ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/320414178

Implementación de la Función XOR con un Perceptrón en Arduino

Article · October 2017

CITATIONS
0

READS
667

2 authors:

Isra Cero
ITSPR
41 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Jesús Sevilla
Universidad Politécnica de Puebla
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project Implementación de la Función XOR con un Perceptrón en Arduino. View project

Project Entrenamiento función XOR View project

All content following this page was uploaded by Jesús Sevilla on 16 October 2017.

The user has requested enhancement of the downloaded file.

Implementación de la Función XOR con un Perceptrón en Arduino.

Israel.Cerón-Morales[1]

Entrenamiento de neurona (código y cálculos) Anexos: Jesús Sevilla Flores

Universidad Politécnica de Puebla
3er. Carril del ejido serrano s/n, San Mateo Cuanalá, Puebla, México. C.P. 72640

Israel.ceron@uppuebla.edu.mx

***Resumen*- Este documento presenta un ejemplo que permite implementar un Perceptron, este desarrollo se presenta muy pocas veces en la literatura relacionada con la inteligencia artificial, por lo que con el fin de ayudar a los estudiantes en el entendimiento de este tema se realiza este documento. Los resultados muestran la implementación de un Perceptron con el cual se realiza la función lógica XOR.**

***Palabras Clave*- Algoritmo, Arduino, Entrenamiento, Perceptron.**

INTRODUCCIÓN

El Perceptrón es una de las arquitecturas más básicas en el desarrollo de la inteligencia artificial y el entrenamiento de este tipo de neuronas artificiales suele ser poco detallado en la mayoría de los textos que bordan este tópico, muchas veces el entrenamiento de los perceptrones suele ser realizado con programas de computo, sin embargo existe interés por estudiantes que desean saber cuál es el paso a paso en el entrenamiento de un Perceptron. Por lo cual se da un ejemplo en este ayors s.

ANTECEDENTES

Trabajar con inteligencia artificial hoy día es muy común por parte de estudiantes universitarios y muchas veces el entrenamiento de redes neuronales se vuelve difícil de asimilar debido a que no siempre se tiene practica en el entrenamiento de redes neuronales paso a paso y esto es debido a la gran cantidad de aritmética empleada, por lo que en este ayors s se desarrolla un ejemplo básico de entrenamiento lo que permite a los estudiantes interesados tener un punto de referencia de que es lo que realmente hacen los programas de computo al momento de entrenar redes neuronales artificiales tipo Perceptrón.

DESARROLLO

Consideremos un Perceptrón con dos entradas y dos pesos sinápticos, además de un valor de predisposición llamado BIAS (B) como lo muestra la Figura 1.

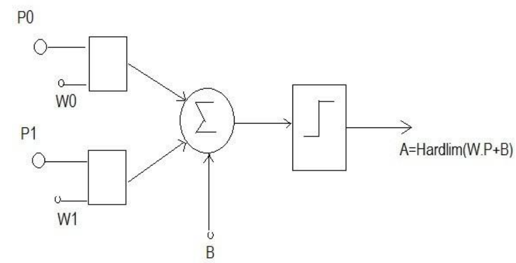


Figura. 1. Diagrama de un Perceptron con dos entradas y un valor de predisposición B.

La función limitador fuerte (Hardlim) se define en la ecuación Ec.1.

$$A = \begin{cases} 1 & \vec{W}\cdot\bar{P} + B \geq 0 \\ 0 & \vec{W}\cdot\bar{P} + B < 0 \end{cases} \qquad \text{Ec.1}$$

En donde:

$$\vec{W} = [W0 \quad W1] \qquad \text{Ec.2}$$

$$\bar{P} = \begin{bmatrix} P0 \\ P1 \end{bmatrix} \qquad \text{Ec.3}$$

La ecuación Ec.2 se llama la matriz de pesos sinápticos, en este ejemplo es una matriz de 1 fila y 2 columnas, y la ecuación Ec.3 si es un vector vertical de 2 filas. De manera que el ayors de la matriz W y el vector P tiene como resultado un escalar. Además el valor de predisposición o BIAS (B) es un escalar, de manera que el término $\vec{W}\cdot\bar{P} + B$ es un escalar. Sobre este escalar se aplica la definición de Ec.1.

Se procede con el algoritmo de entrenamiento mediante el cual se encuentran los valores de $\vec{W}$ y B con los cuales la red neuronal cumple con la función deseada.

P0	P1	AND	OR	XOR
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Tabla. 1. Tabla que muestra las entradas Po y P1 y las funciones lógicas a ser utilizadas para programar un Perceptron XOR.

