

1. Introducción

En el año de 1943, el ser humano, en un intento de simular el comportamiento biológico del cerebro y entender como se lleva a cabo el complejo proceso del aprendizaje, propuso un modelo matemático capaz de aprender: llamado Neurona de McCulloch-Pitts, siendo este concepto la primera Neurona Artificial de que se tenga conocimiento. A partir de entonces, surgieron numerosos desarrollos científicos basados en esta aportación.

La Neurona Artificial proporciona los medios idóneos para realizar clasificación de patrones, imitando el funcionamiento del sistema biológico sensorial. Fue el primer modelo computacional de la red nerviosa del cerebro humano, y es en sí, el primer trabajo de un autómata de estados finitos.

2. Antecedentes

La Neurona biológica es la unidad básica del sistema nervioso. Se estima que el cerebro humano tiene aproximadamente 10^{11} neuronas y cada una puede conectarse con otras 10,000 neuronas. La cantidad de interconexiones en el cerebro humano promedio, está quizás en el orden de 10^{14} o 10^{15} . La neurona es una célula viva que no se duplica a si misma y durante el proceso individual de aprendizaje, puede desarrollar múltiples conexiones con otras neuronas.

Los principales elementos de una neurona son:

- Las dendritas, son ramificaciones (estructuras) que sirven para recibir las señales de entrada que serán procesadas por el cuerpo de la célula.
- El cuerpo de la célula (soma), realiza las funciones centrales de la célula.
- El axón, es el conducto que lleva las señales al punto de salida. Al final de éste es donde suceden las interacciones con otras neuronas. El punto en el cual dos o más neuronas interactúan es conocido como *sinapsis*. La fisura sináptica, es el área donde se encuentran la terminal del axón de una neurona y la dendrita de otra. La señal es transmitida por un proceso químico conocido como neurotransmisión.

La siguiente figura muestra los componentes de una neurona biológica.

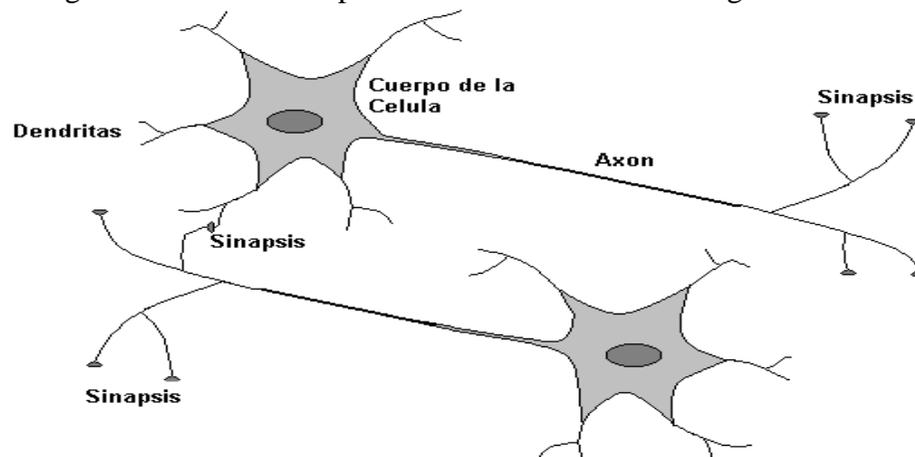


Fig. 1. Componentes de una neurona biológica.

El modelo de la red neuronal biológica, está basado fundamentalmente en la interconexión existente entre nodos (unidades de procesamiento que simulan la acción de una neurona). Las entradas y salidas se enlazan entre elementos, simulando una sinapsis (uniones especializadas mediante las cuales las células del sistema nervioso envían señales unas a otras). Se le asignan pesos a las entradas para simular la acción de la neurotransmisión. Un algoritmo es usado para ajustar los pesos de las entradas y así producir la señal de salida deseada, esto simula el proceso de aprendizaje. Tales algoritmos son frecuentemente referidos como algoritmos de aprendizaje o algoritmos de entrenamiento.

3. Aprendizaje

El aprendizaje, lejos de ser programado, es una habilidad intrínseca del ser humano. El aprendizaje puede ser visto como un cambio en el comportamiento adquirido, resultante de la práctica o experiencia a lo largo del tiempo. Esto ocurre si la efectividad de acoplamiento entre las neuronas es incrementada.

