CAPÍTULO I
ALGUNOS HECHOS INTERESANTES DE LAS NEURONAS

- La idea de ser parte constituyente de la estructura cerebral la introducen Ramón y Cajal en 1911.

- El tamaño de una neurona es de cinco a seis veces más pequeño que una compuerta lógica en un chip de silicona.

- Los eventos en un chip ocurren en rangos de nanosegundos (10^{-9}), mientras que en una neurona es en milisegundos (10^{-3}).

- La efectividad de la neurona consiste en sus mecanismos de interconexión a través de conexiones masivas. Alrededor de 10 billones de neuronas y 60 billones de conexiones existen en la corteza cerebral. (Shepherd y Kock, 1990)

- El consumo energético del cerebro es 10^{-16} joules por operación por segundo, mientras que para el mejor computador de hoy en día está en el orden de 10^{-6} joules por operación por segundo. (Faggin, 1991)

ALGO DE HISTORIA

- McCulloch y Pitts (1943) fueron los pioneros al desarrollar los primeros cálculos lógicos de redes neuronales.

- Von Neuman (1949). Difunde la teoría McCulloch-Pitts después de ponerla en práctica en el computador ENIAC.

- Hebb (1949). Inicia sus estudios en Redes Neuronales Artificiales (RNA) poniendo en práctica por primera vez las modificaciones sinápticas entre dos neuronas como parte de las reglas de aprendizaje debido a la repetida activación de una neurona por otra a través de la sinapsis. Introduce la ley de aprendizaje de Hebb.

- Rosenblatt (1957). Desarrolla la máquina Mark I Perceptrón. Primer clasificador lineal basado en entrenamiento supervisado. Introdujo la ley de aprendizaje Perceptrón.

- Widrow y Hoff (1960). Introducen el algoritmo Least Mean-Squared (LMS) usado para derivar los adaptadores lineales ADALINE y MADALINE en problemas de clasificación. Se inicia el uso de la regla Delta como mecanismo de entrenamiento para el cambio de los pesos sinápticos.

- Minsky (1967). Difunde un texto llamado Perceptrón y advierte sus limitaciones. (Caso del XOR). Se alinean hacia la Inteligencia Artificial. Primeros ensayos en Reinforcement como nuevo régimen de aprendizaje.

- Grossberg (1960-1970), Kohonen (1962-1980). Redescubren el régimen de entrenamiento no supervisado mediante aprendizaje

competitivo. A su vez, las redes de Hopfield aparecen como una variante de entrenamiento no supervisado usando el concepto de funciones de energía y conexiones simétricas.

- Backpropagation (1974, Werbos) (1986, Rumelhart). Algoritmo de aprendizaje para perceptrones multicapas. De alta popularidad para la solución de problemas de clasificación y pronóstico.

- Radial Basis Functions, RBF (1988, Broomhead and Lowe). Alternativa al perceptrón multicapa basado en funciones potenciales y centros gravitacionales.

CARACTERISTICAS RESALTANTES DE LAS REDES NEURONALES ARTIFICIALES

- **Tolerante a las fallas.** Los pesos sinápticos pueden exhibir daños graduales y de este modo no muestran daños violentos. La red de neuronas está en capacidad de re-entrenarse sustituyendo las neuronas dañadas o en su defecto ir paulatinamente degradando su capacidad (enfermedad de Alzheimer)

- **Adaptativa.** En virtud de disponer los pesos sinápticos la capacidad de ajustarse a los cambios en su ambiente y a su vez, de disponer algoritmos de aprendizaje, la red neuronal puede adaptar su habilidad de clasificación o predicción. Más aun, cuando el ambiente es no estacionario, la red puede adaptar sus pesos sinápticos a tiempo real.

- **Infiere soluciones directamente desde los datos.** Contrariamente a los modelos convencionales, la red neuronal usa los datos para construir un mapa entre las entradas y las salidas de la red. Tal enfoque es inferencia no paramétrica, modelo libre de estimación.

- **Habilidad para generalizar durante el aprendizaje.** El uso de patrones (conjunto de datos) de entrenamiento contribuye a que las redes neuronales bien entrenadas pronosticarían satisfactoriamente desde nuevos conjunto de datos nunca vistos antes.

- **Información contextual.** El conocimiento está representado en la red neuronal. De este modo, cada neurona en la red está potencialmente afectada por cualquier actividad global del resto de las neuronas.

- **Implementa relaciones no lineales.** Una neurona es un mecanismo no lineal. De este modo la red en si misma es no lineal, lo cual la hace una de sus principales propiedades.

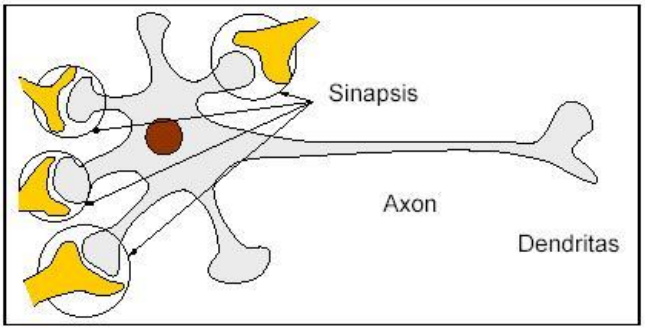
- **Operación paralela.** Esta condición de paralelismo garantiza a cada neurona la capacidad de memoria local, por un lado. Por el otro, le da adicionalmente una capacidad de procesamiento distribuido de la información debido a su conectividad a través de la red

- **Uniformidad en el análisis y diseño.**

LA NEURONA NATURAL Y LA ARTIFICIAL

La neurona natural.

- Cuerpos celulares que forman parte constituyente del cerebro.
- Especializadas para comunicaciones y computación.
- Su forma es diferente de los otros cuerpos celulares e inclusive, su forma en el cerebro varía de acuerdo a su función.
- Su membrana externa puede generar impulsos nerviosos. Su estructura incluye la sinapsis como medio que permite transferir información desde una neurona a otra.
- Es funcionalmente simple pero con un alto grado de conectividad con el resto de las neuronas



- Cuerpo celular: Contiene los bioquímicos que producen moléculas esenciales para la vida de la célula.
- Dendritas: conexiones que proveen la llegada de las señales. En la sinapsis, las dendritas reciben las señales a través de los sitios receptores.

- Axón: conectores que proveen la ruta para las señales salientes. Son más largas que las dendritas ramificándose solo al final, donde se conecta a las dendritas de otras neuronas.

LA SINAPSIS

- Se corresponde con el lugar donde es transferida la información de una neurona a otra.

- La sinapsis más común es entre un axón y una dendrita.

- No hay ningún tipo de conexión física o eléctrica en la sinapsis.

- Algunas sinapsis son excitatorias y otras son inhibitorias. El cuerpo celular al recibir los neurotransmisores, los combina y envía una señal de salida si es apropiado.

- Las vesículas sinápticas al ser activadas, liberan los neurotransmisores, que son estructuras esféricas contenidas en el axón.

- El tiempo de transmisión a través de la fisura sináptica es de 100 nanosegundos.

COMO SE CONSTRUYE LA INFORMACION

Dentro de la neurona, la información esta en forma de una señal eléctrica. Sin embargo, a través de la neurona la información es transmitida químicamente.

La señal eléctrica está en forma de pulsos de la misma magnitud. De allí, la información es enviada a través de pulsos, que variará su frecuencia en la medida que varíe la magnitud de los estímulos. Las frecuencias varían desde menos de 1 por segundo hasta cientos por segundo.