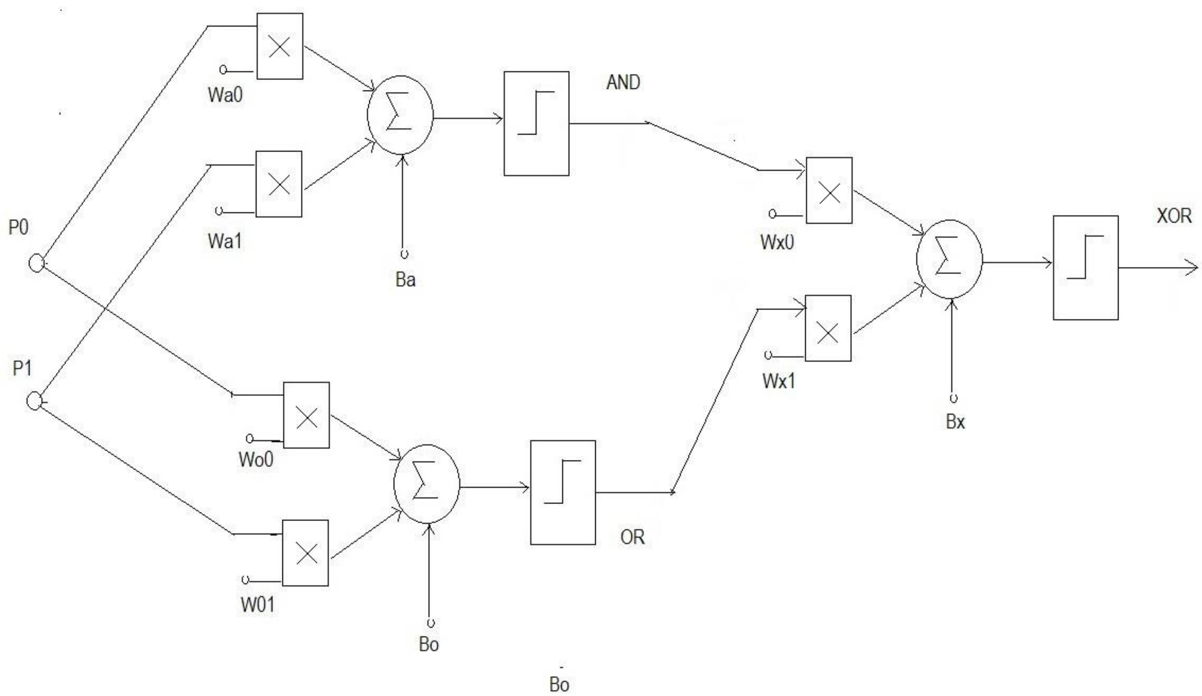


Figura. 2. Diagrama de un Perceptron con tres neuronas que desempeñan la función XOR.

La Tabla 1 muestra las entradas P0 y P1 que serán procesadas por el Perceptron, para obtener la función XOR (OR Exclusiva) es necesario hacer uso de dos funciones de apoyo, debido a que la función XOR no se puede implementar con una única neurona de tipo Perceptron. Es necesario ayors s dos neuronas una que desempeñe la función AND y otra la función OR y con estas saludad entrenar una tercera neurona para finalmente llegar al objetivo de tener una neurona Perceptron. XOR.

Después de ayors s el entrenamiento de las neuronas se tienen los valores de los pesos sinápticos y los valores de las predisposiciones para cada neurona. Con los valores calculados se desarrolla el código de programación que se observa en el Cuadro 1.

En la Figura 3 Se observa la fotografía de la implementación del Perceptron en una tarjeta de desarrollo Arduino Uno.

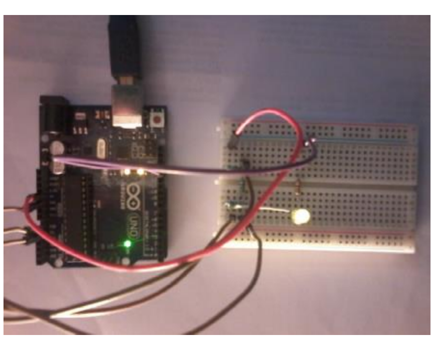
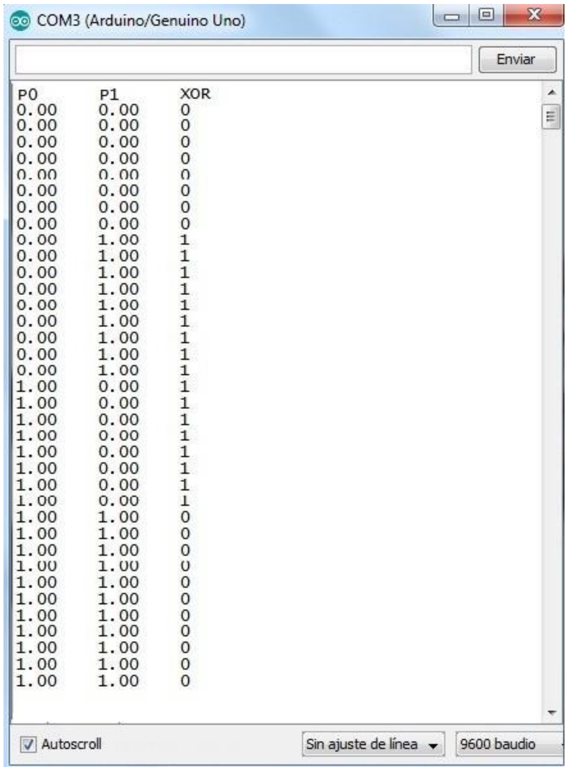


