

# CONTENIDO

[1 CONTENIDO 1](#_heading=h.30j0zll)

[2 PRESENTACIÓN 2](#_heading=h.1fob9te)

[3 EL PERCEPTRÓN 4](#_heading=h.3znysh7)

[4 LÓGICA DIFUSA - INTRODUCCIÓN 5](#_heading=h.2et92p0)

[5 CONCLUSIONES 6](#_heading=h.tyjcwt)

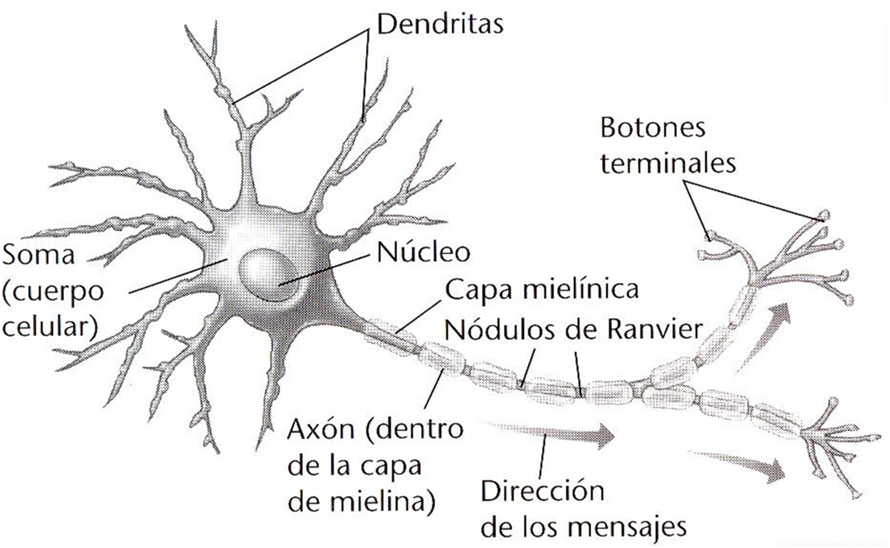
[6 BIBLIOGRAFÍA 7](#_heading=h.3dy6vkm)

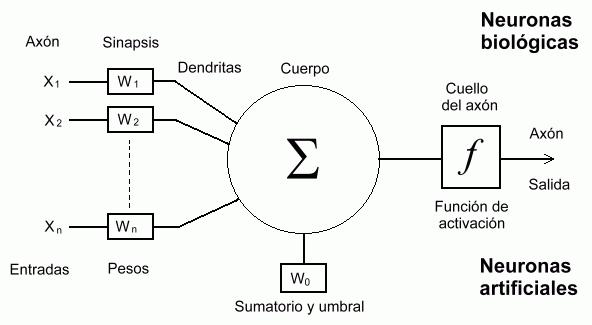
# PRESENTACIÓN

La presente monografía está orientada a la descripción de los elementos básicos de las neuronas artificiales, en particular el perceptrón, y la teoría fundamental de la lógica difusa.

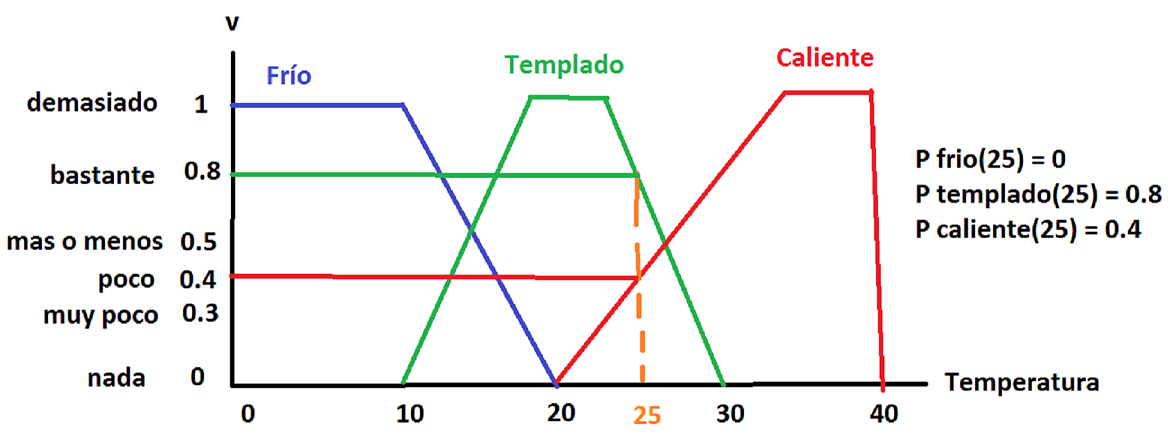
En el documento se analizan los diferentes elementos que componen ambas tecnologías, mostrando las relaciones matemáticas que dan soporte a las funcionalidades tanto del perceptrón como a los factores de incertidumbre que dan sentido a la lógica difusa.

A grandes rasgos, las redes neuronales se basan en los modelos que subyacen a las redes neuronales biológicas. El siguiente diagrama adelante algunos elementos presentes en esta tecnología.





La lógica difusa se basa en la concepción de que la verdad (y la falsedad) no son absolutas. Por este motivo, todos los conceptos que concibe el ser humano tienen cierto grado de certeza, el cual se expresa fácilmente si recurrimos a un esquema como el que se ve a continuación.



