Perceptrón

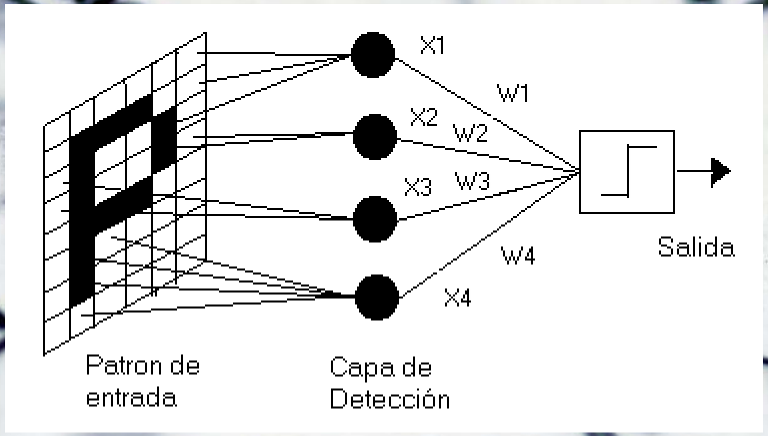
PRESENTA:

M. EN C. FRANCISCO RAFAEL SUÁREZ RUIZ

Objetivo

Comprender el funcionamiento del elemento mas básico de una red neuronal.

Introducción



Frank Rosenblatt inventó el perceptrón en 1957 en el Laboratorio de Aeronáutica de Cornell en un intento de comprender la memoria humana, el aprendizaje, y el proceso cognoscitivo.

En junio de 1960, demostró la MARK I Perceptrón, la primera máquina que podía aprender a reconocer e identificar patrones ópticos.

Características

Características

- Aprendizaje supervisado, por corrección de error.
- Clasificación de patrones linealmente separables.
- Reconocimiento de patrones sencillos. (compuertas lógicas, Caracteres impresos).
- Entradas y salidas digitales.

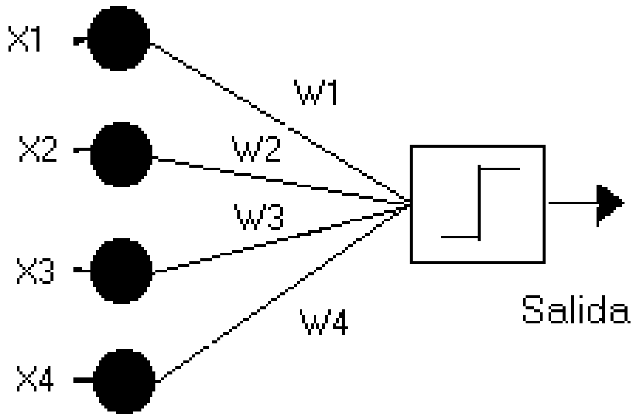
Arquitectura

Sin capa oculta

Perceptrón simple con un sola neurona

Perceptrón múltiple con varias neuronas en su salida

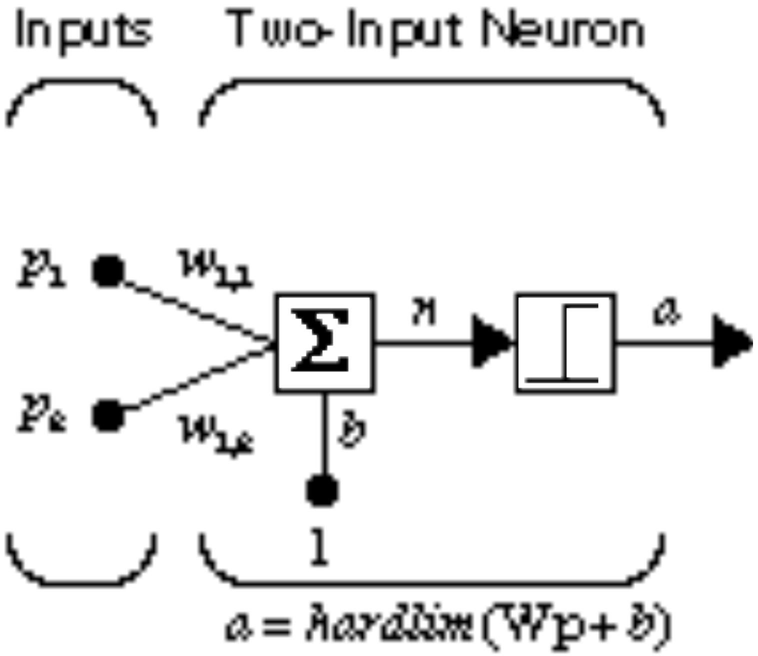
Función de transferencia en escalón .