Figura. 3. Implementación en Arduino del Perceptron que realiza la función XOR.

```
float analogPin0 = 0; float analogPin1 = 1; int axon = 13;


float p0; float p1; int k =0;

int wa0 = 2; int wa1 = 1; int ba = -3;
int wo0 = 1; int wo1 = 1; int bo = -1;
int wxa = -3; int wxo = 1; int bx = -1;
int fand; int ffor; int a;


void setup() {
pinMode(11,OUTPUT);
pinMode(axon, OUTPUT);
Serial.begin(9600); }


void loop() {
while(k<=200) { for(k=0;
k<=200; k++) {

p0=(analogRead(0))/384; p1=(analogRead(1))/384;


if( wa0*p0+wa1*p1+ba < 0)
fand =0;

else if ( wa0*p0+wa1*p1+ba >= 0)
fand =1;


if( wo0*p0+wo1*p1+bo <0)
```

Cuadro. 1. Código en Arduino del Perceptron que realiza la función XOR.

Finalmente en la Figura 4 vemos la salida del monitor Serial de Arduino en el cual se observa el comportamiento XOR programado con este Perceptron.



P0	P1	XOR
0.00	0.00	0
0.00	0.00	0
0.00	0.00	0
0.00	0.00	0
0.00	0.00	0
0.00	0.00	0
0.00	0.00	0
0.00	0.00	0
0.00	1.00	1
0.00	1.00	1
0.00	1.00	1
0.00	1.00	1
0.00	1.00	1
0.00	1.00	1
0.00	1.00	1
0.00	1.00	1
0.00	1.00	1
1.00	0.00	1
1.00	0.00	1
1.00	0.00	1
1.00	0.00	1
1.00	0.00	1
1.00	0.00	1
1.00	0.00	1
1.00	0.00	1
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0
1.00	1.00	0

Figura. 4. Salida del monitor Serial de Arduino que muestra el desempeño del Perceptron realizando la Función XOR.

RESULTADOS

Se obtuvo la función XOR mediante el entrenamiento de un Perceptron de tres neuronas, y la implementación fue ayors s en Arduino sin requerir demasiados accesorios, solo fueron necesarias unas resistencias y un Diodo LED para ayors s la salida, la cual está entre cero y uno lógico.

CONCLUSIONES

1.- Implementar un Perceptron en Arduino que desempeñe la función Lógica XOR es ayors s solo hay que ser cuidadosos en el desarrollo del entrenamiento de las neuronas que ayors s el Perceptron, Además se muestra el comportamiento de la función XOR en el monitor serial, las entradas P0 y P1 se muestran con puntos ayors s porque son entradas analógicas, la salida ayor ayors digital.

2.- Este Perceptron puede ser utilizado con ayors ventajas que las que ofrece la compuesta lógica XOR debido a que las entradas son analógicas y no tiene una zona de trabajo indeterminada como ocurre con las compuertas de la familia TTL y CMOS.

REFERENCIAS

[1] – Howard Demuth, Mark Beale, Neural Network Toolbox or Use with MATLAB, User Guide, Version 4, The MathWorks, Inc. 2002.

ANEXOS

(Código de entrenamiento para la función XOR y cálculos para las funciones AND y OR)

Entrenamiento para la función AND

Pa=$\begin{bmatrix}1\\0\end{bmatrix}$ Pb=$\begin{bmatrix}0\\1\end{bmatrix}$ Pc=$\begin{bmatrix}1\\0\end{bmatrix}$ Pd=$\begin{bmatrix}1\\1\end{bmatrix}$

Ta=0 Tb=0 Tc=0 Td=1 $w = [0 \quad 0]$ b=0

1)

a=H ($\left([0 \quad 0]\begin{bmatrix}1\\0\end{bmatrix} + 0\right)$

a=H (0*1+0*0+0)

a=H (0)

