOR sin embargo muestra un comportamiento peculiar y solo es posible trabajar con ella al utilizar dos capas de neuronas, primero una capa de dos neuronas con las líneas que vemos en la gráfica que son una combinación de NAND y OR, y la segunda capa con una sola neurona ya que los resultados de estas son separables de forma lineal. Esto se verá con más detalle en las tablas de verdad mostradas a continuación.

Sus tablas de verdad corresponden a:

X1	X2	Y	X1	X2	Y	X1	X2	Y
0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	1	1
1	0	0	1	0	1	1	0	1
1	1	1	1	1	1	1	1	0
AND			OR			XOR		

Tabla. 1. Tablas de verdad de compuertas lógicas AND, OR y XOR.

Como se puede apreciar la compuerta AND y OR son clasificables de manera lineal, sin embargo la XOR requiere

de dos rectas para este fin. Por lo tanto utilizaremos un estado intermedio donde se utilizaran dos neuronas para cada una de estas rectas y otra más para obtener el resultado final que se colocara en una segunda capa. La tabla de verdad quedaría de la siguiente forma:

X1	X2	N1	N2	Y
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0
ENTRADAS		NEURONAS		SALIDA

Tabla. 2. Tabla de verdad con etapa intermedia de compuerta lógica XOR.

Observamos que N1 se comporta como una compuerta OR y N2 como una NAND o AND negada. Así mismo, quedando el caso con entradas (0,0) obsoleto, se puede generar una neurona N3 tipo AND cuyas entradas sean N1 y N2 para generar el comportamiento de la XOR a su salida.

En general la ecuación a la cual responderán las neuronas será:

$$Y = w1 * X1 + w2 * X2 - U$$

Siendo U el umbral de activación representado por un voltaje fijo y un comparador.

Las ecuaciones de clasificación serán las siguientes:

- AND: $Y = (X2 > 1.5 - X1)$
- OR: $Y = (X2 > 0.5 - X1)$
- XOR – N1: $N1 = (X2 > 0.5 - X1)$
- XOR – N2: $N2 = (X2 < 1.5 - X1)$
- XOR – N3: $Y = (N1 > 0.5 - N2)$

Estas ecuaciones muestran el comportamiento lineal X2 y X1 son las entradas, Y es la salida y N1 y N2 son salidas intermedias de la primer capa de neuronas presentes en el caso de la compuerta XOR. Los valores independientes representan el valor de comparación que utilizaremos en el circuito para hacer la clasificación.

III. CIRCUITO ELÉCTRICO

El circuito eléctrico se construyó por medio de OPAMS que fungirán como la función de activación y resistencias que constituirán los pesos; de la misma manera nos auxiliaremos de LEDs y de botones para introducir las entradas y visualizar las salidas de nuestras neuronas.

A. Lista de materiales.

En específico se utilizaron los siguientes componentes electrónicos:

- 29 resistencias de 10 kΩ
- 1 resistencia de 22 kΩ
- 2 resistencias de 100 kΩ
- 4 OPAMS serie TL084
- 3 Leds (uno para la salida de cada compuerta)
- 2 fuentes de 5V para alimentar el sistema
- 2 botones (para las entradas del sistema)
- Protoboard

- Cable para conexiones

B. Simulación.

En el entorno de simulación de circuitos eléctricos Proteus se realizaron las simulaciones de los circuitos antes de probarlos de forma física.

La compuerta AND consta de un sumador inversor, un inversor y un comparador no inversor a 1.5V como se mostró en la ecuación.

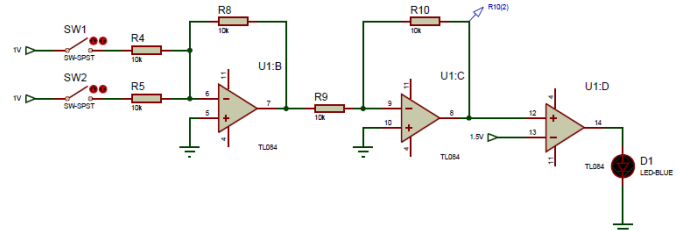


Fig. 3. Compuerta AND simulada.

La compuerta OR consta de un sumador inversor, un inversor y un comparador no inversor a 0.5V como se mostró en la ecuación.

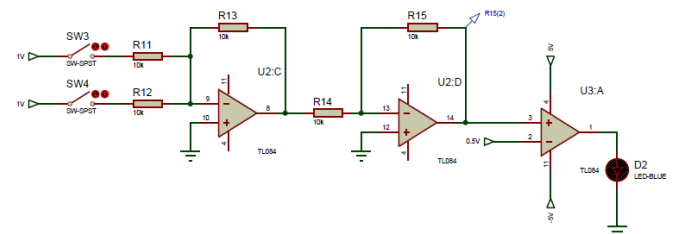


Fig. 4. Compuerta OR simulada.

La compuerta XOR consta de tres neuronas, la primera es una OR exactamente igual a la presentada con anterioridad. La segunda es una NAND que difiere de la AND en que se utiliza un comparador inversor para obtener la tendencia contraria. Al final estas respuestas se pasan por una AND con umbral de activación de 0.5V, es decir, el valor del comparador.