El pulso eléctrico es generado por unas bombas de sodio y potasio dispuestas en la membrana de la neurona. En descanso, el interior de la neurona es aprox. 70 mv negativo con respecto al exterior. La concentración de sodio en el interior es 10 veces más bajo que en el exterior y su concentración de potasio es 10 veces más alta.

Las bombas mantienen el equilibrio reemplazando iones de sodio y potasio. Cuando la neurona se activa, estas bombas reversan su acción, bombeando sodio dentro de la célula y potasio fuera de ella. Esto sucede en aprox. 1 mseg. El potencial interno de la célula alcanza un pico de aprox. 50 mv positivos con respecto al exterior. Las bombas reversan nuevamente y el voltaje de aprox 70 mv negativo es restaurado.

Un período refractario de varios milisegundos sigue a la activación de cada pulso. Durante este tiempo no pueden ocurrir activaciones. El potencial cambiado de –70 mv a 50 mv viaja a lo largo del axón hacia las conexiones sinápticas. Se activan los canales de voltaje controlados que liberan los neurotransmisores a través de la fisura sináptica.

Qué es una neurona artificial

- Elemento de procesamiento que forman parte constituyente de un modelo neuronal artificial.

- Pueden tener memoria local representada a través de los pesos sinápticos, los cuales pueden ser mejorados para fortalecer el aprendizaje.

- Poseen una función de transferencia o activación la cual puede usar memoria local o señales de entradas para producir una señal de salida.

- Las conexiones tanto de entradas como de salidas que puedan estar disponibles en una neurona artificial, determinan el grado de fortaleza de conexión en la red y le dan el carácter de procesamiento distribuido de información.

- El Sumador Lineal es la representación más simple de una neurona artificial.

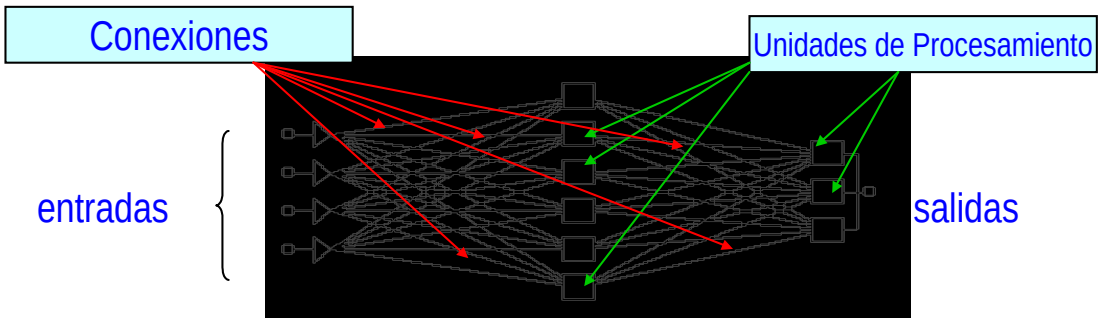
LA RED NEURONAL NATURAL Y ARTIFICIAL

Qué es una Red Neuronal (RN)

Es un conjunto de neuronas dispuestas en el cerebro (unidades de procesamiento) con memoria local propia, interconectadas mediante canales de comunicación (dendritas y axones) por los cuales se conducen impulsos mediante los neurotransmisores que son activados en la sinapsis que ocurre entre neuronas.

Qué es una Red Neuronal Artificial (RNA)

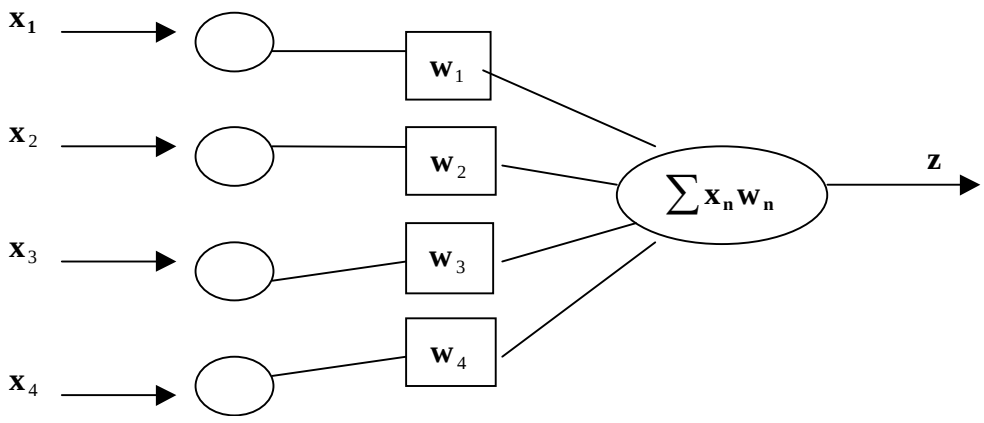
Es una red de unidades de procesamiento que tienen memoria local propia, interconectadas mediante canales de comunicación por los cuales se conducen datos numéricos. Cada unidad ejerce operaciones a través de las conexiones que son excitatorias o inhibitorias controladas por los pesos sinápticos.



UTILIDAD DE LA RNA

- Puede procesar cualquier función. (ej.: útil para clasificación, aproximación de funciones, pronóstico).
- Parte esencial en la minería de datos para el reconocimiento de patrones en los datos.
- Procesos de control y aseguramiento de la calidad.
- Control de procesos de plantas.
- Transformación de los patrones de entrenamiento mediante reducción de datos y/o selección de variables.
- Problemas de análisis de supervivencia
- Construcción de índices de seguridad y de riesgo financiero.
- Control de fraudes electrónicos.
- Reconocimiento de imágenes.
- Débil para modelos que una memorización de todo su dominio (ej.: números primos, factorización, etc.)

EL SUMADOR[1] LINEAL



z es la combinación lineal de las entradas x(s) y de los pesos sinápticos w(s).

En notación matricial, el sumador lineal para múltiples entradas (x_1, x_2, …., x_n) y múltiples salidas (z_1, z_2, z_3, …., z_m), sería: z=Wx, donde W es una matriz de orden mxn (m salidas y n entradas)

SUMADOR LINEAL vs NEURONAS

Similaridades:

- El aprendizaje se logra a través de los cambios en los pesos. En las neuronas lo hacen a través de la sinapsis.
- La suma de las entradas combinadas a sus pesos para producir su asociado z. Las neuronas se inhiben o activan para producir información.

Diferencias:

- La salida del sumador lineal no satura.
- No existen umbrales en los sumadores lineales.
- No hay salidas negativas para las neuronas.

[1] Es una traducción de linear associator. Se entiende como la sumatoria de la combinación lineal de los pesos sináptico o parámetros y los valores de las variables de entrada o ejemplos de entrada.

LA LEY DE APRENDIZAJE DE HEBB

De acuerdo al postulado de Hebb (1949), nosotros tenemos:

"Cuando un axón de la célula A excita la célula B y, repetidamente la activa, algún proceso de crecimiento o cambio metabólico toma lugar en una o ambas células tal que la eficiencia de A como la de B crecen."

En otras palabras la actividad coincidente tanto en las neuronas presinápticas como postsinápticas es crítica para el fortalecimiento de la conexión entre ellas. (S.A. Set. 1992).

Procedimiento:
Para un vector x, el vector de salida es z=Wx.

Si un vector de salida deseado z_1 es previamente conocido, entonces $W=z_1x_1$.

Sustituyendo a z_1, tendríamos $z= z_1x_1^T x_1$.