Existen dos tipos de aprendizaje: el primero de ellos, se obtiene como resultado del entrenamiento, generalmente es utilizado en las redes de neuronas artificiales; el segundo tipo, podría decirse que es el que poseen los seres humanos. La capacidad de incrementar el conocimiento, sumado a la habilidad para razonar, permite eventualmente la transformación de las diversas destrezas, sin embargo ambos tipos de aprendizaje se complementan y requieren tiempo para efectuarse.

El aprendizaje puede ocurrir en dos modos: supervisado y no supervisado. Para llevar a cabo procesos de aprendizaje supervisado, es necesario contar con ejemplos (instancias) provenientes de bases de datos de entrenamiento. En el aprendizaje no supervisado, no es necesario contar con instancias conocidas, es decir, el aprendizaje sucede de manera adaptativa.

El aprendizaje no supervisado es muy común en sistemas biológicos, consecuentemente, también es aplicado en las redes neuronales artificiales, especialmente cuando no se cuenta con información disponible para su entrenamiento.

4. Modelo Básico de Neurona Artificial

Los autores de este modelo visualizaron al cerebro como un organismo computacional. Propusieron un enfoque lógico y pensaron en la construcción de una máquina con una estructura abstracta, similar a las redes neuronales biológicas, enfocadas desde una perspectiva lógica combinatoria. Los primeros esfuerzos dieron más importancia al proceso de aprendizaje resultante de la interconexión entre las unidades. Se demostró que las unidades sencillas conectadas entre sí (mediante sinapsis), en presencia de elementos excitadores e inhibidores, aunado a la definición apropiada de un valor de umbral para la activación de la unidad de salida, son capaces de representar adecuadamente las leyes lógicas fundamentales. A estas neuronas básicas les llamaron neuronas formales y a la interconexión de varias unidades simples de neuronas formales se le llamó red neuronal.

La Neurona biológica es la unidad básica para el sistema nervioso. Similarmente, la Neurona Artificial es la unidad básica de una Red Neuronal Artificial (RNA).

Estas unidades básicas no solo permiten modelar matemáticamente los diversos procesos relacionados con el aprendizaje, sino que facilitan la simulación computacional y eventualmente la implementación física tanto en dispositivos semiconductores de alta escala de integración (VLSI) como en dispositivos de aplicación específica (ASIC).

Las características de una neurona artificial son:

- La salida de la neurona está en cualquiera de los dos estados: 1 o 0.
- La salida de la neurona depende de la suma de los pesos de las entradas. Cierta nivel debe ser superado para hacer que la neurona se active.
- La asociación de los pesos con las entradas modelan la eficiencia de la sinapsis (acoplamiento). Una mayor eficiencia de la sinapsis tendrá un mayor peso. La neurona es entrenada para ajustar esos pesos.

Estas características se muestran en la siguiente figura.

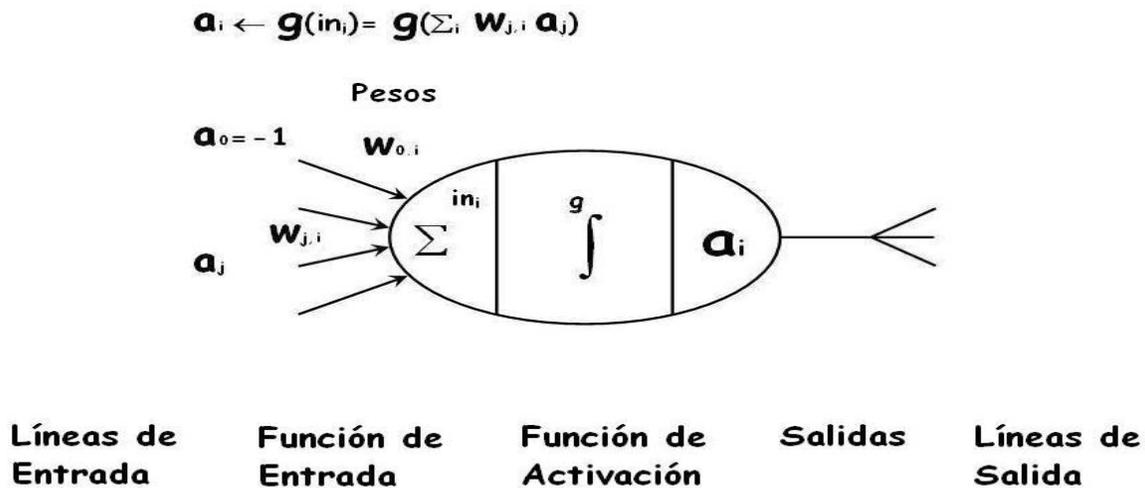


Fig. 2. Características de una neurona artificial.