En este esquema se afirma que el Frío, la sensación de Templado, y algo que es Caliente, son curvas que varían de acuerdo con la temperatura, según se ve. En el caso particular de tener una temperatura ambiente de 25 grados, dicha temperatura tendrá un valor de verdad respecto de “Caliente” de sólo 0.4. En cambio, los 25 grados representarán, en la curva de “Templado”, un valor de verdad de 0.8. Se aprecia, además, que dichos valores se relacionan, de manera bastante cercana, con frases y/o palabras que utiliza el ser humano para describir situaciones de la vida real.

En las próximas secciones se verán estas tecnologías con un mayor grado de detalle.

**AUTOR:**

**Carlos Eduardo Hincapié López – 1010112674 – Carlos.hincapie@utp.edu.co – github.com/CarlosH14/ComputacionBlanda-G1**

**Alejandro Agudelo Toro - 1087494498 - alejandro.agudelo1@utp.edu.co - github.com/alejandroat/Computacion-Blanda**

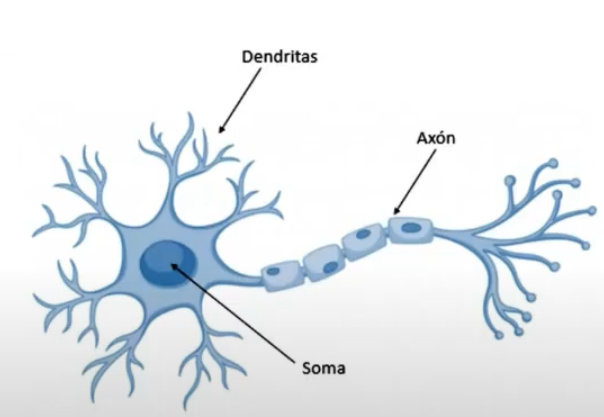
**Juan Manuel Sánchez Pareja - 1010107723 - juanmanuel.sanchez@utp.edu.co - https://github.com/omipareja/Computacion-Blanda**

# EL PERCEPTRÓN

La teoría básica del perceptrón se presenta a continuación:

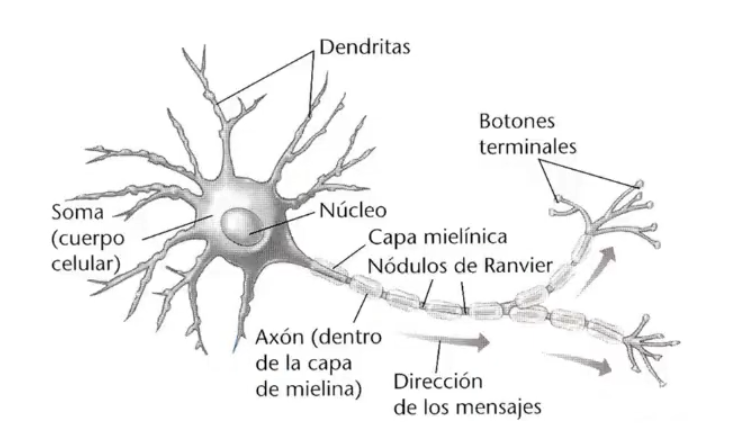
**Las redes neuronales:** Lo que hacen las redes neuronales es aprender, reconocer clasificar según la inteligencia artificial, tienen que ver con el cerebro con el modelo del cerebro que conocemos y la naturaleza de las neuron

**La neurona:**



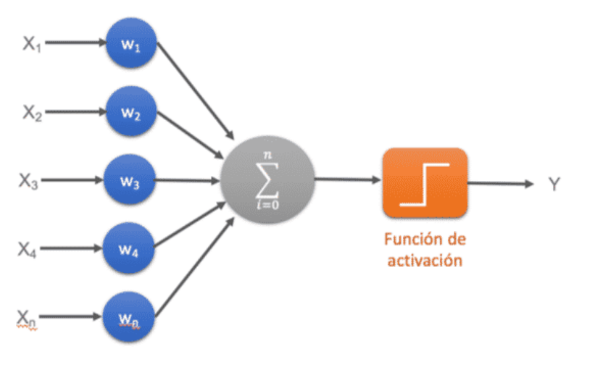
Una neurona es un componente biológico presente en el sistema nervioso,cuya característica es que se convierte en un transformador de energía de pulsos de señal eléctrica que la transmite a través de canales, de la cual aparece la inteligencia derivada de esta función.

La neurona tiene un núcleo (soma), unos capturadores de señal de entrada(Dendritas) , y unos transmisores de energía (Axón). A continuación se mostrará un análisis más detallado.



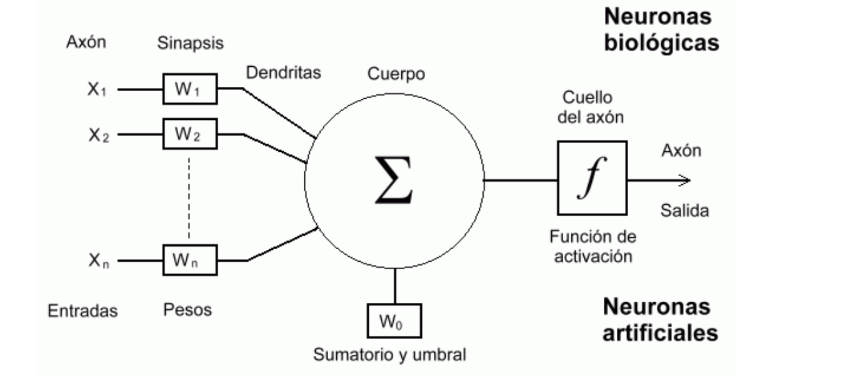
* **Núcleo-** Es una estructura ubicada en el centro de la neurona, generalmente muy visible, en la que se concentra toda la información genética.
* **Soma-** es el cuerpo celular de la neurona.
* **Dendritas-** Son múltiples ramificaciones que parten del precario y que actúan como zona de recepción de estímulos y alimentación celular, además de establecer conexiones entre las neuronas.
* **Axón-** El axón se encarga de conducir el impulso nervioso a lo largo del cuerpo y también hacia otras neuronas a través de las dendritas.
* **Nódulos de Ranvier-** interrupciones que ocurren a intervalos regulares a lo largo de la longitud del axón en la vaina de mielina que lo envuelve.
* **Botones terminales-** Es donde conecta con las demás neuronas.