Si los vectores son ortonormales (delta de Kronecker) , entonces

$$x_1^T x_1=1 \text{ y } z =z_1.$$

Esto quiere decir que si seleccionamos a W como fue indicado, nosotros podemos crear la asociación deseada entre la entrada y la salida.

Supongamos ahora que disponemos de otro vector de entrada x_2.

La salida actual será $Wx_2 = z_1x_1^Tx_2$.

Si x_1 y x_2 son ortonormales, entonces su producto $x_1^T x_2=0$ y además Wx_2 será también 0.

De este modo, para obtener el valor deseado de z_2 cuando x_2 es la entrada (también se puede decir patrón de entrada) se deben calcular los nuevos pesos W. Es decir

$$W_{nuevo} = W_{viejo}+z_2x_2^T = z_1x_1^T +z_2x_2^T.$$

Para todos los patrones de entrada ortonormales, este proceso se repite hasta que en general, la ley de aprendizaje de Hebb quedaría como:

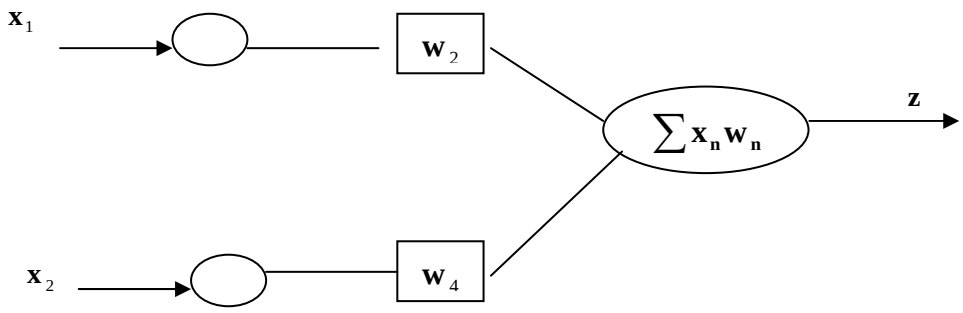
$$W_{nuevo} = W_{viejo} + z_kx_k^T,$$

Esto es válido para los k patrones de entrada/salida, cuyos pesos iniciales serían puestos en cero.

En conclusión esta ley es útil siempre y cuando los vectores sean ortonormales. El problema es que esta condición es muy estricta De ahí que podríamos decir que el número de ejemplos para entrenamiento estaría limitado a la dimensionalidad de x. Así, en un espacio n-dimesional, el número de patrones de entrada que pueden asociarse será n.

Qué pasaría con z si los patrones de entrada no son ortogonales?

Un ejemplo de aplicación de la ley de Hebb con dos patrones de entrada $x=(x_1\ \ x_2)$; es decir (0 1) y (1 0) y dos patrones de salida respectivamente 2 y 5.



Para el primer par x y z, [0 1] y 2, $W=z_1x^T = 2\ [0\ 1] = [0\ 2]$

Para el segundo par [1 0] y 5, entonces actualizaríamos los pesos para establecer la nueva asociación. Es decir $W_{nuevo} = W_{viejo} + zx^T$.

$$W_{nuevo} = [0\ 2] + 5\ [1\ 0] = [5\ 2].$$

Chequeamos si hubo entrenamiento y los pesos se ajustaron para reconocer las asociaciones.

Si la entrada hubiera sido [1 0], la salida z sería

$$z = Wx^T = [5\ 2]\ [1\ 0]^T = 5.$$

Si la entrada hubiera sido [0 1], la salida z sería

$$z = Wx^T = [5\ 2]\ [0\ 1]^T = 2.$$

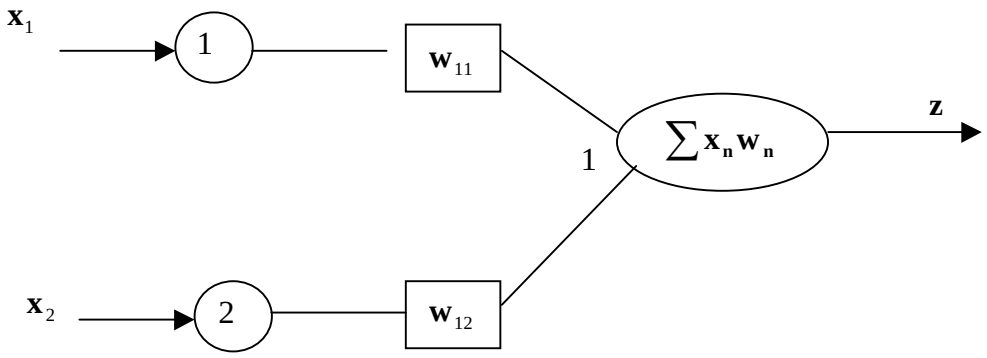
Además [1 0] y [0 1] son ortonmormales (unitarios y ortogonales).

Otro ejemplo.

Dado los vectores $\mathbf{x_1} = \left(1/\sqrt{30}\right)(1\ \ 2\ \ 3\ \ 4)^{\mathbf{T}}$, $\mathbf{y_1} = (3\ \ 2\ \ 1\ \ 0\ \ -1)^{\mathbf{T}}$, $\mathbf{x_2} = \left(1/\sqrt{81}\right)(8\ \ -3\ \ 2\ \ -2)^{\mathbf{T}}$, $\mathbf{y_2} = (3\ \ 5\ \ -2\ \ 9\ \ -1)^{\mathbf{T}}$, construir la matriz de aprendizaje de Hebb (W) para ambos pares de ejemplos. Verificar si se logró la asociación.

En general, la ley de Hebb realiza cambios locales. Es decir, si por ejemplo se toma el par de entrenamiento **z** y **x**, el cambio en la matriz de pesos estaría dado por $\mathbf{zx}^T$. Así por ejemplo, el peso w_{12} estaría afectado solamente por la entrada y la salida que involucra la conexión 1, 2. De allí que esta ley se le conozca con el nombre de **Regla de Aprendizaje Local**. Es decir, los cambios de los pesos pueden ser implementado por los procesos locales.

NOTA: La designación de los pesos mediante subíndices (w_{ij}), permite establecer un estándar en la lectura de los mismos. De este modo, por ejemplo w_{12} se corresponde con el peso correspondiente a la salida 1 y entrada 2 de la arquitectura de la red.



APRENDIENDO ASOCIACIONES USANDO LA REGLA DELTA

También conocida como la regla de Widrow y Hoff.

Supongamos que tenemos un conjunto de vectores que no son ortonormales pero que si son linealmente independientes, entonces la pregunta es si nosotros podemos asociar, supongamos L patrones de entrada con sus correspondientes salidas.

La respuesta sería si, siempre y cuando a la ley de Hebb se le hace la siguiente variación.

Partiendo de un conjunto inicial de pesos aleatorios W,

a) Calcular $\mathbf{W_{nuevo}} = \mathbf{W_{viejo}} + \eta\delta^{nuevo}\mathbf{x_{nuevo}}^{\mathbf{T}}$;

b) $\Delta\mathbf{W} = \eta\delta^{\mathbf{nuevo}}\mathbf{x}^{\mathbf{T}}$

c) $\delta^{nuevo} = \mathbf{z_{deseado}} - \mathbf{W_{viejo}x_{nuevo}}$

donde:
$\eta =$ es una constante positiva seleccionada por el usuario y su valor está basado en la experiencia. Típicamente el valor está entre 0.01 y 10. El valor de 0.1 es el más usado.
$\delta =$ se corresponde con el error observado entre el valor deseado y el valor calculado de las salidas.