El total de las entradas con sus respectivos pesos es igual a:

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_n x_n = \sum_{i=1}^n w_i x_i$$

Si esta suma excede el valor umbral, entonces la neurona se activa, es decir la salida es 1, de otra manera permanece apagada. Se trata de una unidad de proceso con n entradas x, \dots, x_n y una salida (y), con dos únicos valores de salida posibles y constantes. La unidad realiza una función de proceso sobre los valores de las entradas, y si el valor obtenido supera el umbral de activación, se obtiene como respuesta el valor de activación. Por el contrario, si el resultado de aplicar la función de proceso a los valores de entrada no supera este umbral, la neurona permanecerá inactiva y la salida será nula.

La función de proceso de las entradas es típicamente una suma ponderada de los valores de las mismas, lo que significa que no todos los valores de entrada tienen igual aportación para la suma,

sino que cada uno de ellos se modifica multiplicándolo por un valor de ponderación o peso, w_i , antes de proceder a sumar. Expresado de manera formal, el comportamiento de la neurona es el siguiente:

$$y = f \left[\sum_{i=1}^n w_i x_i - \theta \right]$$

Donde $f(x)$ representa a la función escalón (comúnmente conocida como la función de Heaviside) definida como:

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

w_i son los pesos sinápticos y θ el umbral de activación. Obsérvese que en función del signo de los pesos sinápticos, estos pueden actuar tanto como elementos activadores (signo positivo) o como elementos inhibidores (signo negativo).

El modelo de neurona artificial presenta numerosas analogías con las neuronas biológicas. Los cables o conexiones son análogos a las dendritas y axones, los pesos de ponderación de las conexiones equivalen a las sinapsis y el umbral de activación representa la actividad de la neurona.

Al eliminar el umbral de activación, es posible definir el comportamiento de la neurona mediante la siguiente expresión:

$$y = f \left[\sum_{i=1}^n w_i x_i \right]$$

Cabe mencionar que al eliminar el umbral de activación, los índices de las entradas implicadas, deben ser considerados a partir del elemento 1 y no desde el elemento 0. Ahora bien, si se observa la expresión, no cuesta reconocer en ella el producto escalar de dos vectores. Por tanto basta considerar el conjunto de n entradas de una neurona como un vector X de dimensión $n+1$ (asignando $x_0 = 1$), y el conjunto de pesos de la neurona como otro vector W de la misma dimensión.

Algunas veces puede sustituirse la función Heaviside por funciones que realizan transiciones en el tiempo (rampa o sigmoideal) y que además representan una opción atractiva, no solo por su fácil manejo sino por el bajo costo computacional de las operaciones implicadas.

La función sigmoideal se puede expresar como: $y = \frac{1}{1 + e^{-kx}}$

Donde y es la salida, x es el número total de entradas, siendo k el factor de ganancia que controla el punto de transición de 1 a 0.

La función sigmoideal, es frecuentemente utilizada porque su derivada es fácil de calcular.

Si $y = \frac{1}{1 + e^{-kx}}$, entonces su derivada es $\frac{dy}{dx} = ky(1 - y)$.

Para ilustrar como funciona la neurona de McCulloch-Pitts, se representará la neurona artificial, como sigue:

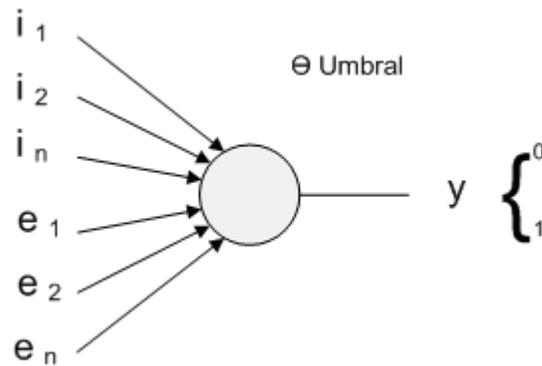


Fig. 3. Representación de una neurona artificial.