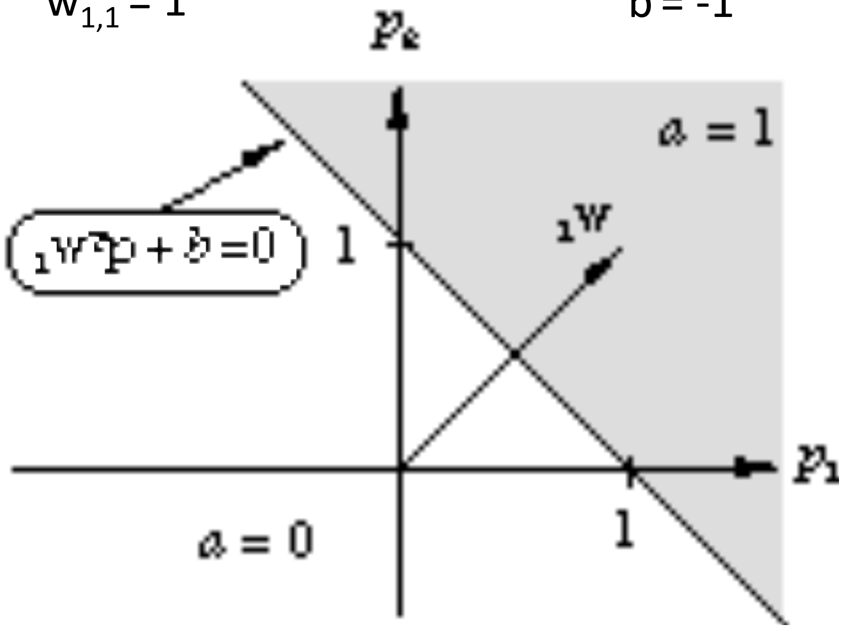
Características perceptrón

- Aprendizaje supervisado, por corrección de error.
- Clasificación de patrones linealmente separables.
- Reconocimiento de patrones sencillos. (compuertas lógicas, Caracteres impresos)
- Entradas y salidas digitales

Perceptrón simple



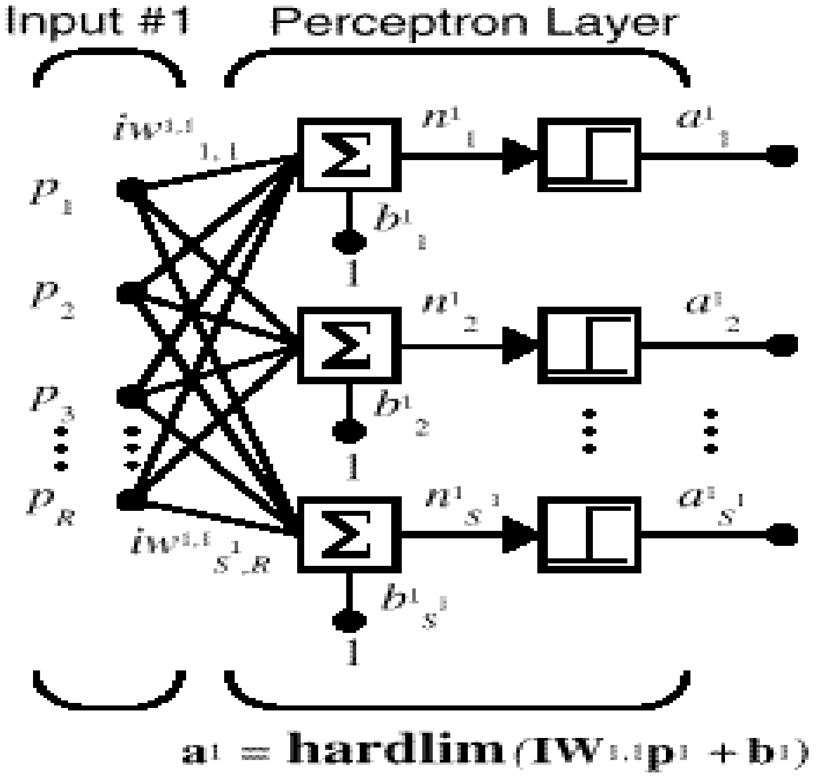
$w_{1,1} = 1$ $w_{1,2} = 1$ $b = -1$



$$a = hardlim({}_1\mathbf{w}^T\mathbf{p} + b) = hardlim(w_{1,1}p_1 + w_{1,2}p_2 + b)$$

Perceptrón Múltiple

Cada neurona introduce una frontera de decisión.