De este modo, aplicando la Regla Delta se encontrará una matriz de pesos que minimizará el error entre los valores deseados y los valores calculados.
Este método garantiza converger en un número finito de pasos. Para lograr la convergencia, los pares de entrenamiento (x,z) serán usados

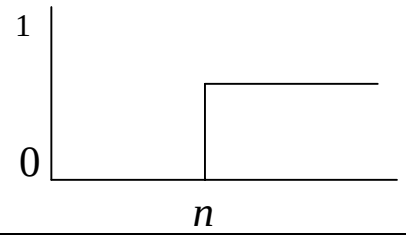
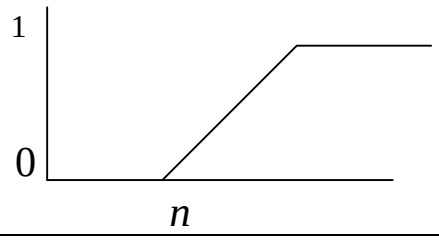
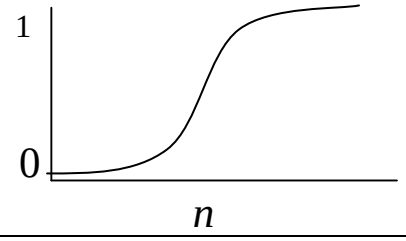
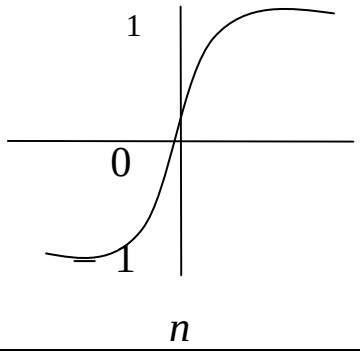
repetidamente hasta que satisfaga los requerimientos del error seleccionado por el usuario.

FUNCIONES DE ACTIVACION MÁS USADAS

Sea dada la combinación de las entradas y sus respectivos pesos mediante la suma representada por $n = \sum x_i w_i$.

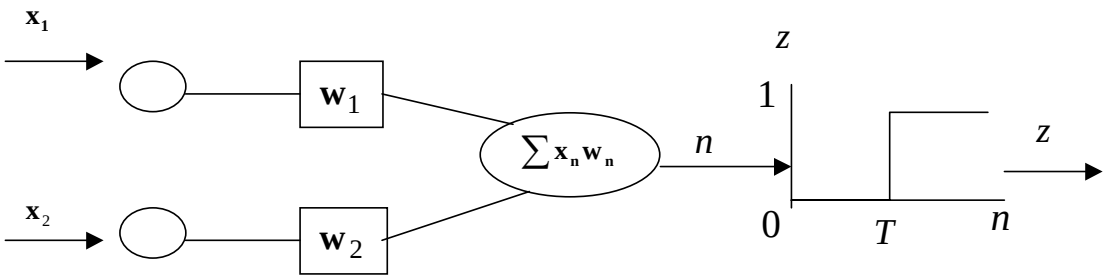
Esta respuesta de asociación podría ser afectada mediante una función de activación o transferencia de carácter no lineal a los fines de establecer valores extremos a la salida de la unidad de procesamiento.

Las más comunes son:

Función umbral:	
Función continua saturada:	
Función sigmoide:	
Función tangente hiperbólica:	

EL PERCEPTRON DE ROSENBLATT

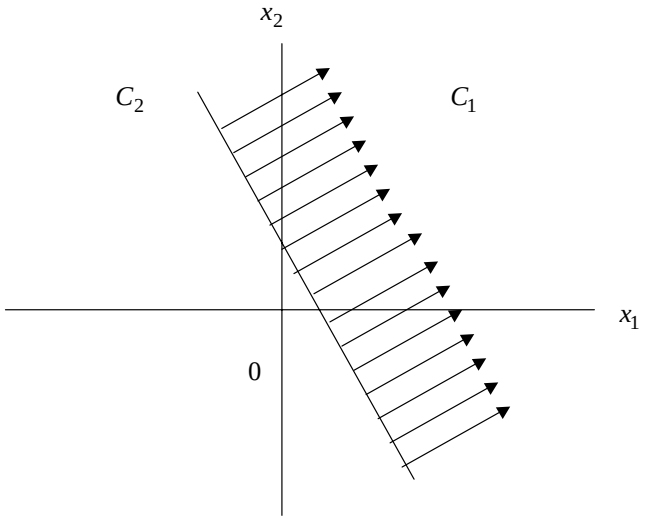
Antes que nada veamos la Unidad Lógica Umbral (Threshold Logic Unit - **TLU**)



(x_1 x_2) es el conjunto de entrada de entrenamiento, donde el valor de z quedaría clasificado de acuerdo al umbral T en

z = 1 si $n >= T$
z = 0 de cualquier otra manera

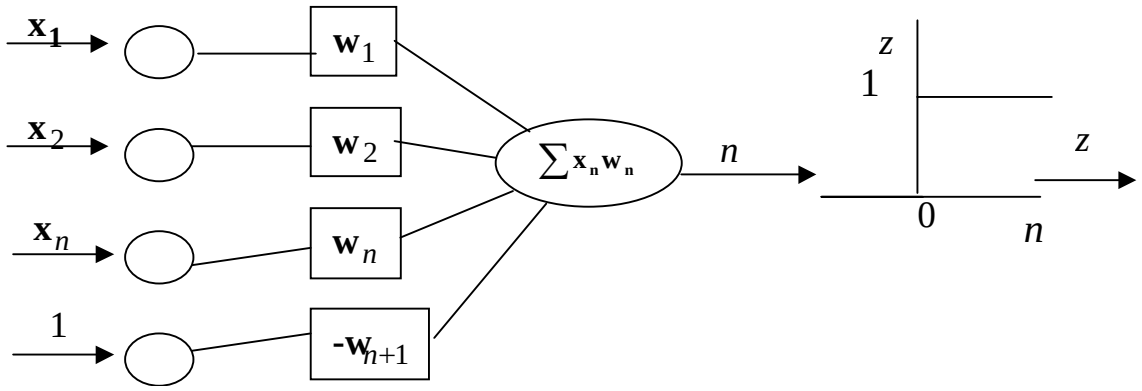
De este modo, todos los vectores de entrada quedarán clasificados en dos grandes clases C_1 y C_2.
En el caso de un vector de entrada de dos componentes, estaremos hablando de una línea de decisión que separa cualquier (x_1 x_2) de las dos clases.



La recta de decisión es $w_1x_1+w_2x_2-T = 0$.

Si al valor umbral T lo consideramos justamente otro peso (w_{n+1}), entonces la Regla Delta puede ser usada para variar el valor de T.

De este modo, el vector de entrada sería aumentado en un nuevo componente. Es decir, si existen x_n componentes en el vector de entrenamiento, el vector aumentado incluiría el componente x_{n+1} cuyo valor es la unidad.
Para mantener la misma nomenclatura, entonces tendríamos un nuevo peso w_{n+1} tal que la arquitectura de la red para un TLU sería:



Respetando la expresión anterior, tendríamos que la recta de decisión para clasificación sería:

$w_1x_1+w_2x_2+\ldots.+w_nx_n-w_{n+1} = 0.$

Donde $z = 1$ si $\sum x_i w_i - w_{n+1} >= 0$
$z = 0$ de otro modo

Esta unidad se le conoce como el Perceptrón de Rosenblatt y su modo de entrenamiento es a través de la ley de aprendizaje de Rosenblatt, que para una unidad de neurona sería:

$$w_{nuevo} = w_{viejo} + (z - z`)x^T$$

esta expresión es una versión de la Regla Delta con una tasa de aprendizaje $\eta = 1$, donde:

z es la salida deseada de la unidad (1 o 0)

z` es la salida estimada por la unidad (1 o 0)
x es el vector de entrada para el entrenamiento de la unidad
w es el vector con todos los pesos conectados a la unidad.