5. Ejemplos

5.1 Compuerta AND

La siguiente Neurona ha sido entrenada para responder con un valor de salida igual a 0, cuando se presenten los patrones 00,01 y 10. Cuando se presente el patrón de entrada 11 el valor de la salida será igual a 1, es decir, esta es la operación de una compuerta booleana AND. Como se puede observar en la tabla de verdad, la única manera de que la neurona se active es cuando ambas entradas excitadoras tengan el valor de 1, es decir $\theta=2$.

e_1	e_2	y
0	0	0
0	1	0
1	0	0
1	1	1

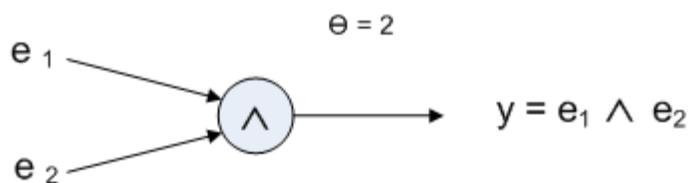


Fig. 4. Compuerta lógica AND, representada con el modelo de neurona artificial de McCulloch-Pitts.

5.2 Compuerta OR

La siguiente Neurona ha sido entrenada para obtener una salida de valor 0 cuando se presente el patrón 00. Para los demás patrones de entrada (01,10 y 11) la salida obtenida será de valor 1, es decir, esta es la operación de una compuerta booleana OR. Como se puede observar en la tabla de verdad, la salida tendrá el valor de 1 cuando cualquiera de las entradas sea 1, y por consiguiente el valor del umbral $\theta = 1$.

e_1	e_2	y
0	0	0
0	1	1
1	0	1
1	1	1

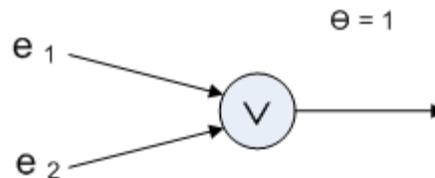


Fig. 5. Compuerta lógica OR, representada con el modelo de neurona artificial de McCulloch-Pitts.

5.3 Compuerta NOT

La siguiente Neurona ha sido entrenada para responder como un inversor (compuerta NOT booleana), de tal manera que al presentarle un 0 a la entrada, su respuesta a la salida será un 1 y viceversa, por consiguiente el valor del umbral $\theta = 0$.

i_1	y
0	1
1	0

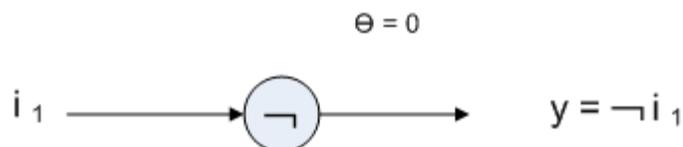


Fig. 6. Compuerta lógica NOT, representada con el modelo de neurona artificial de McCulloch-Pitts.

5.4 Flip – Flop (SR)

Hay dos tipos de circuitos secuenciales: síncronos y asíncronos. Su clasificación depende del tiempo de actuación de sus señales. Generalmente un circuito secuencial emplea elementos de almacenaje llamados flip-flops (FF) que sólo se les permite el cambio de su valor almacenado en intervalos discretos de tiempo. Un circuito secuencial es un sistema en el cual las salidas dependen del valor almacenado y de los valores presentes en las entradas.

Un circuito FF puede mantenerse en algún estado binario de forma indefinida (siempre que el suministro no se vea interrumpido). Un FF siempre se comporta como dispositivo de almacenamiento, controlado por una señal que marca el momento preciso en el cual el valor almacenado tiene que ser actualizado. La diferencia principal entre los diversos FF radica en el número de entradas que poseen y en la manera en la cual las entradas afectan el valor del estado binario almacenado.

Estos circuitos pueden construirse con compuertas NAND o NOR. Cada circuito forma un FF básico, la conexión y el acoplamiento cruzado mediante la salida de una compuerta a la entrada de otra constituye una trayectoria de retroalimentación. Por esta razón los circuitos se clasifican como secuenciales asíncronos. Cada FF tiene 2 salidas Q y Q' y dos entradas, SET para ajustar y RESET para restaurar. A este tipo de FF se les llama RS directamente acoplado o seguro.