Analysis of Perceptron Network

Geometric properties (single output):

$$y = f(\boldsymbol{w}^T\boldsymbol{x}) = f\left(\sum_{i=0}^{n} w_i\, x_i\right)$$

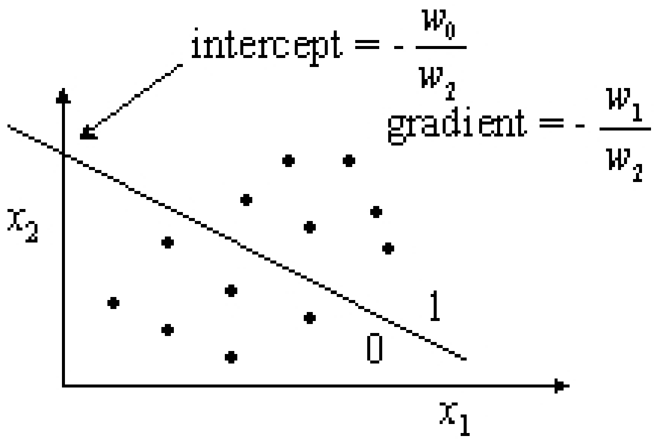
where *f()* is a binary transfer function.

When is $\boldsymbol{w}^T\boldsymbol{x} = 0$?

For 2 inputs:

$$w_0 + w_1x_1 + w_2x_2 = 0$$

$$x_2 = -\frac{w_1}{w_2}x_1 - \frac{w_0}{w_2}$$



Weight vector forms a **decision boundary** in the input space.

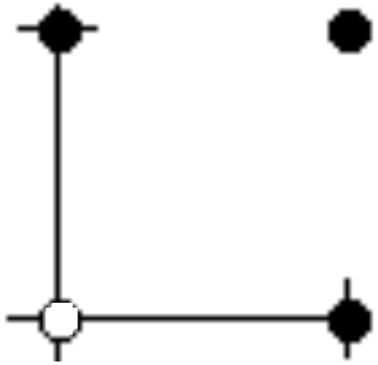
Ejemplo Compuerta OR

$$\left\{\mathbf{p}_1 = \begin{bmatrix}0\\0\end{bmatrix}, t_1 = 0\right\}$$

$$\left\{\mathbf{p}_4 = \begin{bmatrix}1\\1\end{bmatrix}, t_4 = 1\right\}$$

$$\left\{\mathbf{p}_2 = \begin{bmatrix}0\\1\end{bmatrix}, t_2 = 1\right\}$$

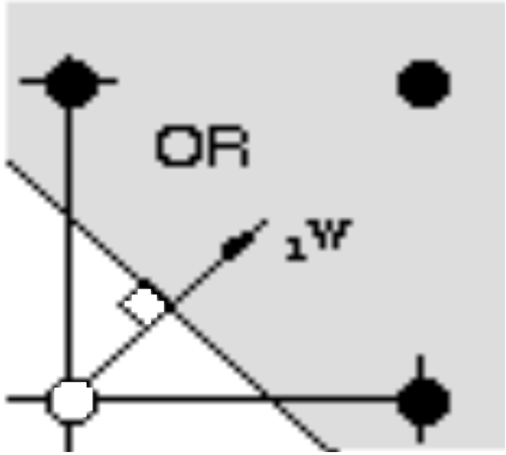
$$\left\{\mathbf{p}_3 = \begin{bmatrix}1\\0\end{bmatrix}, t_3 = 1\right\}$$



Solución

El vector de pesos debe ser ortogonal a la frontera de decisión.

$$ {}_1\mathbf{w} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} $$



Seleccionar un punto sobre la frontera de decisión para encontrar el umbral.

$$ {}_1\mathbf{w}^T\mathbf{p} + b = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} + b = 0.25 + \mathrm{b} = 0 \quad \Rightarrow \quad b = -0.25 $$

Regla de aprendizaje

Regla de Aprendizaje del Perceptrón

El entrenamiento comienza asignándole valores iniciales pequeños (aleatorios) a los parámetros de la red (W y b).

$$W^{new} = W^{old} + ep^{T}$$

$$b^{new} = b^{old} + e$$

Regla de Aprendizaje del Perceptrón

Donde:

W es el peso sináptico.