UN PROBLEMA DE CLASIFICACION

Dado un conjunto de vectores de entrada con sus clasificaciones conocidas.

1 0 1 -> 1
0 0 0 -> 0
1 1 0 -> 0

Encontrar los pesos que permitirán al TLU clasificar correctamente al conjunto.

(por ej. $w^T = [0.5 \; –0.5 \; –0.2 \; –0.1]$

Cuál sería la arquitectura de la TLU aumentada?

METODO DE ENTRENAMIENTO DE LA REGLA DELTA PARA EL TLU

1. Determinar un conjunto de entrenamiento que pueda ser clasificado en una o dos clases (ej: 0 o 1)

2. Inicializar los pesos con valores aleatorios. Tener presente que si el vector de pesos es n-dimensional, entonces habrán n+1 pesos.

3. Aumentar el conjunto de entrenamiento con un nuevo componente de entrada con un valor de 1. Dicho componente es el bias y representa el componente n+1 en el vector n-dimensional de entrad.

4. Modificar cada peso usando:

   $\Delta \mathbf{w_i} = C(\mathbf{z} - \mathbf{z}')\mathbf{x_i}$, sobre la base de un peso a la vez de una salida

   $\Delta \mathbf{w} = C(\mathbf{z} - \mathbf{z}')\mathbf{x}^{\mathbf{T}}$, sobre la base de un vector de pesos y una salida

   Donde:
   C es una constante de aprendizaje previamente seleccionada.
   x es el vector de entrada aumentado.
   z es el valor deseado y **z`** el valor estimado.

5. Repetir los pasos 3 y 4 para cada uno de los patrones de entrada.

6. Repetir 3, 4 y 5 para todo el conjunto de entrenamiento hasta que todos queden clasificados.

NOTA: UN EPOCH (EPOCA) ES UN ITERACIÓN REALIZADA CON TODO EL CONJUNTO DE ENTRENAMIENTO PARA MEJORAR LOS PESOS.

EJEMPLO

Dado el conjunto de entrenamiento:
1 0 1 -> 1
0 0 0 -> 0
1 1 0 -> 0

Y el conjunto aumentado a través del bias con valor unidad:
1 0 1 1 -> 1
0 0 0 1 -> 0
1 1 0 1 -> 0

La constante de aprendizaje *C* es 0.5.

Los pesos iniciales elegidos son [0 0 0 0].

El cálculo iterativo (por epochs) hasta alcanzar ningún cambio en los pesos comienza así:

Entrada	z	w_{viejo}	n	z`	(z-z`)	$C(z-z`)x^T$	w_{nuevo}
1 0 1 1	1	0 0 0 0	0	0	1	0.5 0 0.5 0.5	0.5 0 0.5 0.5
0 0 0 1	0	0.5 0 0.5 0.5	0.5	1	-1	0 0 0 -0.5	0.5 0 0.5 0
1 1 0 1	0	0.5 0 0.5 0					

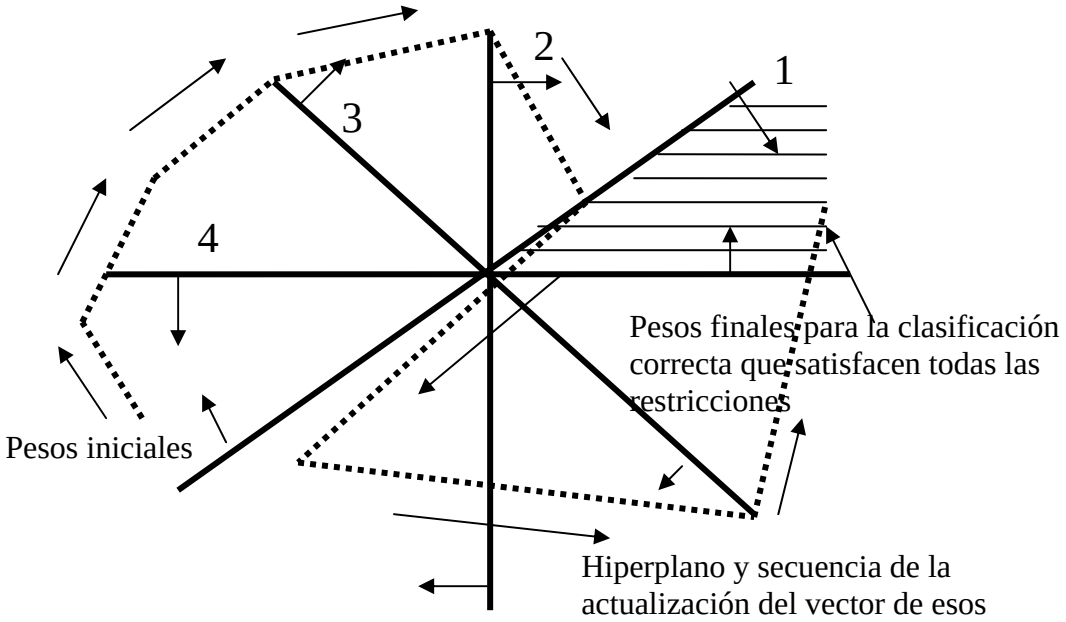
(Completar el cálculo…..)

Y la arquitectura de la red con los pesos estimados es:

COMENTARIOS ACERCA DE LA REGLA DELTA

Si los patrones o vectores de entrada son *linealmente separables*, entonces la regla delta en un número finito de pasos encontrará un conjunto de pesos que clasificará las entradas correctamente. La constante de aprendizaje debe ser mayor que 0.

Por ejemplo, sean cuatro patrones de entrada aumentados representados por el hiperplano de la figura. Las flechas indican sobre que lado de cada hiperplano, el vector de pesos debe estar para que pueda clasificar correctamente los patrones de entrada. Los patrones están presentados en secuencia: 1,2,3,4,1,2,3,4, etc.



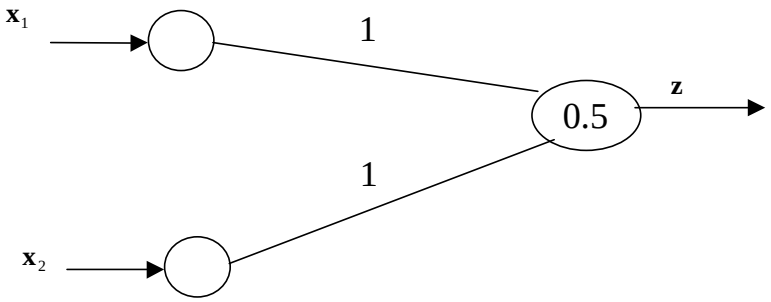
El tamaño que se considere para la constante de aprendizaje determina que tan rápido el algoritmo pueda converger a una solución. El valor de la constante de aprendizaje puede ser seleccionado en el rango entre 0,01 y 10. El valor más común de comienzo es 0,1.

Una limitación de la regla delta es que no funciona si los vectores de entrada no son linealmente separables. Es decir totalmente independientes. Esto produciría procesos de búsqueda de solución cíclica oscilando continuamente.

IMPLEMENTACION DE LAS FUNCIONES LOGICAS CON TLU

LAS ENTRDAS Y SALIDAS SON BINARIAS.