S	R	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	?

Memoria
Reset
Set
Indeterminado

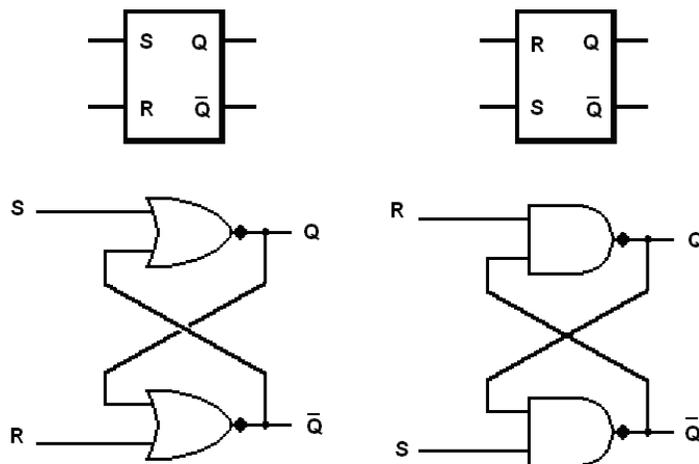


Fig. 7. Diagrama esquemático del Flip-Flop SR.

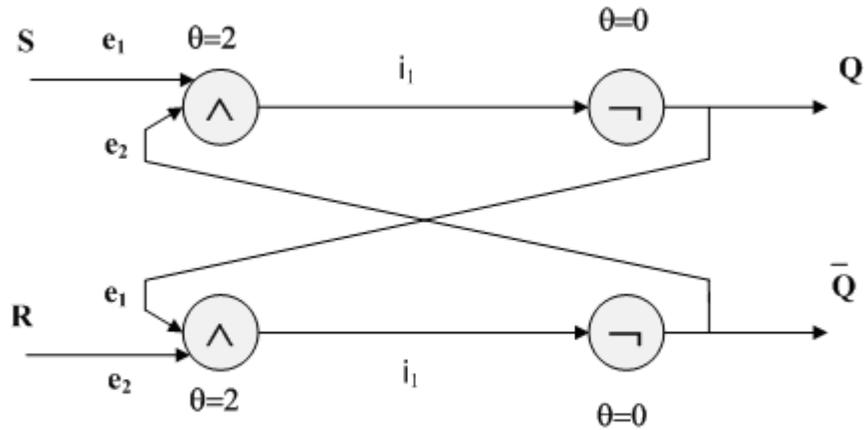


Fig. 8. Flip-Flop SR, representado con el modelo de neurona artificial de McCulloch-Pitts.

5.5 Flip - Flop (JK)

El FF JK es un refinamiento del FF SR. El FF JK supera el problema de obtener un estado indeterminado al presentarse el patrón de entrada 11. Las entradas J y K se comportan como las entradas S y R para ajustar y despejar el FF. Cuando se aplican las señales de entrada en forma simultánea tanto a J como a K, el FF cambia a su estado complementario, esto es, si $Q=1$, cambia a $Q=0$ y viceversa.

J	K	Q(t+1)	
0	0	Q(t)	Memoria
0	1	0	Reset
1	0	1	Set
1	1	Q'(t)	Complementario

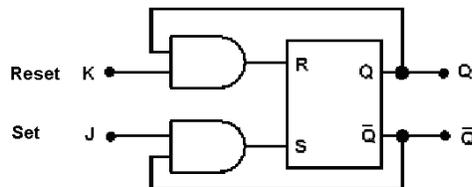


Fig. 9. Diagrama esquemático del Flip-Flop JK.