P es el patrón de entrada, o elementos de entrada.

b es el umbral de activación de la neurona

e es el error de salida de la neurona al introducir un patrón P.

Regla de Aprendizaje del Perceptrón

Donde:

e es el error de salida de la neurona al introducir un patrón P.

t es el valor objetivo asignado para un determinado patrón.

a es la salida de la red.

$$e = t - a$$

Algoritmo de aprendizaje

1. Se aplica un patrón de entrada Pi.

2. Se obtiene la salida (a) al aplicar la regla de propagación:

$$a = hardlim({}_1\mathbf{w}^T\mathbf{p} + b) = hardlim(w_{1,1}p_1 + w_{1,2}p_2 + b)$$

3. Se calcula el error al obtener la diferencia entre el valor objetivo (t) y el entregado por la red.

$$e = t - a$$

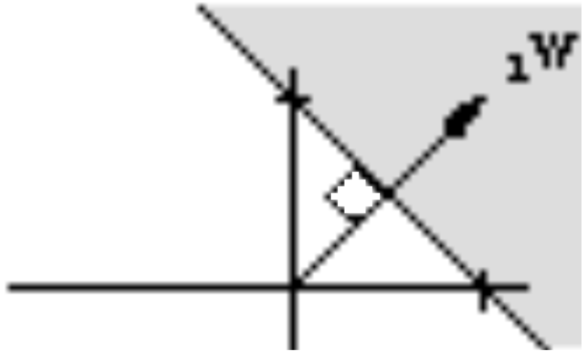
Algoritmo de aprendizaje

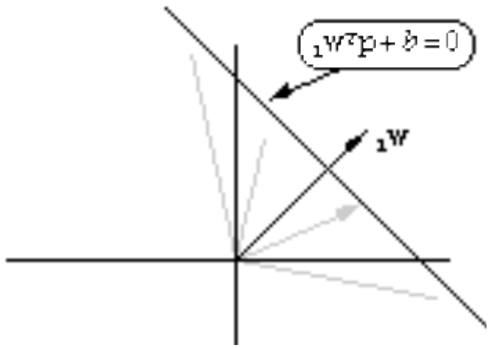
4. Se actualizan los pesos W y en su caso el umbral b. Siguiendo las reglas:

$$W^{new} = W^{old} + ep^T \qquad b^{new} = b^{old} + e$$

5. Se repiten pasos 1 a 4 con todos los vectores de entrada.

6. Si el Error es cero para cada uno de los patrones, detenerse, si no repetir algoritmo.





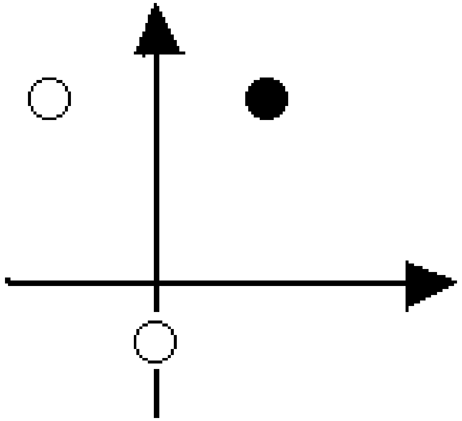
Frontera de Decisión (F)

- Todos los puntos sobre la frontera de decisión tienen el mismo producto punto respecto al vector de pesos.
- Por lo que ellos tienen la misma proyección sobre el vector de pesos y deben caer en la línea ortogonal al vector de pesos.

Ejemplo 1

Ejemplo 1

Clasifique el siguiente conjunto de patrones linealmente separables empleando un Perceptrón Simple y su regla de aprendizaje.



$$\left\{\mathbf{p}_1 = \begin{bmatrix}1\\2\end{bmatrix}, t_1 = 1\right\}$$

$$\left\{\mathbf{p}_2 = \begin{bmatrix}-1\\2\end{bmatrix}, t_2 = 0\right\}$$

$$\left\{\mathbf{p}_3 = \begin{bmatrix}0\\-1\end{bmatrix}, t_3 = 0\right\}$$

Calcule W y B que solucionan el problema.

