# EL PERCEPTRÓN ANALÓGICO

Universidad Politécnica Salesiana, Ingeniería Electrónica,

Julio Cajamarca, Diego Idrovo, Juan Andrés Valverde, 8 de Diciembre de 2009.

**Abstract.** The present work aims to design and implementation of an analog neuron, their behavior as such, and obtaining data to make it behave in a predetermined form.

**Resumen.** El presente trabajo, tiene como objetivo el diseño e implementación de una neurona analógica, su comportamiento como tal, y la obtención de datos para hacer que se comporte de una manera predeterminada.

***Palabras clave***: Perceptrón, red neuronal artificial (RNA), peso sináptico, bias, función de correspondencia.

## 1 Introducción

El perceptrón fue la primera neurona artificial en aparecer, esto fue por los años 60 gracias a Frank Rosenblatt el cual diseño esta neurona con una capacidad de aprender y reconocer patrones sencillos, los cuales estarán claramente identificados en el momento de tomar una decisión, pues actuara según datos previamente aprendidos.

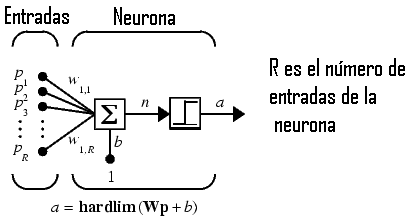
## 2 El Perceptrón

Un perceptrón es una neurona artificial que tiene como salida la suma ponderada de todas sus entradas para luego ser comparadas con un umbral.

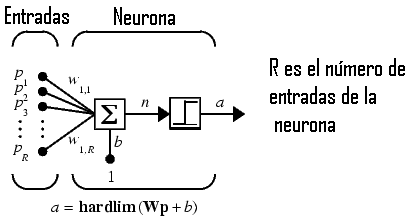
Se caracterizan porque tienen un aprendizaje supervisado. Un perceptrón puede ser implementado en software o hardware, y únicamente tienen la capacidad de separar datos de naturaleza lineal.

### 2.1 Modelo de Perceptrón

El modelo matemático de la neurona artificial, es el mostrado en la figura 1.



**Figura 1.** Modelo de perceptrón

Donde Pij son las entradas de una neurona, Wij son pesos sinápticos de la i-esima neurona, b es el nivel de disparo de la i-ésima neurona y el bloque  función de activación o función de transferencia.

### 2.2 Construyendo un Perceptrón

Un perceptrón en hardware, es básicamente representado por un amplificador operacional configurado como sumador, esto se muestra en la figura 2



**Figura 2.** Perceptrón analógico

Donde:

Como podemos ver en la ecuación del sumador, representa los pesos sinápticos, mientras que Pi representa las dendritas o entradas de la neurona artificial.

## 3 Desarrollo de la Práctica

En esta práctica se va a diseñar un solo perceptrón, en la cual con tan solo cambiar el valor de los pesos sinápticos o el bias, este deberá comportarse como una AND, NAND, OR, NOR, el diseño de este circuito se muestra en la figura 3



**Figura 3.** Circuito de la práctica

Vamos a variar los pesos sinápticos moviendo los potenciómetros Rv4 y Rv5, esta variación se realizara según los resultados obtenidos en matlab para distintas compuertas lógicas.

### 3.1 Compuerta AND

El código de MATLAB para que nuestro perceptrón aprenda a comportarse como una compuerta AND es:

clc, close all;

red=newp([0 5;0 5],1)

p = [0 0 5 5;0 5 0 5]%entradas

t=[0 0 0 1]% valor esperado

red=train(red,p,t); %entrenamiento

peso=red.iw{1}% visualizar pesos

bias=red.b{1}% visualizar bias

sim(red,p)% probar sistema

hold on;

figure,plotpv(p,t); %dibujar P y t

plotpc(red.IW{1},red.b{1}) %dibujar frontera decisión

grid on

hold off;



**Figura 4.** AND

Ya que el código es el mismo (únicamente cambiamos los valores esperados) para las siguientes compuertas, adjuntamos las graficas.

### 3.2 Compuerta OR



**Figura 4.** OR

### 3.3 Compuerta NAND



**Figura 5.** NAND

### 3.4 Compuerta NOR



**Figura 5.** NOR

### 3.5 Tabla de pesos y bias para diferentes compuertas

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| OR | 5 | 5 | -1 |
| NOR | -5 | -5 | 0 |
| AND | 5 | 5 | -27 |
| NAND | -5 | -5 | 26 |

**Tabla 1.** Pesos y bias para diferentes compuertas

## 3 Conclusiones

El comportamiento con el que actuara el perceptron ante determinadas entradas estará dado por los pesos puestos en ellas y el factor bias que regulemos en el mismo.

En la práctica realizada podemos ver que los valores medidos tienen un cierto error en relación a la simulacion en matlab, ya que las resistencias no son exactas y poseen una cierta tolerancia, además los voltajes que en el ingresan no son completamente puros.

Nuestro perceptrón al ser un sistema no muy extenso de dos entradas y de una sola capa, el aprendizaje de la neurona es relativamente corto pues las iteraciones que realiza son pocas. El diseño y la puesta a punto del hardware es sencillo y de bajo costo, pero hay que tomar en cuenta que al modelar un perceptron que necesiten mayor cantidad de entradas, salidas, mayor numero de capas, o muchas mas neuronas la implementación sería muy compleja, teniendo que optar por circuitos integrados o desarrollar todo nuestro sistema en software.

## Referencias

[1] MATLAB, RNA TOOLBOX. Obtenido de: http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/nnet\_tocframe.shtml

[2] JAIRO ALONSO TUNJANO, Presentaciones RNA. Obtenido de: http://dis.unal.edu.co/~fgonza/courses/2003/pmge/present/presentacionRNA.ppt

[3] CLAUDIO CASTELLANOS S´ANCHEZ, RNA : El perceptron simple y su regla de aprendizaje, AGOSTO 2008, http://www.tamps.cinvestav.mx/~castellanos/teaching/2009/2009RNA/Notes/RNA-Presentacion04.pdf

[4] El Perceptrón Simple, CAPITULO 4, obtenido de: http://www.lcc.uma.es/~jmortiz/archivos/Tema4.pdf