**La neurona Artificial:**

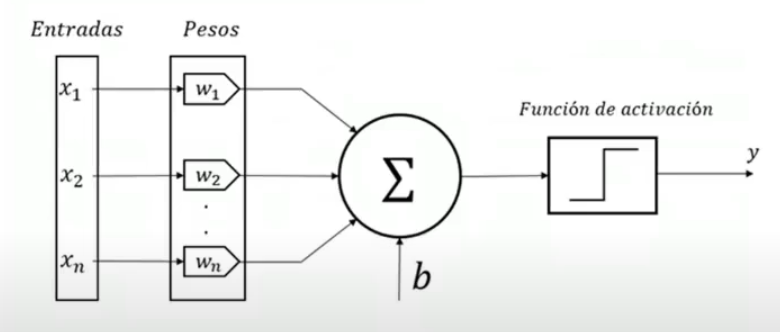
****

La neurona artificial posee unas entradas en este caso de X1 a Xn, y tiene un acceso a la neurona, la neurona artificial es la agrupación del sumador y de la función de activación, entonces las señales pueden ser amplificadas o reunidas este efecto es llamado sinapsis pero en el modelo artificial es llamado los pesos.

**El perceptrón:**

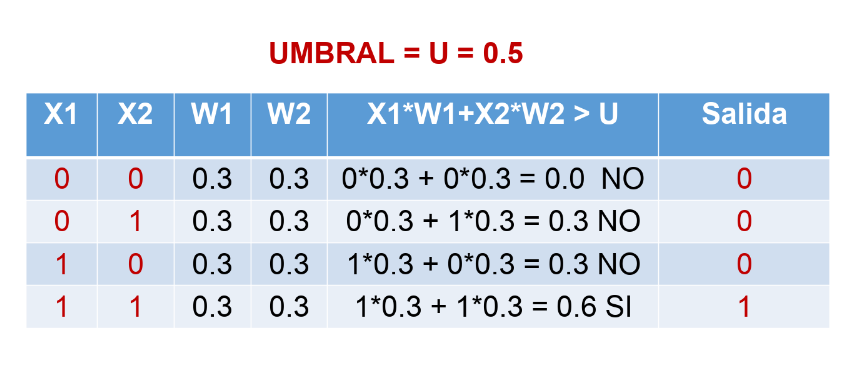
****

En este modelo se puede observar en la parte superior las partes de la neurona biológica y en la parte inferior las partes de una neurona artificial, mostrando la relación y comparación que hay entre los dos modelos de neurona.



Para propósitos generales en este caso nosotros tenemos unas entradas (X1,X2,Xn ) Y unos pesos (W1,W2,Wn) los cuales llegan a un núcleo donde se suman, hay un bias, el bias es el umbral, es un desplazamiento que hay que vencer, hay una función de activación en este caso es una onda cuadrada, cuando alcanzamos el umbral obtenemos una salida de “1” y en caso de que no se alcance obtenemos una salida de “0”.

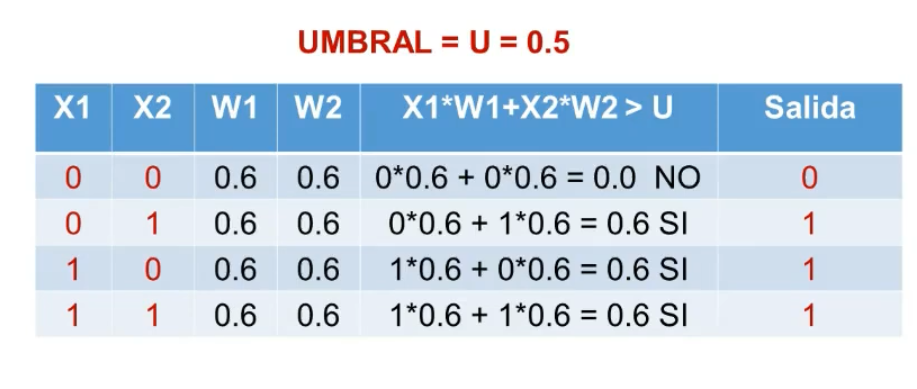
**Compuerta AND**

****

En este ejemplo se tienen dos entradas X1 y X2 y dos pesos W1 y W2 los cuales no cambian con valores de “0,3”, en este ejemplo la energía aplicada es “X1\*W1+X2\*W2”

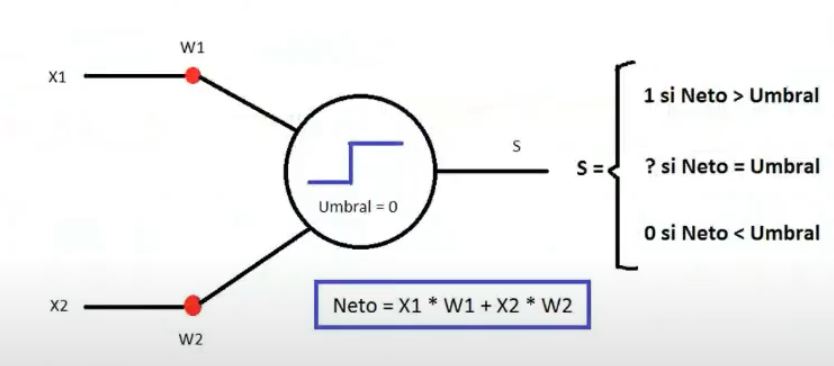
Se realizan todas las operaciones matemáticas en todas las filas, el umbral definido para este ejemplo es “0.5 “ y el único caso donde se supera el umbral es la última fila. Por lo tanto el comportamiento de esta neurona es el de la compuerta AND.