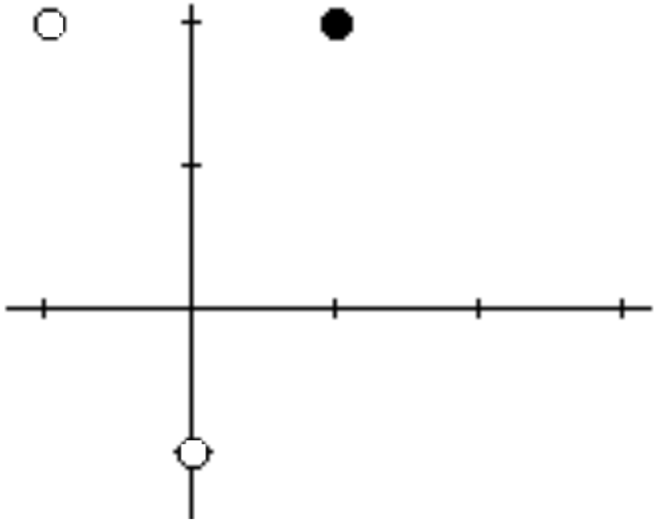
Grafique los patrones Pi, el vector de pesos Wf y la frontera F.

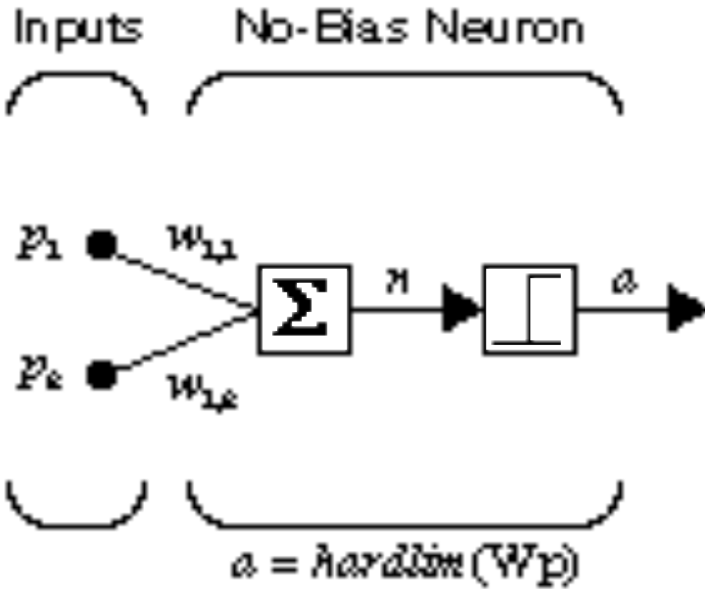
Escriba el código en NNT

Regla de Aprendizaje Problema de Prueba

$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$

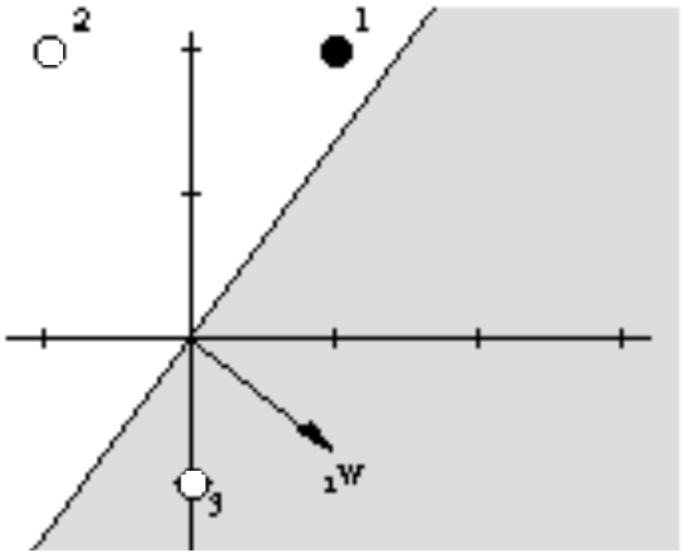
$$\left\{\mathbf{p}_1 = \begin{bmatrix}1\\2\end{bmatrix}, t_1 = 1\right\} \quad \left\{\mathbf{p}_2 = \begin{bmatrix}-1\\2\end{bmatrix}, t_2 = 0\right\} \quad \left\{\mathbf{p}_3 = \begin{bmatrix}0\\-1\end{bmatrix}, t_3 = 0\right\}$$