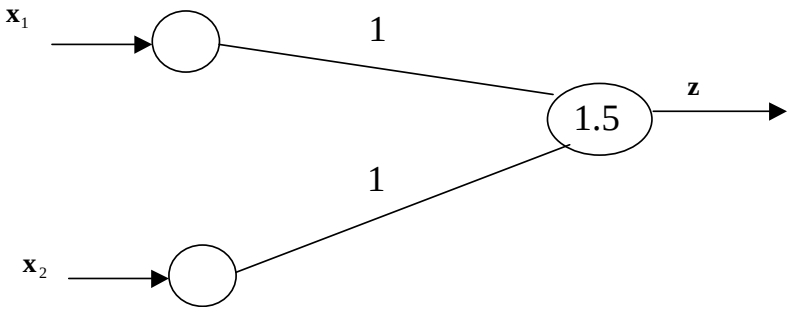
A) X_1 OR X_2



OR

X_1	X_2	Z
0	0	0
1	0	1
0	1	1
1	1	1

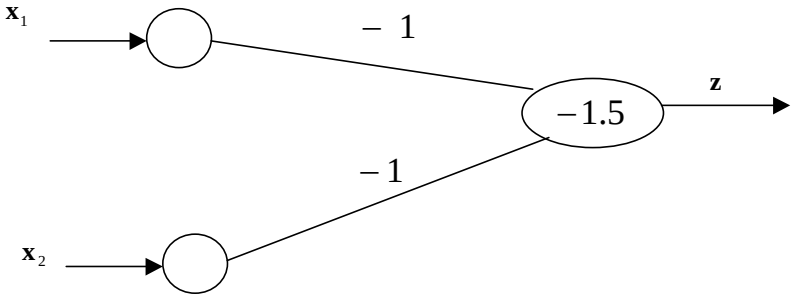
B) X_1 AND X_2



AND

X_1	X_2	Z
0	0	0
1	0	0
0	1	0
1	1	1

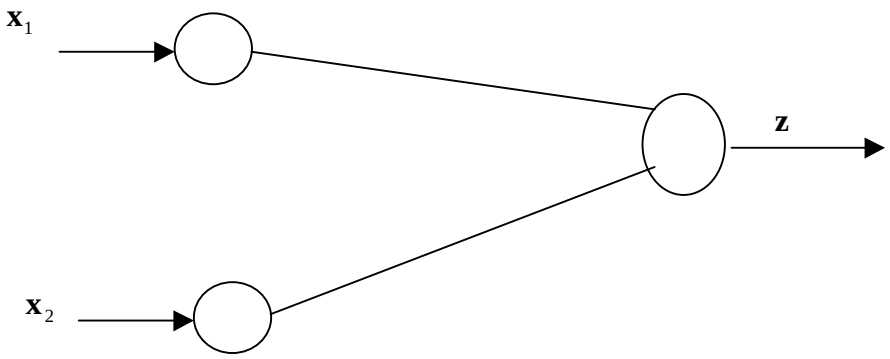
C) NAND: NOT(X_1 AND X_2) / X_1 NOT X_2



NOT

X_1	X_2	Z
0	0	1
1	0	1
0	1	1
1	1	0

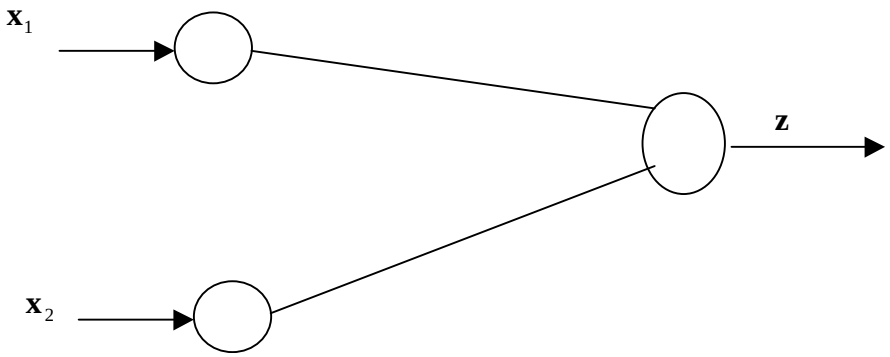
D) NOR: NOT(X_1 OR X_2)



NOR

X_1	X_2	Z

E) FALSE



FALSE

X_1	X_2	Z

F) XOR

X_1	X_2	Z
0	0	0
0	1	1
1	0	1
1	1	0

FUNCIONES LOGICAS DE DOS ENTRADAS

X_1 0	X_2 0	X_1 0	X_2 1	X_1 1	X_2 0	X_1 1	X_2 1	FUNCION
0		0		0		0		FALSE
0		0		0		1		AND
0		0		1		0		
0		0		1		1		
0		1		0		0		
0		1		0		1		
0		1		1		0		XOR
0		1		1		1		OR
1		0		0		0		
1		0		0		1		OR EQ.
1		0		1		0		
1		0		1		1		
1		1		0		0		
1		1		0		1		
1		1		1		0		NAND
1		1		1		1		TRUE

14 de ellas se pueden implementar en una red neuronal TLU con una sola capa.

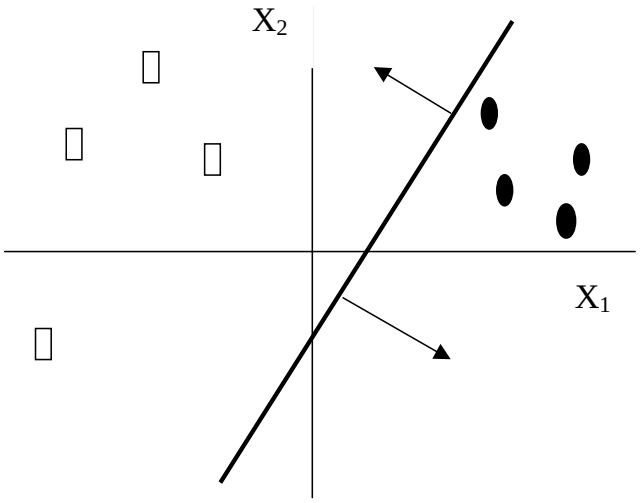
En general, si se tienen n entradas binarias, entonces habrán $P=2^n$ diferente combinaciones de las entradas.

Y en consecuencia, existen 2^P diferentes funciones binarias de n entradas binarias.

Ejemplo. Qué pasaría con patrones de entrada definidos por tres entradas binarias.

QUE ES SEPARABILIDAD LINEAL

Dado un grupo de patrones de entrada pertenecientes a una o más categorías, se dice que dichos patrones son linealmente separables si un TLU puede clasificarlos apropiadamente a todos ellos.
Por ejemplo, una entrada de dos dimensiones con una entrada que no es binaria.



Las clases están representadas por □ y ●.