**Compuerta OR**



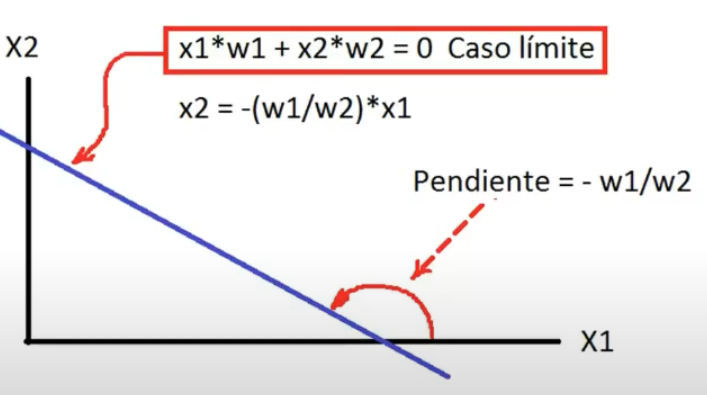
Para la compuerta OR el procedimiento es igual al ejemplo anterior pero en este caso se utiliza un valor de “0,6” para los pesos.

**Ecuación de umbral(en este caso cero)**

****

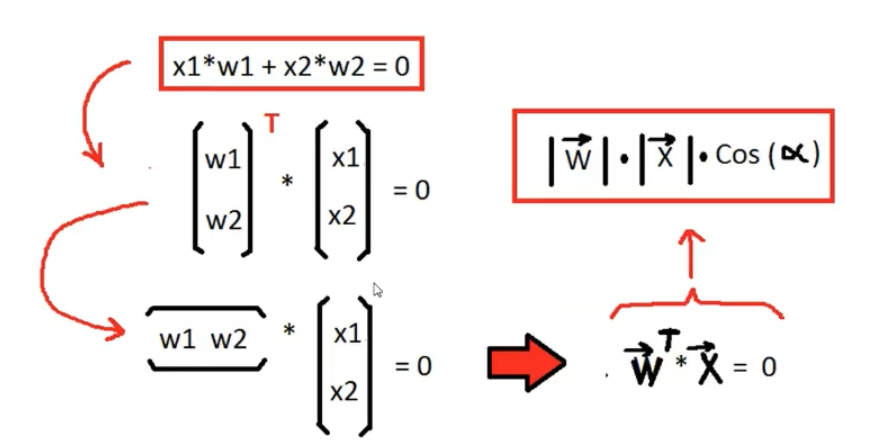
El neto es la sumatoria de toda la energía, En el gráfico se pueden observar los valores para que neto pueda traspasar el umbral, si el neto llega a ser igual al valor del igual podría ser “0” o “1” Esta zona es mejor evitarla.

**La recta frontera:**

****

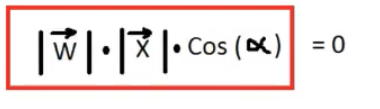
Se habla del caso límite cuando el umbral es igual a 0, tenemos dos variables de entrada (X1 , X2) y los pesos (W1,W2) en realidad son constante, los cuales serian los parametros de la función dando como origen una recta con pendiente X2 = -(w1/w2)\*x1.

**Análisis vectorial**

****

Todo lo que sucede de redes neuronales surge de esta demostración

**Perpendicularidad vectorial:**

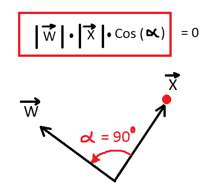
****

El producto punto de dos vectores es igual al producto punto de sus magnitudes por el coseno del ángulo entre dos vectores.

La recta solución es aquella para la cual el producto es igual a cero. En tal caso, esto significa que el ángulo alfa debe ser igual a 90 grados.

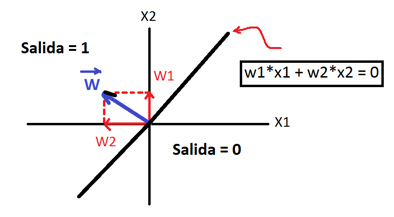
El vector W debe ser perpendicular al vector X para que se cumpla esta condición.