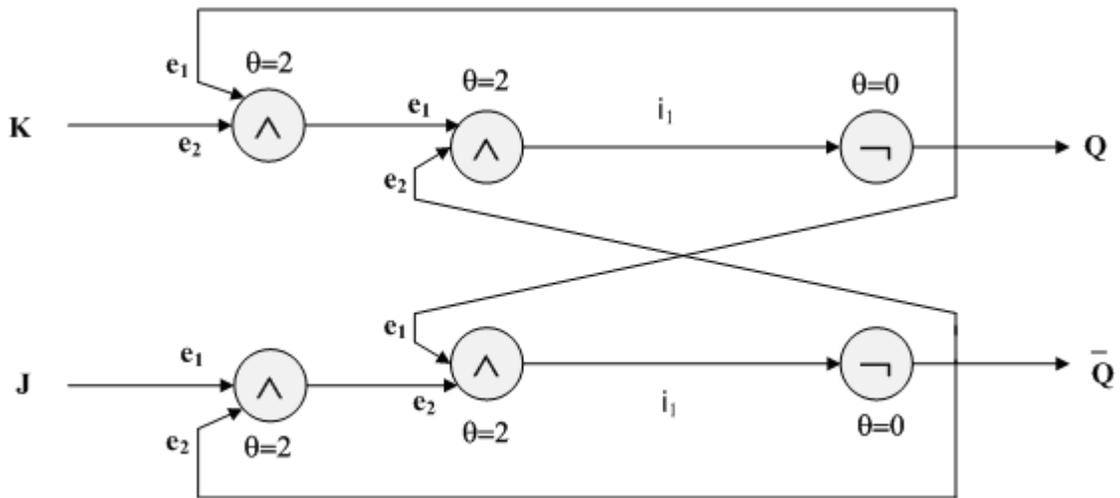


Fig. 10. Flip-Flop JK, representado con el modelo de neurona artificial de McCulloch-Pitts.

5.6 Flip-Flop (D)

El FF D recibe esta denominación por su capacidad de transferir (en un instante de tiempo posterior) los datos presentes de la entrada a la salida. En forma general, este FF puede ser implementado a partir de un FF RS con un inversor en la entrada R; el inversor agregado reduce el número de entradas de 2 a 1. La entrada D va en forma directa a la entrada S y su complemento a la entrada R.

Un Flip Flop D también se le llama FF transparente, porque refleja el dato que se presente a la entrada D inmediatamente a la salida Q, es decir si se coloca un 0 en la entrada D, no importa el valor de Q, esta tomará el valor de cero, y viceversa si se le aplica un 1, la salida Q irá inmediatamente a 1.

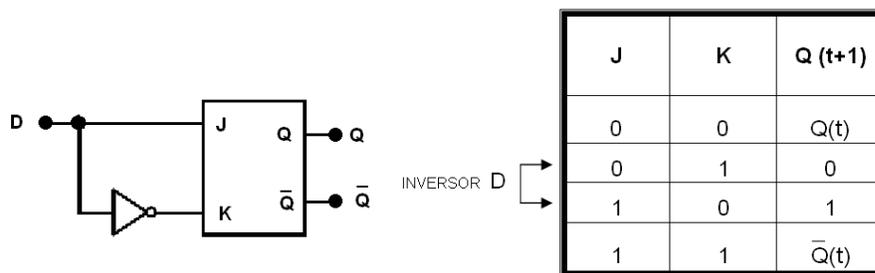


Fig. 11. Diagrama del FF tipo D, a partir del Flip-Flop tipo JK.