Un TLU puede clasificarlos en dos categorías mediante la siguiente expresión, como ya se ha visto.

$$z = 1 \text{ si } w_1x_1 + w_2x_2 - w_3 >= 0$$
$$z = 0 \text{ si } w_1x_1 + w_2x_2 - w_3 < 0$$

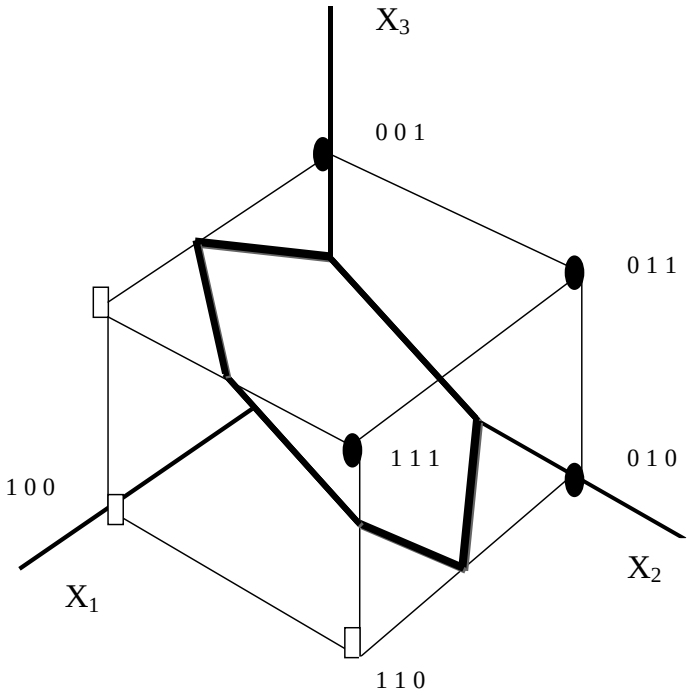
donde w_3 se corresponde con el peso del bias

Exactamente en el borde entre ambas clasificaciones la relación que se mantiene es la siguiente:

$$w_1x_1 + w_2x_2 - w_3 = 0$$

Podemos observar que la expresión define la línea de clasificación donde los valores de intersección de dicha línea con los ejes x_1 y x_2 son w_3/w_2 y w_3/w_1 respectivamente. Esto sería válido para un TLU con una sola capa.

Para el caso de patrones en un espacio tridimensional, la clasificación entre categorías estaría definido por un plano.

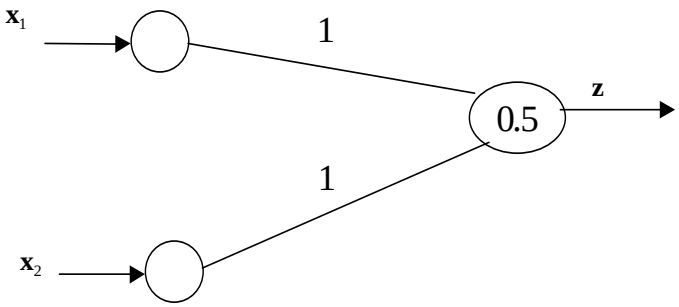


Para patrones mayores de tres dimensiones, el TLU implementaría hiperplanos.

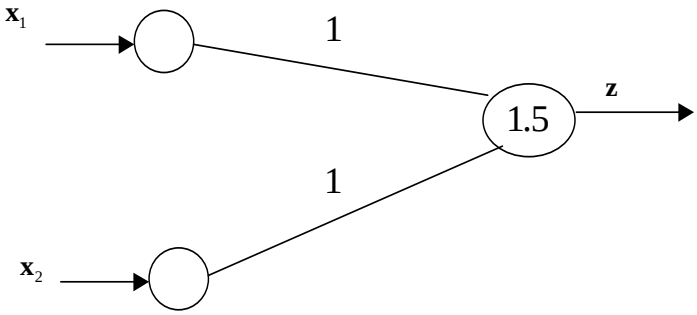
Supongamos que queremos separar los patrones de entrada en más de dos categorías.

Veamos los ejemplos anteriores acerca de la clasificación del OR, AND y NAND.

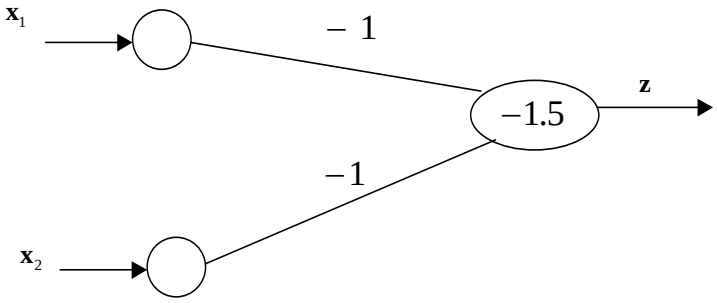
OR



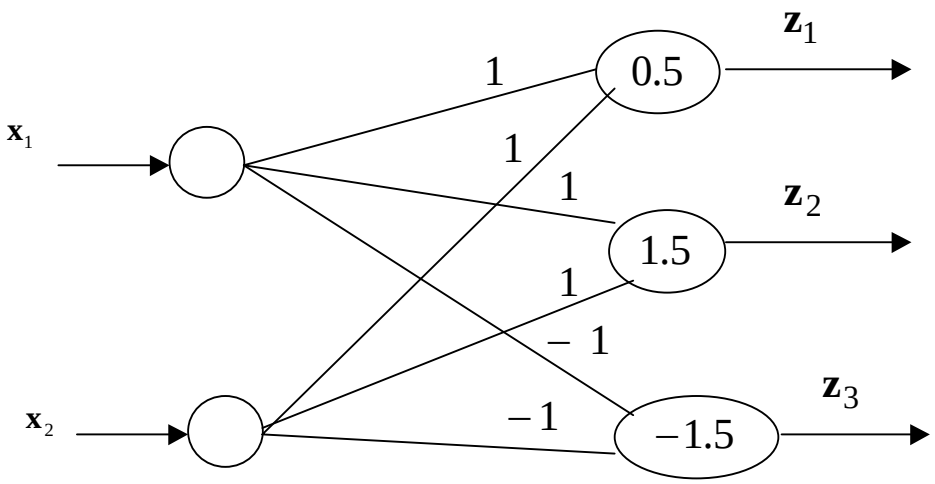
AND



NAND

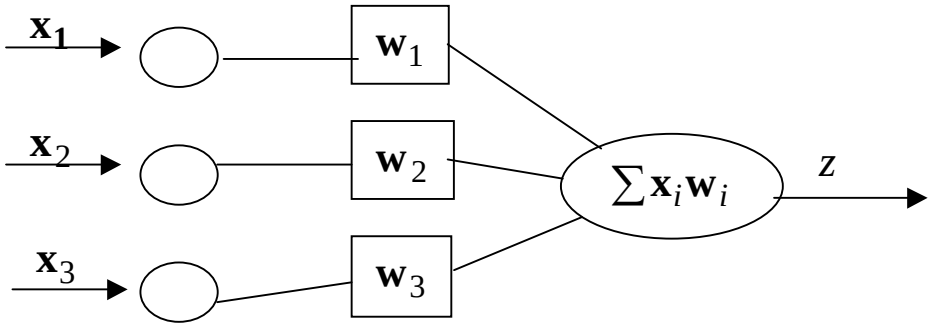


Agrupados en una sola red con tres salidas, una para cada categoría sería:



z_1 = 1 solamente cuando la entrada pertenece a la clase 1
z_2 = 1 solamente cuando la entrada pertenece a la clase 2
z_3 = 1 solamente cuando la entrada pertenece a la clase 3

EJERCICIOS DE REPASO



Dado el sumador lineal mostrado en la parte superior, donde la salida está dada por $\sum x_i w_i$ y dado los pares de entrenamiento (entradas y salidas)

x_1	x_2	x_3	z
1	0	1	1
0	0	1	0

La regla de Hebb podría generar los pesos correctos para construir las asociaciones anteriores?