**Condición de frontera**



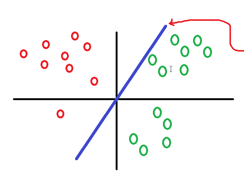
Si el vector W es perpendicular a todos los puntos de una cierta recta, entonces dicha recta cumple con la restricción de dividir el espacio en dos zonas, en una de las cuales la salida es uno, y en la otra es cero, según se vio en el perceptrón

**Separación espacial**



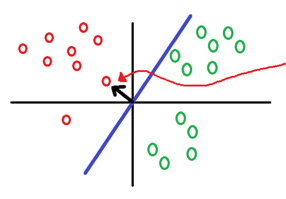
En el gráfico se puede observar la recta que divide las dos zonas surgiendo de cada una de ellas una salida, la primera de ellas que es hacia donde apunta el vector de pesos W es igual a 1, y su contraparte que va en dirección contraria al vector de pesos W es igual a 0

**Solución probable**



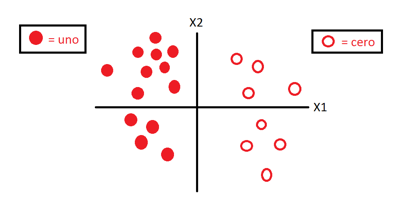
La recta de color azul señalada es una recta de una solucion probable al problema de dividir el espacio en dos areas de clasificación donde se separen los circulos de colores. Esto se puede calcular de forma automática modificando los pesos de manera iterativa hasta que dichos pesos sean un vector perpendicular a la recta solución

**Solución iterativa**



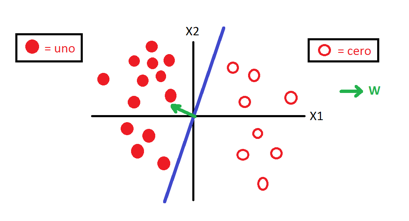
El vector de pesos W esta representado por la flecha que sale del origen y es de color negro y que es perpendicular a la recta azul que divide el espacio del grafico. Este vector se obtuvo de manera iterativa.

**Datos del entrenamiento**



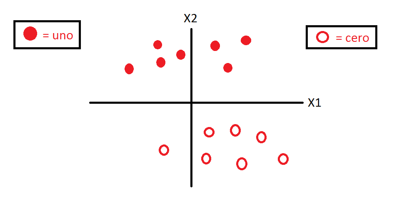
En el gráfico se muestra un plano cartesiano donde están distribuidos varios puntos, algunos rellenos y otros huecos, que servirán como datos de entrenamiento para ejemplificar algunas soluciones

**Solución intuitiva. W apunta hacia los UNOS**



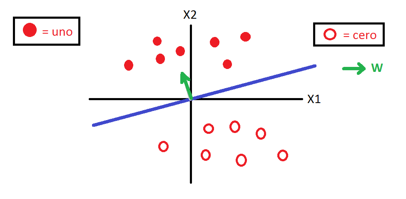
Como se mostró anteriormente para encontrar la recta que dividiera el gráfico en dos zonas donde se separen los datos, es necesario encontrar un vector perpendicular a la recta que se obtiene de forma intuitiva, este vector por lo general siempre apunta hacia los valores positivos.

**Datos de entrenamiento**



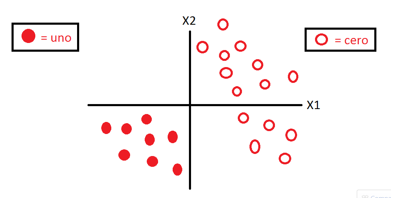
Este es un segundo ejemplo donde están distribuidos los círculos rellenos y huecos en el gráfico, posteriormente se grafica una recta la cual divide las zonas donde estan los circulos del mismo tipo

**Solución intuitiva. W apunta hacia los UNOS**



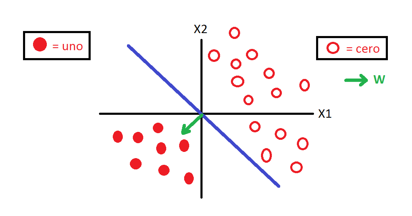
Intuitivamente se traza una recta que separa los círculos rellenos de los circulos huecos, y luego se traza el vector de los pesos que por lo regular apunta hacia los valores positivos y siempre será perpendicular a la recta de división

**Datos de entrenamiento**



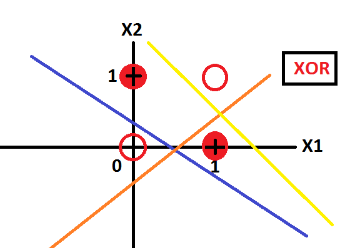
En este tercer y último ejemplo se puede apreciar que los círculos tienen otra disposición en el plano pero de igual manera es fácil deducir el sector por donde se puede trazar una línea que separe ambos tipos de datos

**Solución intuitiva. W apunta hacia los UNOS**



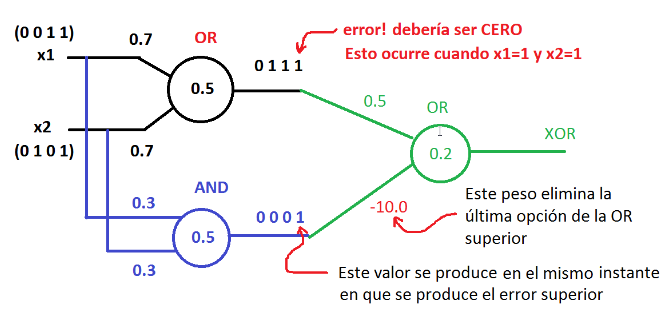
Cabe aclarar que la recta que se muestra no es la única recta que puede dividir ambas zonas, se podrían trazar algunas más inclinadas hacia los lados pero que pueden seguir cumpliendo con la función de dividir el gráfico en las zonas correspondientes, y siempre el vector de pesos W será perpendicular a esta recta y siempre estará apuntando hacia los valores positivos independientemente del cuadrante en el que se encuentren

**Compuerta OR - No Separabilidad:**



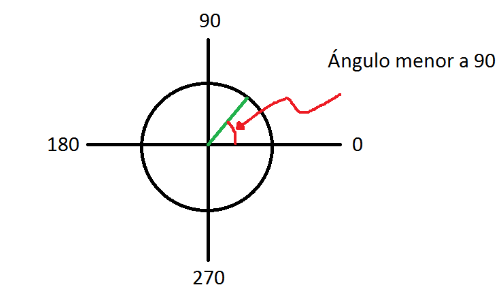
La función XOR no es clasificable con una neurona simple ya que así sea con muchos intentos de acomodar una recta ideal no es posible que en las zonas que quedan estén ubicados rojos y blancos por separado. Para poder clasificar cada área se debe dividir con más neuronas y formar subgrupos.