A=1 E=0-1 E=-1

$$w^{new} = [0 \quad 0] + (-1)[0 \quad 0]$$

$$w^{new} = [0 \quad 0] + [0 \quad 0]$$

$$w^{new} = [0 \quad 0]$$

$$b^{new} = 0 + (-1)$$

$$b^{new} = -1$$

2)

a=H ($\left([0 \quad 0]\begin{bmatrix}0\\1\end{bmatrix} + (-1)\right)$

a=H (0*0+0*1+(-1))

a=H (-1)

A=0 E=0-0 E=0

$$w^{new} = [0 \quad 0] + (0)[0 \quad 1]$$

$$w^{new} = [0 \quad 0] + [0 \quad 0]$$

$$w^{new} = [0 \quad 0]$$

$$b^{new} = (-1) + 0$$

$$b^{new} = -1$$

3)

a=H ($\left([0 \quad 0]\begin{bmatrix}1\\0\end{bmatrix} + (-1)\right)$

a=H (0*1+0*0+(-1))

a=H (-1)

A=0 E=0-0 E=0

$$w^{new} = [0 \quad 0] + (0)[1 \quad 0]$$

$$w^{new} = [0 \quad 0] + [0 \quad 0]$$

$$w^{new} = [0 \quad 0]$$

$$b^{new} = (-1) + 0$$

$$b^{new} = -1$$

4)

a=H ($\left([0 \quad 0]\begin{bmatrix}1\\1\end{bmatrix} + (-1)\right)$

a=H (0*1+0*1+(-1))

a=H (-1)

A=0 E=1-0 E=1

$$w^{new} = [0 \quad 0] + (1)[1 \quad 1]$$

$$w^{new} = [0 \quad 0] + [1 \quad 1]$$

$$w^{new} = [1 \quad 1]$$

$$b^{new} = (-1) + 1$$

$$b^{new} = 0$$

5)

a=H ($\left([1 \quad 1]\begin{bmatrix}0\\0\end{bmatrix}+0\right)$

a=H (1*0+1*0+0)

a=H (0)

A=1 E=0-1 E=1

$$w^{new}=[1 \quad 1]+(-1)[0 \quad 0]$$

$$w^{new}=[1 \quad 1]+[0 \quad 0]$$

$$w^{new}=[1 \quad 1]$$

$$b^{new}=0+(-1)$$

$$b^{new}=-1$$

6)

a=H ($\left([1 \quad 1]\begin{bmatrix}0\\1\end{bmatrix}+(-1)\right)$

a=H (1*0+1*1+(-1))

a=H (0)

A=1 E=0-1 E=1

$$w^{new}=[1 \quad 1]+(-1)[0 \quad 1]$$

$$w^{new}=[1 \quad 1]+[0 \quad -1]$$

$$w^{new}=[1 \quad 0]$$

$$b^{new}=-1+(-1)$$

$$b^{new}=-2$$

7)

a=H ($\left([1 \quad 0]\begin{bmatrix}1\\0\end{bmatrix}+(-2)\right)$

a=H (1*1+0*0+(-2))

a=H (-1)

A=0 E=0-0 E=0

$$w^{new}=[1 \quad 0]+(0)[0 \quad 0]$$

$$w^{new}=[1 \quad 0]+[0 \quad 0]$$

$$w^{new}=[1 \quad 0]$$

$$b^{new}=-2+0$$

$$b^{new}=-2$$

8)

a=H ($\left([1 \quad 0]\begin{bmatrix}1\\1\end{bmatrix}+(-2)\right)$

a=H (1*1+0*1+ (-2))

a=H (-1)

A=0 E=1-0 E=1

$$w^{new}=[1 \quad 0]+(1)[1 \quad 1]$$

$$w^{new}=[1 \quad 0]+[1 \quad 1]$$

$$w^{new}=[2 \quad 0]$$

$$b^{new}=-2+1$$

$$b^{new}=-1$$

9)

a=H $\left(\begin{bmatrix}2 & 0\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}+(-1)\right)$

a=H (2*0+0*0+ (-1))

a=H (-1)

A=0 E=0-0 E=0

$w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}+(0)\begin{bmatrix}0 & 0\end{bmatrix}$

$w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}+\begin{bmatrix}0 & 0\end{bmatrix}$

$w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}$

$b^{new}=-1+0$

$b^{new}=-1$

10)

a=H $\left(\begin{bmatrix}2 & 1\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}+(-1)\right)$

a=H (2*0+1*1+ (-1))

a=H (0)

A=1 E=0-1 E=-1

$w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}+(-1)\begin{bmatrix}0 & 1\end{bmatrix}$

$w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}+\begin{bmatrix}0 & -1\end{bmatrix}$

$w^{new}=\begin{bmatrix}2 & 0\end{bmatrix}$

$b^{new}=-1+(-1)$

$b^{new}=-2$

11)

a=H $\left(\begin{bmatrix}2 & 0\end{bmatrix}\begin{bmatrix}1\\0\end{bmatrix}+(-2)\right)$

a=H (2*1+0*0+ (-2))

a=H (0)