Si __ _____

_____

No __ _____

_____

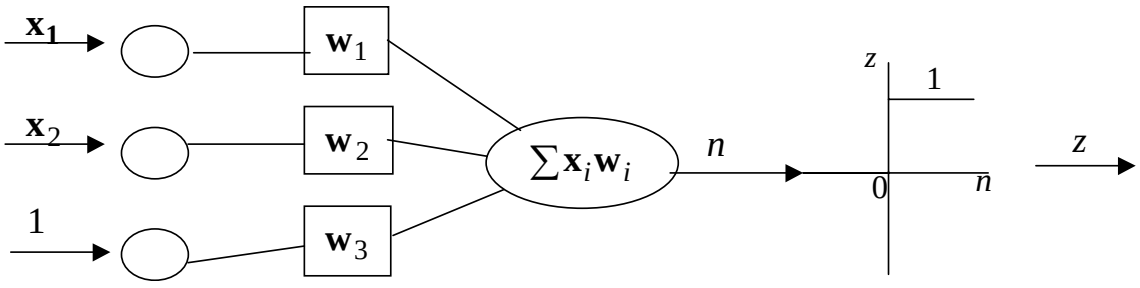
La Regla Delta podría generar los pesos correctos para construir las asociaciones anteriores?

Si __ _____

_____

No __ _____

_____



Con el apropiado conjunto de pesos, podría la red mostrada arriba asociar correctamente los siguientes patrones de entrada.

x_1	x_2	z
0.5	0.5	1
1.5	1.5	0
1	0.5	0
1.5	0.5	1
1	0.25	0

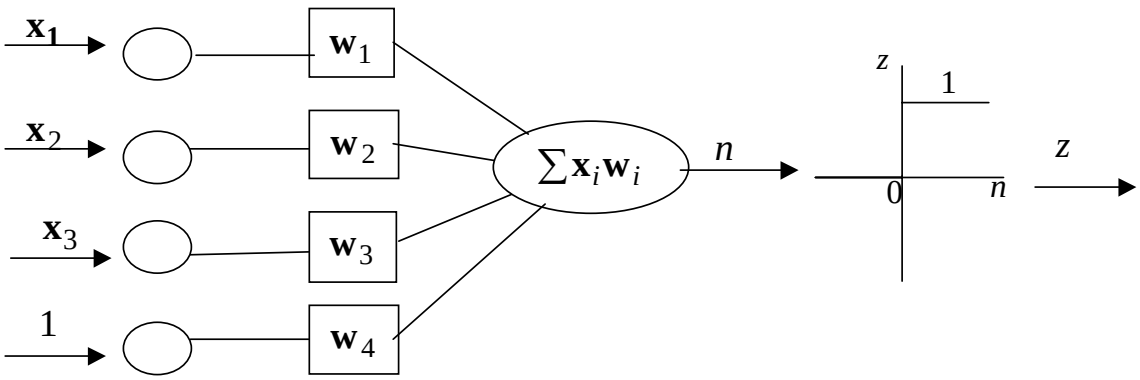
Si __ _____

_____

No __ _____

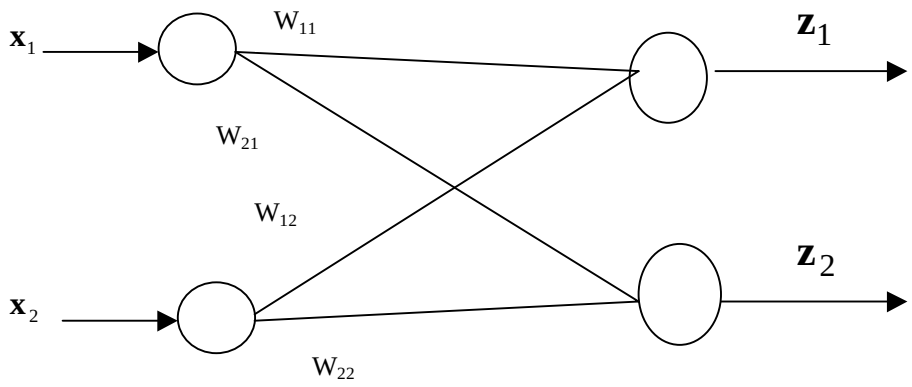
_____

Cómo estaría representada en un plano (x_1 x_2)



Use la Regla Delta para entrenar la red mostrada en la parte superior tal que haga la clasificación de los ejemplos mostrados abajo correctamente. Asuma una constante de aprendizaje de 0.1. Asuma los pesos iniciales w_1= 0.2, w_2 = 0.2, w_3 = 0, w_4 = 0.15. Muestre la secuencia de aprendizaje hasta su estabilidad.

x_1	x_2	x_3	z
1	0	0	1
1	1	1	0
0	0	0	0
1	0	1	1



Las dos unidades de entradas solamente distribuyen las señales de entrada y las unidades de salida son solamente sumadores lineales sin incluir ninguna función no lineal ni umbral (threshold)

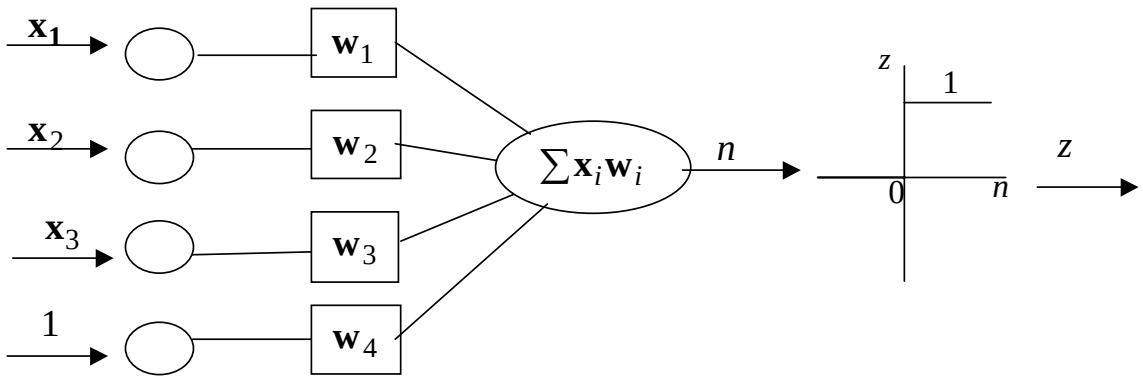
Si la matriz inicial de los pesos es $\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ y deseo asociar el vector de entrada $1/5\begin{bmatrix} 3 \\ 4 \end{bmatrix}$ con el vector de salida $\begin{bmatrix} 1 \\ 2 \end{bmatrix}$ y el vector de entrada $1/5\begin{bmatrix} 4 \\ -3 \end{bmatrix}$ con el vector de salida $\begin{bmatrix} 2 \\ 1 \end{bmatrix}$.

Cuál sería la matriz resultante si se aplica la regla de aprendizaje de Hebb.

$$\begin{bmatrix} & \\ & \end{bmatrix}$$

Cuál sería la matriz resultante si se aplica la regla de aprendizaje de Delta.

$$\begin{bmatrix} & \\ & \end{bmatrix}$$



Use la Regla Delta para entrenar la red mostrada en la parte superior tal que haga la clasificación de los ejemplos mostrados abajo correctamente. Asuma una constante de aprendizaje de 0.5. Asuma los pesos iniciales w_1= 0.5, w_2 = -0.2, w_3 = 0, w_4 = 0.15.

x_1	x_2	x_3	z
1	1	1	0
1	1	0	0
0	0	0	0
1	0	1	1

Al aplicar la regla Delta, cuáles son los valores de los pesos después que el primer ejemplo de entrada es aplicado la primera vez y se hayan realizado las correcciones.

Cuáles son los pesos en el último epoch (época).