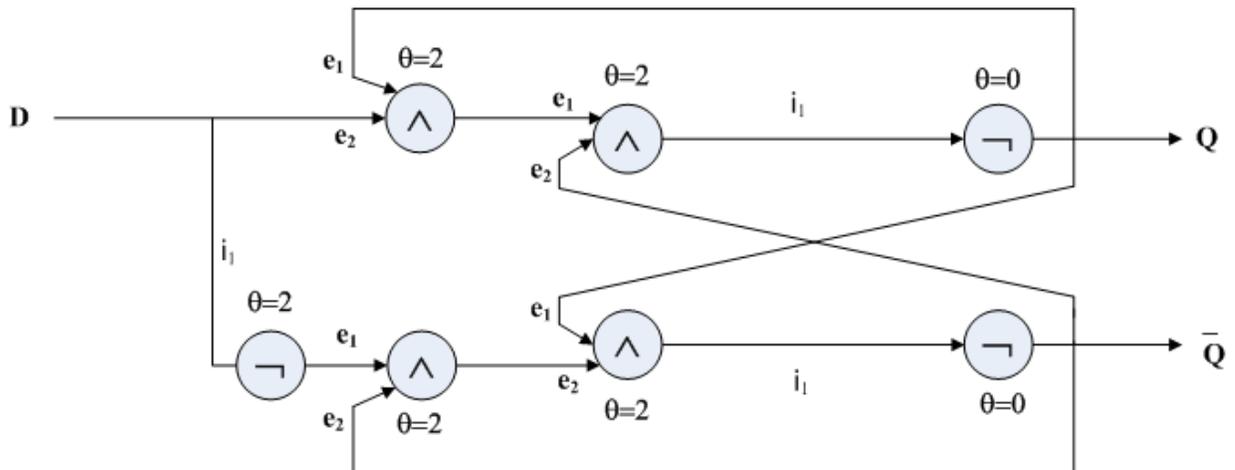


Fig. 12. Flip-Flop D, representado con el modelo de neurona artificial de McCulloch-Pitts.

5.7 Flip - Flop (T)

El FF T es una versión de una sola entrada del FF JK, el FF T se obtiene mediante un FF tipo JK si ambas entradas se conectan en corto. La denominación T proviene de la capacidad del FF para conmutar (toggle) o cambiar de estado. Sin importar el estado presente del FF, asume el estado complementario cuando ocurre el pulso de reloj mientras la entrada T es lógica 1.

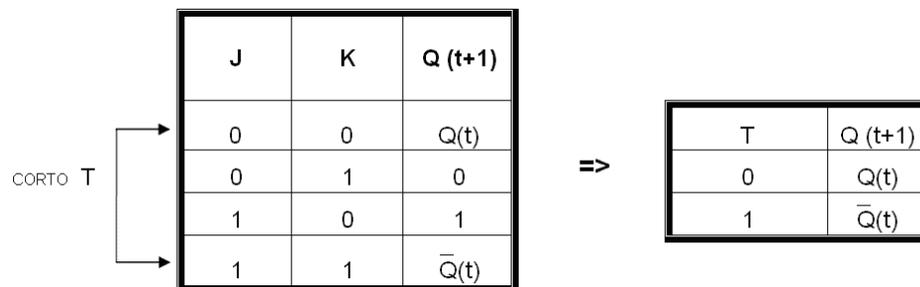


Fig. 13. Obtención del FF tipo T, a partir del Flip-Flop tipo JK.

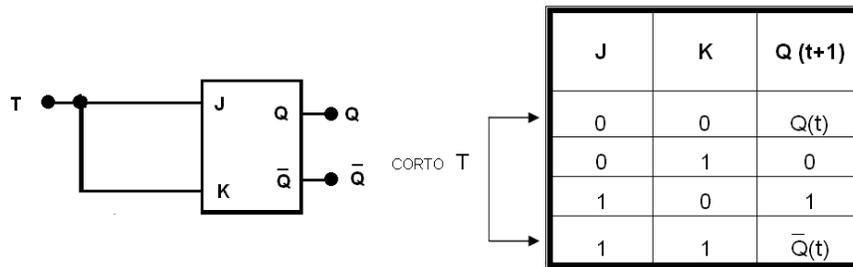


Fig. 14. Diagrama del FF tipo T, a partir del Flip-Flop tipo JK.