A=1 E=0-1 E=-1

$w^{new}=\begin{bmatrix}2 & 0\end{bmatrix}+(-1)\begin{bmatrix}1 & 0\end{bmatrix}$

$w^{new}=\begin{bmatrix}2 & 0\end{bmatrix}+\begin{bmatrix}-1 & 0\end{bmatrix}$

$w^{new}=\begin{bmatrix}1 & 0\end{bmatrix}$

$b^{new}=-2-1$

$b^{new}=-3$

12)

a=H $\left(\begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}+(-3)\right)$

a=H (1*1+0*1+ (-3))

a=H (-2)

A=0 E=1-0 E=1

$w^{new}=\begin{bmatrix}1 & 0\end{bmatrix}+(1)\begin{bmatrix}1 & 1\end{bmatrix}$

$w^{new}=\begin{bmatrix}1 & 0\end{bmatrix}+\begin{bmatrix}1 & 1\end{bmatrix}$

$w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}$

$b^{new}=-3+1$

$b^{new}=-2$

13)

a=H $\left([2 \quad 1]\begin{bmatrix}0\\0\end{bmatrix} + (-2)\right)$

a=H (2*0+1*0+ (-2))

a=H (-2)

A=0 E=0-0 E=0

$$w^{new} = [2 \quad 1] + (0)[0 \quad 0]$$

$$w^{new} = [2 \quad 1] + [0 \quad 0]$$

$$w^{new} = [2 \quad 1]$$

$$b^{new} = -2 + 0$$

$$b^{new} = -2$$

14)

a=H $\left([2 \quad 1]\begin{bmatrix}0\\1\end{bmatrix} + (-2)\right)$

a=H (2*0+1*1+ (-2))

a=H (-1)

A=0 E=0-0 E=0

$$w^{new} = [2 \quad 1] + (0)[0 \quad 1]$$

$$w^{new} = [2 \quad 1] + [0 \quad 0]$$

$$w^{new} = [2 \quad 1]$$

$$b^{new} = -2 + 0$$

$$b^{new} = -2$$

15)

a=H $\left([2 \quad 1]\begin{bmatrix}1\\0\end{bmatrix} + (-2)\right)$

a=H (2*1+1*0 (-2))

a=H (0)

A=1 E=0-1 E=-1

$$w^{new} = [2 \quad 1] + (-1)[1 \quad 0]$$

$$w^{new} = [2 \quad 1] + [-1 \quad 0]$$

$$w^{new} = [1 \quad 1]$$

$$b^{new} = -2 + (-1)$$

$$b^{new} = -3$$

16)

a=H $\left([1 \quad 1]\begin{bmatrix}1\\1\end{bmatrix} + (-3)\right)$

a=H (1*1+1*1+ (-3))

a=H (-1)

A=0 E=1-0 E=1

$$w^{new} = [1 \quad 1] + (1)[1 \quad 1]$$

$$w^{new} = [1 \quad 1] + [1 \quad 1]$$

$$w^{new} = [2 \quad 2]$$

$$b^{new} = -3 + 1$$

$$b^{new} = -2$$

17)

a=H $\left(\begin{bmatrix}2 & 2\end{bmatrix}\begin{bmatrix}0\\0\end{bmatrix}+(-2)\right)$

a=H (2*0+2*0 +(-2))

a=H (-2)

A=0 E=0-0 E=0

$$w^{new}=\begin{bmatrix}2 & 2\end{bmatrix}+(0)\begin{bmatrix}0 & 0\end{bmatrix}$$

$$w^{new}=\begin{bmatrix}2 & 2\end{bmatrix}+\begin{bmatrix}0 & 0\end{bmatrix}$$

$$w^{new}=\begin{bmatrix}2 & 2\end{bmatrix}$$

$$b^{new}=-2+0$$

$$b^{new}=-2$$

18)

a=H ($\left(\begin{bmatrix}2 & 2\end{bmatrix}\begin{bmatrix}0\\1\end{bmatrix}+(-2)\right)$

a=H (2*0+2*1+ (-2))

a=H (0)

A=1 E=0-1 E=-1

$$w^{new}=\begin{bmatrix}2 & 2\end{bmatrix}+(-1)\begin{bmatrix}0 & 1\end{bmatrix}$$

$$w^{new}=\begin{bmatrix}2 & 2\end{bmatrix}+\begin{bmatrix}0 & -1\end{bmatrix}$$

$$w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}$$

$$b^{new}=-2+(-1)$$

$$b^{new}=-3$$

Los valores finales para la función AND son:

$$\boldsymbol{w^{new}=\begin{bmatrix}2 & 1\end{bmatrix}} \qquad \boldsymbol{b^{new}=-3}$$