**Compuerta XOR - Solución:**

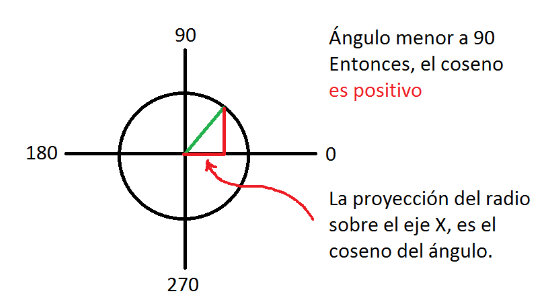


Para la solución de XOR se crea una neurona con una compuerta OR donde los pesos serán 0.7 y la compuerta AND donde los pesos serán 0.3, dando un umbral de 0.5 a cada compuerta. Se usa la compuerta AND para eliminar el último valor de la salida de OR donde se encuentra el error, creándose una nueva neurona OR los pesos para esta serán 0.5 para los valores de la OR anterior y -10 para los valores de AND para que se elimine la última opción de la OR superior el umbral asignado será 0.2.

**La Función coseno (Positivo):**

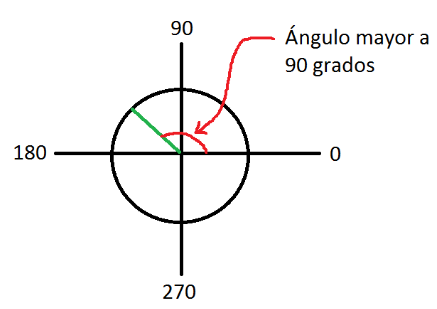
****

La función coseno concuerda con un círculo, se toma un radio donde el ángulo entre este y el eje horizontal .

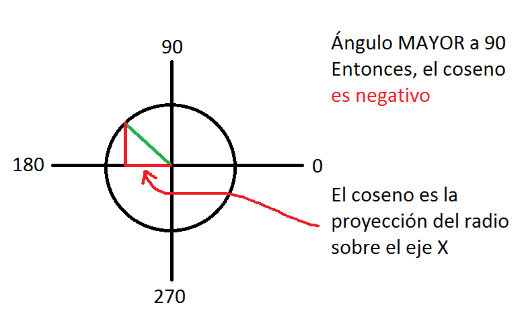


El coseno es la proyección del radio sobre el eje X. Teniendo en cuenta el origen todo lo que está hacia la derecha es positivo y todo lo que está a la izquierda es negativo, por esto en los ángulos menores a 90° el coseno es positivo.

**La Función coseno (Negativo):**

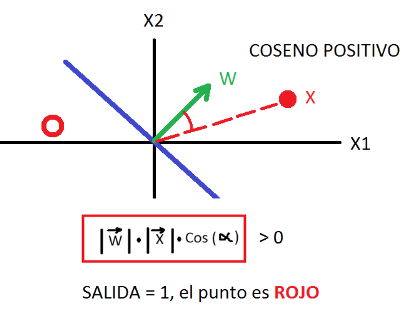


Ahora el radio está ubicado en donde el ángulo es mayor a 90°. ¿Qué pasa con el Coseno?

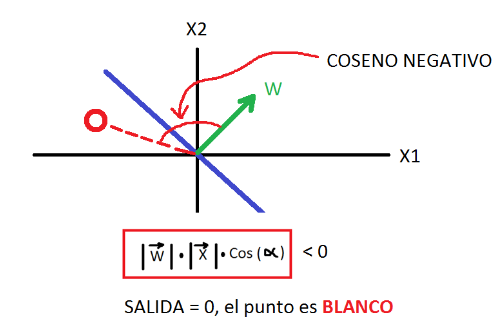


El coseno seguirá siendo la proyección del radio en el eje X, Pero ahora la proyección está en el lado negativo del eje por esto, ángulos mayores de 90° el coseno es negativo.

**Principio de Entrenamiento:**

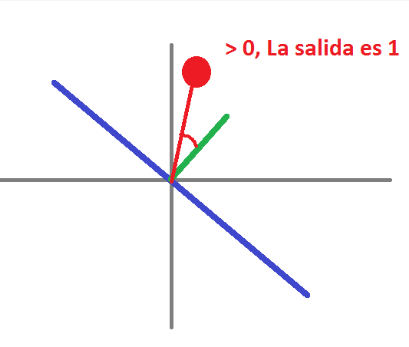
****

La línea azul divide lo que la neurona reconoce, w es el vector encontrado por w1 y w2 como ambos son positivos aparece en el primer cuadrante, el punto rojo cerca de w dispara la neurona ya que la salida de este punto será 1, el ángulo entre w y el punto x es menor a 90° por esto el coseno será positivo. Y el producto de las magnitudes y el coseno del ángulo es positivo o mayor que cero.



Ahora si el punto x está en el lado izquierdo. Dado que el ángulo creado entre w y el punto x es mayor de 90° el coseno es negativo. El producto entre las magnitudes y el coseno del ángulo mayor de 90° será menor que cero y no podrá superar el umbral dado en cero.