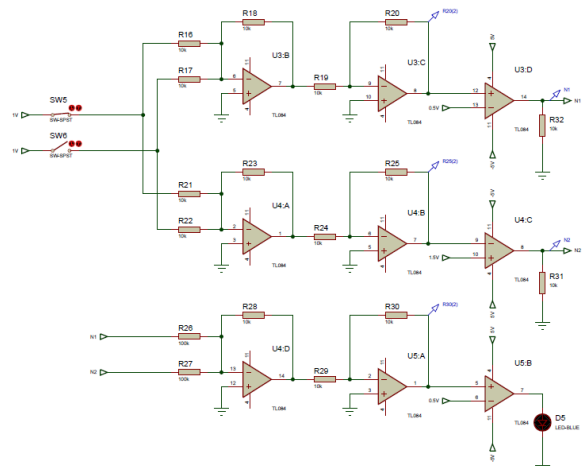


Fig. 5. Compuerta XOR simulada.

C. Implementación.

Con las componentes electrónicas físicas se implementó el circuito en protoboard para comprobar los resultados obtenidos en la protoboard. Estos siguen el diagrama presentado en la parte anterior con la única diferencia de que son reutilizables la neurona AND para la XOR-N3 y la neurona OR para la XOR-N1. Ya que estas presentan exactamente la misma ecuación y por razones económicas no se puede construir de nuevo el circuito.

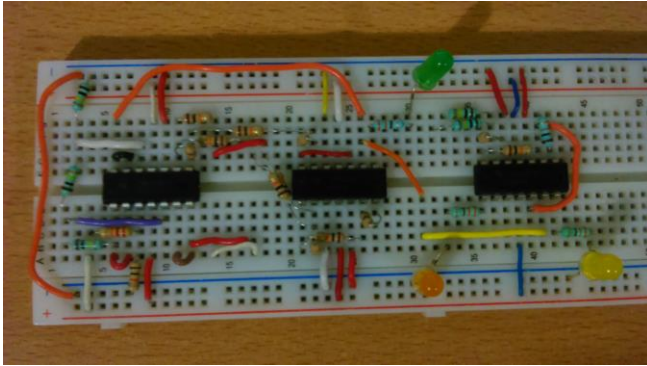


Fig. 6. Circuito armado en protoboard.

IV. RESULTADOS

El sistema cumple con los sets de entrenamiento que son las tablas de verdad propias de cada compuerta y también puede interpolar o extrapolar resultados fuera de este conjunto obteniendo así una respuesta más robusta y estable.

De la misma manera podemos mencionar que el circuito eléctrico presenta siempre la misma estructura sin importar la función de activación que se esté utilizando. Siempre es un sumador inversor, seguido de un inversor y un comparador.

La compuerta XOR fue la más compleja de emular ya que esta requirió tres neuronas y no solo una en una configuración multicapa con dos neuronas en la primera capa y una en la segunda. Esto es de especial reconocimiento ya que, de no ser posible clasificar dos conjuntos de forma lineal o de existir más de dos conjuntos siempre cabe la posibilidad de anexar más capas o más neuronas por capa. Esto originará un sistema más robusto pero al mismo tiempo aumentará el tiempo para el cálculo de los pesos (entrenamiento).

Sus tablas de verdad del sistema físico en volts corresponden a:

X1	X2	Y	X1	X2	Y	X1	X2	Y
0	0	0.02	0	0	0.01	0	0	0.02
0	5	1.14	0	5	1.03	0	5	0.69
5	0	1.05	5	0	1.04	5	0	0.69
5	5	2.15	5	5	2.08	5	5	0.02
AND			OR			XOR		

Tabla. 3. Tablas con respuestas del sistema físico en volts (V)

Como se clarificó con anterioridad para la compuerta AND el umbral de activación es 1.5V por lo que los valores obtenidos anteriormente al pasarlos por el comparador nos arrojan la tabla de verdad deseada. La compuerta OR posee un umbral de 0.5V por lo que sus valores también corresponden a

la tabla de verdad deseada. La compuerta XOR cumple con los valores de igual manera ya que su función de activación es 0.5 como se muestra con anterioridad.

Cabe mencionar que las neuronas no solo pueden ser implementadas en físico mediante circuitos eléctricos también pueden ser parte de un software donde el proceso de entrenamiento también puede ser programado y automático, en contraste con este caso cuya obtención fue manual.

V. REFERENCIAS

- [1] R. A. Magaña Pérez, "Sistemas de inferencia basados en Lógica Borrosa: Fundamentos y caso de estudio." Vol: 7, No: 1, Revista de Ingeniería de Sistemas e Informática, Enero - Julio 2010. Pp. 91-104.
- [2] I. M. Galvan & J. M. Valls, "Introducción a las redes de neuronas" Redes de Neuronas Artificiales, material didáctico para curso, Universidad Carlos III de Madrid.
- [3] I. M. Galvan & J. M. Valls, "Primeros Modelos Computacionales" Redes de Neuronas Artificiales, material didáctico para curso, Universidad Carlos III de Madrid.