Entrenamiento para la función OR

Pa=$\begin{bmatrix}0\\0\end{bmatrix}$ Pb=$\begin{bmatrix}0\\1\end{bmatrix}$ Pc=$\begin{bmatrix}1\\0\end{bmatrix}$ Pd=$\begin{bmatrix}1\\1\end{bmatrix}$

Ta=0 Tb=1 Tc=1 Td=1 $w = [0 \quad 0]$ b=0

1)

a=H ($\left([0 \quad 0]\begin{bmatrix}0\\0\end{bmatrix} + 0\right)$

a=H (0*0+0*0+0)

a=H (0)

A=1 E=0-1 E=-1

$w^{new} = [0 \quad 0] + (-1)[0 \quad 0]$

$w^{new} = [0 \quad 0] + [0 \quad 0]$

$w^{new} = [0 \quad 0]$

$b^{new} = 0 + (-1)$

$b^{new} = -1$

2)

a=H ($\left([0 \quad 0]\begin{bmatrix}0\\1\end{bmatrix} + 0\right)$

a=H (0*0+0*0+(-1)

a=H (-1)

A=0 E=1-0 E=1

$w^{new} = [0 \quad 0] + (1)[0 \quad 1]$

$w^{new} = [0 \quad 0] + [0 \quad 1]$

$w^{new} = [0 \quad 1]$

$b^{new} = (-1) + 1$

$b^{new} = 0$

3)

a=H ($\left([0 \quad 1]\begin{bmatrix}1\\0\end{bmatrix} + 0\right)$

a=H (0*1+1*0+0)

a=H (0)

A=1 E=1-1 E=0

$w^{new} = [0 \quad 1] + (0)[1 \quad 0]$

$w^{new} = [0 \quad 1] + [0 \quad 0]$

$w^{new} = [0 \quad 1]$

$b^{new} = 0 + 0$

$b^{new} = 0$

4)

a=H ($\left([0 \quad 1]\begin{bmatrix}1\\1\end{bmatrix} + 0\right)$

a=H (0*1+1*1+0)

a=H (1)

A=1 E=1-1 E=0

$w^{new} = [0 \quad 1] + (0)[1 \quad 1]$

$w^{new} = [0 \quad 1] + [0 \quad 0]$

$w^{new} = [0 \quad 1]$

$b^{new} = 0 + 0$

$b^{new} = 0$

5)

a=H ($\left([0 \quad 1]\begin{bmatrix}0\\0\end{bmatrix} + 0\right)$

a=H (0*0+1*0+0)

a=H (0)

A=0 E=0-1 E=-1

$$w^{new} = [0 \quad 1] + (0)[0 \quad 0]$$

$$w^{new} = [0 \quad 1] + [0 \quad 0]$$

$$w^{new} = [0 \quad 1]$$

$$b^{new} = 0 + (-1)$$

$$b^{new} = -1$$

6)

a=H ($\left([0 \quad 1]\begin{bmatrix}0\\1\end{bmatrix} + (-1)\right)$

a=H (0*0+1*1+ (-1))

a=H (0)

a=1 E=1-1 E=0

$$w^{new} = [0 \quad 1] + (0)[0 \quad 0]$$

$$w^{new} = [0 \quad 1] + [0 \quad 0]$$

$$w^{new} = [0 \quad 1]$$

$$b^{new} = (-1) + 0$$

$$b^{new} = -1$$

7)

a=H ($\left([0 \quad 1]\begin{bmatrix}1\\0\end{bmatrix} + (-1)\right)$

a=H (0*1+1*0+ (-1))

a=H (-1)

a=0 E=1-0 E=1

$$w^{new} = [0 \quad 1] + (1)[1 \quad 0]$$

$$w^{new} = [0 \quad 1] + [1 \quad 0]$$

$$w^{new} = [1 \quad 1]$$

$$b^{new} = (-1) + 1$$

$$b^{new} = 0$$

8)

a=H ($\left([1 \quad 1]\begin{bmatrix}1\\1\end{bmatrix} + 0\right)$

a=H (1*1+1*1+ 0)

a=H (2)

a=1 E=1-1 E=0

$$w^{new} = [1 \quad 1] + (0)[1 \quad 1]$$

$$w^{new} = [1 \quad 1] + [0 \quad 0]$$

$$w^{new} = [1 \quad 1]$$

$$b^{new} = 0 + 0$$

$$b^{new} = 0$$

9)

$$a=H\left(\left([1\quad 1]\begin{bmatrix}0\\0\end{bmatrix}+0\right)\right.$$

a=H (1*0+1*0+ 0)

a=H (0)