**Algoritmo de Entrenamiento - Correcto punto rojo en zona de salida = 1:**

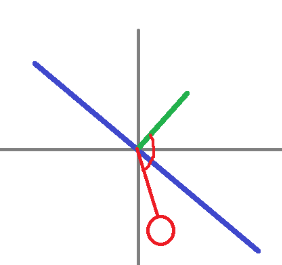


Cómo entrenar la red neuronal:

1. Se traza un vector w, que ayuda a separar el área en dos.
2. Convertir el punto en vector. (punto sacado de la base de datos)
3. Calcular el ángulo entre los dos vectores.

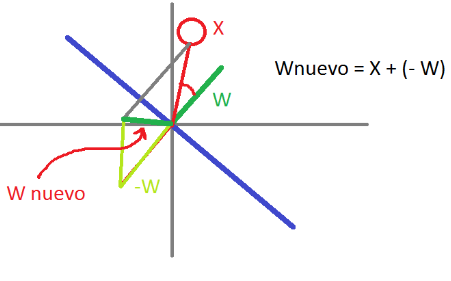
Por el valor del ángulo (menor a 90°) sabemos que la salida es 1.

**Algoritmo de Entrenamiento - Correcto punto blanco en zona de salida = 0**



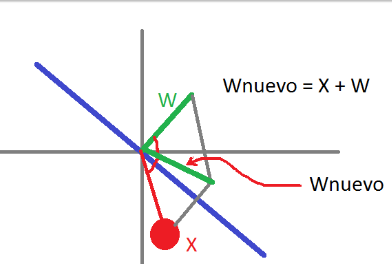
Este nuevo punto sacado de la base de datos se procede de igual manera, Miramos el ángulo que se forma entre los dos vectores como encontramos que es mayor a 90° sabemos que la salida es 0.

**Algoritmo de Entrenamiento - Error punto blanco en zona de salida = 1**



El punto está en la zona donde se produce la salida = 1, pero el punto debe producir una salida = 0. Para arreglar esto se debe organizar el vector w que hace que el punto sea contradictorio, Para esto se le debe cambiar el signo al peso (vector w), se suma el peso cambiado con el vector X, dando el nuevo w. Alejándose más de 90° de X.

**Algoritmo de Entrenamiento - Error punto rojo en zona de salida = 0**

****

El punto está en la zona donde se produce la salida = 0, pero el punto debe producir una salida = 1. Para arreglar esto se debe organizar el vector w que hace que el punto sea contradictorio, Para esto se suma el peso (vector w) con el vector X, dando el nuevo w. Acercándose más de 90° de X.

# LÓGICA DIFUSA - INTRODUCCIÓN

La teoría base de la lógica difusa se presenta a continuación.

La lógica difusa fue investigada en los años sesenta en la Universidad de Berkeley en California, Estados Unidos por Lotfy Zadeh, aunque se podría remontar hasta 2.500 años atrás cuando filósofos griegos, como Aristóteles, consideraban que existían ciertos grados de veracidad y falsedad; o Platón trabajando con grados de pertenencia

Zadeh nació en Azerbaiyán, estudió en la universidad de Teherán, en y posteriormente continuó con sus estudios en el MIT, en Estados Unidos. Es este país también estudió y ejerció en la universidad de Columbia.

Este ingeniero propuso el principio de incompatibilidad que reza: “Conforme la complejidad de un sistema aumenta, nuestra capacidad para ser precisos y construir instrucciones sobre su comportamiento disminuye hasta el umbral más allá del cual, la precisión y el significado son características excluyentes”.

En su trabajo más importante titulado Fuzzy Sets introdujo el concepto de conjunto difuso en el que los elementos sobre los que se construye el pensamiento humano no son números sino etiquetas lingüísticas.

La lógica difusa permite representar el conocimiento común trabajando con datos numéricos y términos lingüísticos simultáneamente; los términos lingüísticos son sustancialmente menos precisos que los datos numéricos, pero en muchas ocasiones aportan una información más ventajosa para el razonamiento humano.

Las características más atractivas de la lógica difusa son su flexibilidad, su tolerancia con la imprecisión, su capacidad para modelar problemas no-lineales, y su base en el lenguaje natural.

Un hito importante en el desarrollo de la lógica difusa fue establecido por Assilian y Mamdani en 1974 en el Reino Unido al desarrollar el primer controlador difuso diseñado para una máquina de vapor, pero la primera implantación real de un controlador de este tipo fue realizada en 1980 por F.L. Smidth & Co. en una planta cementera en Dinamarca.

Takagi y Sugeno desarrollan la primera aproximación para construir reglas fuzzy a partir de datos de entrenamiento, más tarde será el punto de partida para investigar modelos de fuzzy.

El interés de las redes neuronales contribuye gracias a su similitud con los sistemas fuzzy, al relacionar las dos técnicas los resultados son llamados neuro-fuzzy, sistemas fuzzy basados en redes neuronales. B.kosko contribuye en sistemas neuro fuzzy e introdujo lógica difusa a muchos lectores de redes neuronales.

En la década de los 90 aparecen los algoritmos genéticos, Estas tres técnicas se pueden combinar de múltiple manera, son herramientas muy potentes de los sistemas de control en la última década.

La intención de zadeh era manipular más eficiente la vaguedad del razonamiento humano, no se contaba el éxito de la lógica difusa en el capo automático de procesos, esto se debe al boom iniciado en 1987 Este boom fue el resultado de una estrecha colaboración entre el gobierno, las universidades y las industrias japonesas.