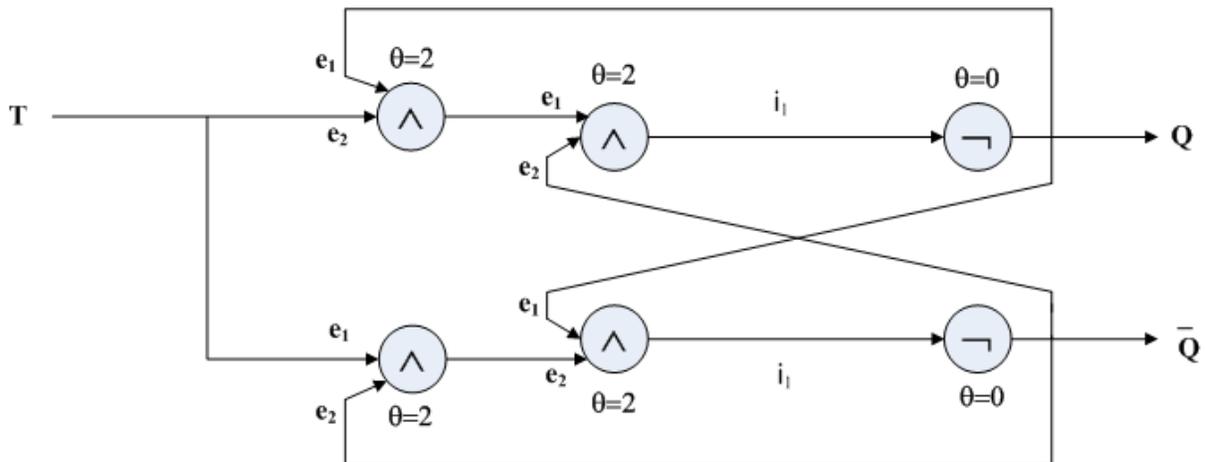


Fig. 15. Flip-Flop T, representado con el modelo de neurona artificial de McCulloch-Pitts.

6. Epílogo

En este informe técnico se presentó el modelo de neurona artificial de McCulloch- Pitts de una manera sencilla, y resumida. El objetivo fundamental de este trabajo es brindar al lector una perspectiva más amplia sobre los elementos que constituyen una red neuronal artificial. A través de los ejemplos presentados en la sección 5, se muestra como implementar las funciones AND, OR y NOT mediante la representación de neuronas artificiales, así mismo, se presentaron tanto los diagramas a bloques, como esquemáticos y representación bajo el modelo de neurona artificial, de los principales tipos flip-flop. Con todo lo anteriormente expuesto, se pretende que el amable lector obtenga la suficiente motivación para continuar investigando sobre los avances en el cam-

po de las redes neuronales artificiales. Lo invito a leer el *Perceptron*, (primera máquina que aprende, desarrollada en el año de 1957 por Frank Rossemblat); ADALINE (Adaptative Linear Neurons, desarrollado por Wildrow – Hoff presentado en el año de 1960 y MADALINE 1962); Backpropagation (Williams, Hinton y Rumelhart 1986).

Como trabajo futuro, a partir de los ejemplos presentados, se pueden realizar con la misma temática cualquier circuito que pueda ser representado como un conjunto de expresiones booleanas.

7. Referencias

- [1] I. Aleksander and H. Morton, *An introduction to neural computing*. New York, NY, USA: Van Nostrand Reinhold Co., 1990.
- [2] I. Aleksander, "Weightless neural tools: towards cognitive macrostructures," *Neural networks: theory and applications*, pp. 1-21, 1991.
- [3] I. Aleksander, *How to Build a Mind: Toward Machines with Imagination*. New York, NY, USA: Columbia University Press, 2001.
- [4] M. A. Arbib, *The Handbook of brain theory and neural networks*. Cambridge, Mass: London: MIT Press, 1995.
- [5] R. Beale and T. Jackson, *Neural computing: an introduction*. Bristol. U.K.: Institute of Physics Publishing. London, 1990.
- [6] M. I. Jordan and C. M. Bishop, "Neural networks," *Massachusetts Institute of Technology, Cambridge, MA, USA, AIM-1562*, 1996.
- [7] D. E. Rumelhart, *Introduction to human information processing*. New York: Wiley, 1977.
- [8] M. Tajine and D. Elizondo, "The recursive deterministic perceptron neural network," *Neural Networks*, vol. 11, no. 9, pp. 1571-1588, 1998.
- [9] D.J. Amit, *Modeling Brain Function*, Cambridge University Press, 1989.
- [10] G. A. Carpenter and S. Grossber "ART Adaptative Pattern Recognition by Self-organizing Neural Networks", *IEEE Computer*, vol. 21, pp. 77-90, 1988.
- [11] M. Chester, *Neural Networks: A Tutorial*, PTR Prentice Hall, Englewood Cliffs, NJ, 1993.
- [12] K. Gurney, *An Introduction to Neural Networks*, Routledge, Taylor & Francis Group, London, 1999.
- [13] M. T. Hagan, H. B. Demuth, and M. Beale, *Neural Network Design*, PWS Publishing Company, Boston, 1996.
- [14] S. Hayking, *Neural Networks: A Comprehensive foundation*, Macmillan College Publishing Company, New York, 1994.
- [15] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the Theory of Neural Computing*, Addison –Wesley Publishing Company, New York, 1991.
- [16] Warren S. McCulloch and Walter Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133. 1943.