a=1 E=0-1 E=-1

$$w^{new}=[1\quad 1]+(-1)[0\quad 0]$$

$$w^{new}=[1\quad 1]+[0\quad 0]$$

$$w^{new}=[1\quad 1]$$

$$b^{new}=0+(-1)$$

$$b^{new}=-1$$

Los valores finales para la función OR son:

$$\boldsymbol{w^{new}=[1\quad 1]}\qquad\qquad \boldsymbol{b^{new}=-1}$$

Entrenamiento para la función XOR

A continuación se adjunta el código para el entrenamiento de la neurona en Arduino:

```
int r,i,s;

int xa1 = 0, xa2=0, ta=0, xb1=0,xb2=1, tb=1, xc1= 1,
xc2=0, tc=1, xd1=1, xd2=1, td=0;

int wa0 = 2, wa1 = 1, ba = -3; //Valores funcion AND

int we0 = 1,we1 = 1, be = -1; // Valores funcion OR

int w1=0, w2=0, b=0;

float e=0, a=0, axon=0;

void setup() {

Serial.begin(9600);

}

void loop() {
```

```
while(r<10){

 Serial.println();

 Serial.print("r");

 Serial.print('\t');

 Serial.print("axn");

 Serial.print('\t');

 Serial.print("w1");

 Serial.print('\t');

 Serial.print("w2");

 Serial.print('\t');
```

```
 Serial.print("b");
 Serial.print('\t');
 for(r=0;r<=10;r++){
axon=(((w1*xa1)+(w2*xa2)+b)+((wa0*xa1)+(wa1*xa2)
+ba));
 if(axon>=0){
 a=1;}
 else if(axon<0){
a=0;}
e=ta-a;
w1=(wa0+e*xa1);
w2=(wa1+e*xa2);
b=b+e;
 Serial.println();
 Serial.print(r);
 Serial.print('\t');
 Serial.print(a);
 Serial.print('\t');
 Serial.print(w1);
 Serial.print('\t');
 Serial.print(w2);
 Serial.print('\t');
 Serial.print(b);
 Serial.print('\t');

if(((w1*xb1)+(w2*xb2)+b)+((wa0*xb1)+(wa1*xb2)+ba)
>=0)
 a=1;
 else
if(((w1*xb1)+(w2*xb2)+b)+((wa0*xb1)+(wa1*xb2)+ba)
<0)
 a=0;
e=tb-a;
w1=(w1+e*xb1);
w2=w2+e*xb2;
b=b+e;
 Serial.println();
 Serial.print(r);
 Serial.print('\t');
 Serial.print(a);
 Serial.print('\t');
 Serial.print(w1);
 Serial.print('\t');
 Serial.print(w2);
 Serial.print('\t');
 Serial.print(b);
 Serial.print('\t');
if(((w1*xb1)+(w2*xb2)+b)+((wa0*xc1)+(wa1*xc2)+ba)>
=0)
 a=1;
 else
if(((w1*xb1)+(w2*xb2)+b)+((wa0*xc1)+(wa1*xc2)+ba)<
0)
 a=0;
e=tc-a;
w1=w1+e*xc1;
w2=w2+e*xc2;
b=b+e;
 Serial.println();
 Serial.print(r);
 Serial.print('\t');
 Serial.print(a);
 Serial.print('\t');
 Serial.print(w1);
 Serial.print('\t');
```

```
 Serial.print(w2);

 Serial.print('\t');

 Serial.print(b);

 Serial.print('\t');

if(((w1*xb1)+(w2*xb2)+b)+((wa0*xd1)+(wa1*xd2)+ba)
>=0)

 a=1;

 else
if(((w1*xb1)+(w2*xb2)+b)+((wa0*xd1)+(wa1*xd2)+ba)
<0)

 a=0;

e=td-a;

w1=w1+e*xd1;

w2=w2+e*xd2;

b=b+e;

 }

 for(i=0;i<=10;i++){

axon=(((w1*xa1)+(w2*xa2)+b)+((we0*xa1)+(we1*xa2)
+be));

 if(axon>=0){

 a=1;}

 else if(axon<0){

 a=0;}

e=ta-a;

w1=(w1+e*xa1);

w2=(w2+e*xa2);

b=b+e;

 Serial.println();

 Serial.print(i);

 Serial.print('\t');

 Serial.print(a);

 Serial.print('\t');

 Serial.print(w1);

 Serial.print('\t');

 Serial.print(w2);

 Serial.print('\t');

 Serial.print(b);

 Serial.print('\t');


if(((w1*xb1)+(w2*xb2)+b)+((we0*xb1)+(we1*xb2)+be)
>=0)

 a=1;

 else
if(((w1*xb1)+(w2*xb2)+b)+((we0*xb1)+(we1*xb2)+be)
<0)

 a=0;

e=tb-a;

w1=(w1+e*xb1);

w2=w2+e*xb2;

b=b+e;

 Serial.println();

 Serial.print(i);

 Serial.print('\t');

 Serial.print(a);

 Serial.print('\t');

 Serial.print(w1);

 Serial.print('\t');

 Serial.print(w2);

 Serial.print('\t');

 Serial.print(b);

 Serial.print('\t');

if(((w1*xb1)+(w2*xb2)+b)+((we0*xc1)+(we1*xc2)+be)>
=0)

 a=1;

 else
if(((w1*xb1)+(w2*xb2)+b)+((we0*xc1)+(we1*xc2)+be)<
0)
```

```
 a=0;
e=tc-a;
w1=w1+e*xc1;
w2=w2+e*xc2;
b=b+e;
Serial.println();
 Serial.print(i);
 Serial.print('\t');
 Serial.print(a);
 Serial.print('\t');
 Serial.print(w1);
 Serial.print('\t');
 Serial.print(w2);
 Serial.print('\t');
 Serial.print(b);
 Serial.print('\t');
if(((w1*xb1)+(w2*xb2)+b)+((we0*xd1)+(we1*xd2)+be)
>=0)
 a=1;
 else
if(((w1*xb1)+(w2*xb2)+b)+((we0*xd1)+(we1*xd2)+be)
<0)
 a=0;
e=td-a;
w1=w1+e*xd1;
w2=w2+e*xd2;
b=b+e;
 }
 for(s=0;s<=10;s++){
axon=(((w1*xa1)+(w2*xa2)+b));
 if(axon>=0){
 a=1;}
 else if(axon<0){
 a=0;}
e=ta-a;
w1=(w1+e*xa1);
w2=(w2+e*xa2);
b=b+e;
 Serial.println();
 Serial.print(p);
 Serial.print('\t');
 Serial.print(a);
 Serial.print('\t');
 Serial.print(w1);
 Serial.print('\t');
 Serial.print(w2);
 Serial.print('\t');
 Serial.print(b);
 Serial.print('\t');

if(((w1*xb1)+(w2*xb2)+b)>=0)
 a=1;
 else if(((w1*xb1)+(w2*xb2)+b)<0)
 a=0;
e=tb-a;
w1=(w1+e*xb1);
w2=w2+e*xb2;
b=b+e;
 Serial.println();
 Serial.print(s);
 Serial.print('\t');
 Serial.print(a);
 Serial.print('\t');
```

```
Serial.print(w1);
Serial.print('\t');
Serial.print(w2);
Serial.print('\t');
Serial.print(b);
Serial.print('\t');
if(((w1*xb1)+(w2*xb2)+b)>=0)
a=1;
else if(((w1*xb1)+(w2*xb2)+b)<0)
a=0;
e=tc-a;
w1=w1+e*xc1;
w2=w2+e*xc2;
b=b+e;
Serial.println();
Serial.print(s);
Serial.print('\t');
Serial.print(a);
Serial.print('\t');
Serial.print(w1);
Serial.print('\t');
Serial.print(w2);
Serial.print('\t');
Serial.print(b);
Serial.print('\t');
if(((w1*xb1)+(w2*xb2)+b)>=0)
a=1;
else if(((w1*xb1)+(w2*xb2)+b)<0)
a=0;
e=td-a;
w1=w1+e*xd1;
w2=w2+e*xd2;
b=b+e;
}
}
}
```