El control difuso ha sido aplicado con éxito en muy diversas ramas tecnológicas, por ejemplo, la metalurgia, robots para la fabricación, controles de maniobras de aviones, sensores de imagen y sonido (sistema de estabilización de la imagen en cámaras fotográfica y de video Sony, Sanyo y Cannon).

Estas son algunas aplicaciones que ya están en el campo de los sistemas expertos, la lógica difusa tiene bastante éxito en sistemas de control, se están buscando nuevos campos de aplicación para esta técnica.

Se está investigando en áreas de reconocimientos visuales o segmentos de ADN , la lógica difusa parece el futuro para abordar la cantidad de datos y gestionar la red.

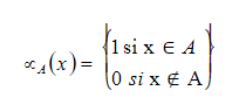
El primer ejemplo usado por Zadeh , fue el conjunto “los hombres altos”, se crea un conjunto donde todos con una estatura mayor a 1.80m es alto y un hombre de 1.79m ya no sería alto, pero no es muy lógico que por 1 cm un hombre no es alto.

El enfoque de la lógica difusa considera que el conjunto “hombres altos” no tiene una frontera clara mediante una función que define la transición de “alto” a “no alto” se asigna a cada valor de altura un grado de pertenencia al conjunto, entre 0 y 1. Ejemplo un hombre que mida 1.79 podría pertenecer al conjunto difuso “hombres altos” con un grado 0.8 de pertenencia.

Visto desde esta perspectiva se puede considerar que la lógica clásica es un caso límite de la lógica difusa en el que se asigna un grado de pertenencia 1 a los hombres con una altura mayor o igual a 1.80 y un grado de pertenencia 0 a los que tienen una altura menor.

La teoría clásica de conjuntos sólo contempla la pertenencia o no pertenencia de elemento a un conjunto, sin embargo la teoría de conjunto a difusos contempla la pertenencia parcial de un elemento a un conjunto, Este grado de pertenencia se define mediante la función característica asociada al conjunto difuso: para cada valor que pueda tomar un elemento o variable de entrada x la función característica μA(x) proporciona el grado de pertenencia de este valor de x al conjunto difuso A.

Formalmente, un conjunto clásico A, en un universo de discurso U, se define de varias formas: enumerando los elementos que pertenecen al conjunto, especificando las propiedades que deben cumplir los elementos que pertenecen a ese conjunto o, en términos de la función de pertenencia μA(x)

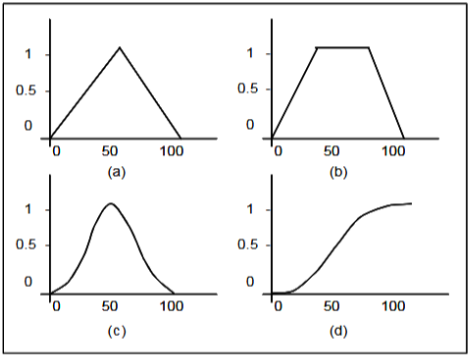
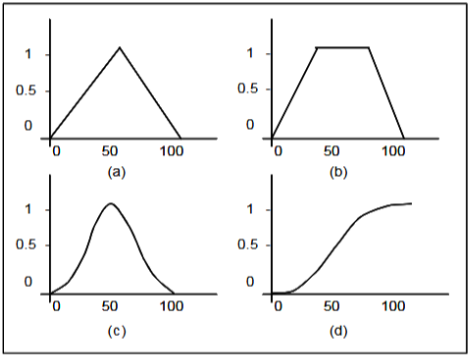
Podemos además decir que el conjunto A es equivalente a su función de pertenencia o característica.

Un conjunto difuso en el universo de discurso U se caracteriza por una función de pertenencia que toma valores en el intervalo [0,1], y puede representarse como un conjunto de pares ordenados de un elemento x, y su valor de pertenencia al conjunto.

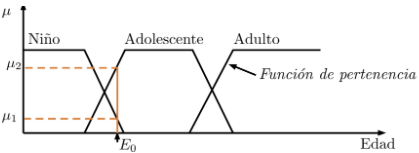
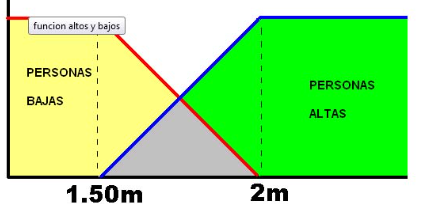
La función característica proporciona una medida del grado de similaridad de un elemento de U con el conjunto difuso. La función característica utilizada, depende del 

criterio aplicado para la resolución de cada problema y difiere por la cultura, geografía, época o punto de vista del usuario.

La única condición que debe cumplir esta es que tome valores entre 0 y 1, con continuidad. Las funciones características más comúnmente utilizadas por su simplicidad matemática y su manejabilidad son: triangular, trapezoidal, gaussiana, sigmoidal, gamma, pi.



Conceptualmente existen dos aproximaciones para determinar la función característica asociada a un conjunto: la primera aproximación está basada en el conocimiento humano de los expertos, y la segunda aproximación es utilizar una colección de datos para diseñar la función.

Ejemplos:

# CONCLUSIONES

* Los Conjuntos Difusos, pueden crear modelos que resultan útiles para tratar con la incertidumbre de manera más "natural" que la lógica y la teoría de conjuntos clásicas.
* Se caracteriza por Funciones de Pertenencia que dan flexibilidad a la modelación utilizando expresiones lingüísticas como: mucho, poco, caliente, frío, joven, viejo, alto, bajo.

# BIBLIOGRAFÍA

<https://repl.it>

<OTROS ENLACES ADICIONALES>