Después de realizar el entrenamiento de las neuronas se tienen los valores de los pesos sinápticos y los valores de las predisposiciones para cada neurona. Como se observa en la siguiente figura:

n	axn	w1	w2	b
0	1.00	0	0	-1
0	0.00	0	1	0
0	1.00	0	1	0
1	1.00	0	1	-1
1	1.00	0	1	-1
1	0.00	1	1	0
2	1.00	1	1	-1
2	1.00	1	1	-1
2	1.00	1	1	-1
3	0.00	1	1	-1
3	1.00	1	1	-1
3	1.00	1	1	-1
4	0.00	1	1	-1
4	1.00	1	1	-1
4	1.00	1	1	-1
5	0.00	1	1	-1
5	1.00	1	1	-1
5	1.00	1	1	-1
6	0.00	1	1	-1
6	1.00	1	1	-1
6	1.00	1	1	-1
7	0.00	1	1	-1
7	1.00	1	1	-1
7	1.00	1	1	-1
8	0.00	1	1	-1
8	1.00	1	1	-1
8	1.00	1	1	-1
9	0.00	1	1	-1
9	1.00	1	1	-1
9	1.00	1	1	-1
10	1.00	-1	2	-2
10	1.00	-1	1	-1
10	1.00	-2	1	-1

Los valores finales para la función XOR son:

$$\boldsymbol{w^{new} = [-2 \quad 1]} \qquad \boldsymbol{b^{new} = -1}$$

View publication stats