Inicio

$$ {}_1\mathbf{w} = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} $$

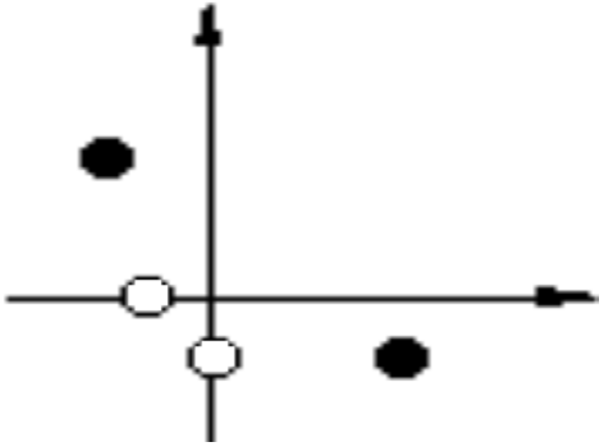


Presentar P1 a la red:

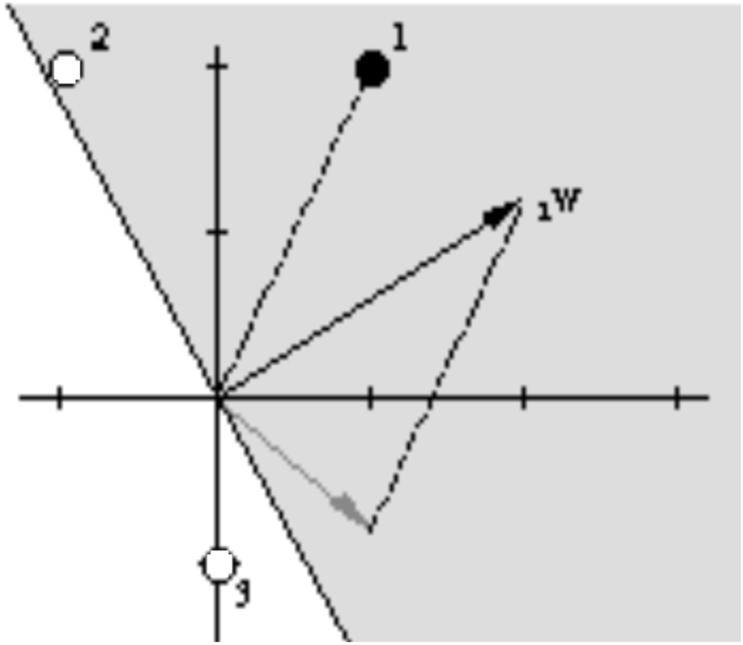
$$ a = hardlim({}_1\mathbf{w}^T\mathbf{p}_1) = hardlim\left(\begin{bmatrix} 1.0 & -0.8 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right) $$

$$ a = hardlim(-0.6) = 0 $$

Regla de aprendizaje tentativa



$$ {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix} $$



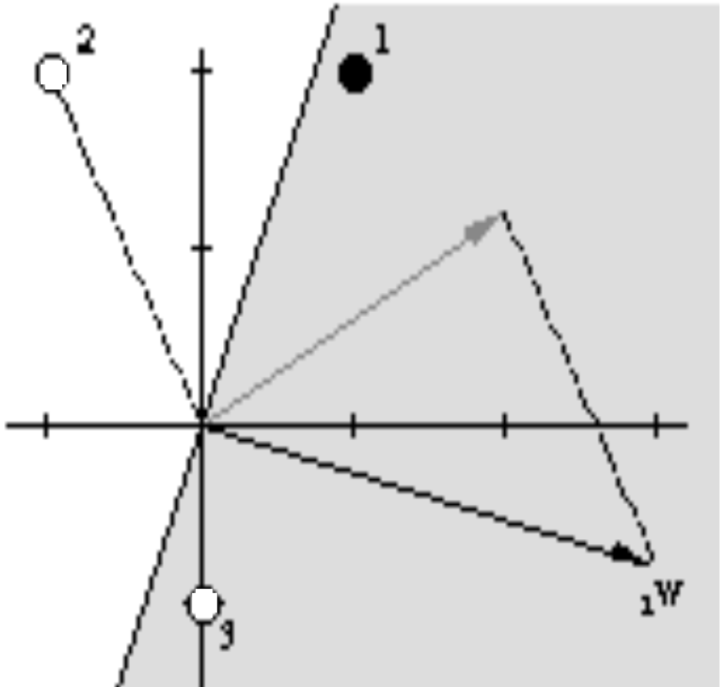
Segundo Vector de entrada

$$a = hardlim({}_1\mathbf{w}^T \mathbf{p}_2) = hardlim\left(\begin{bmatrix} 2.0 & 1.2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right)$$

$$a = hardlim(0.4) = 1$$

If $t = 0$ and $a = 1$, then ${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}$

$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}_2 = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix} - \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix}$$



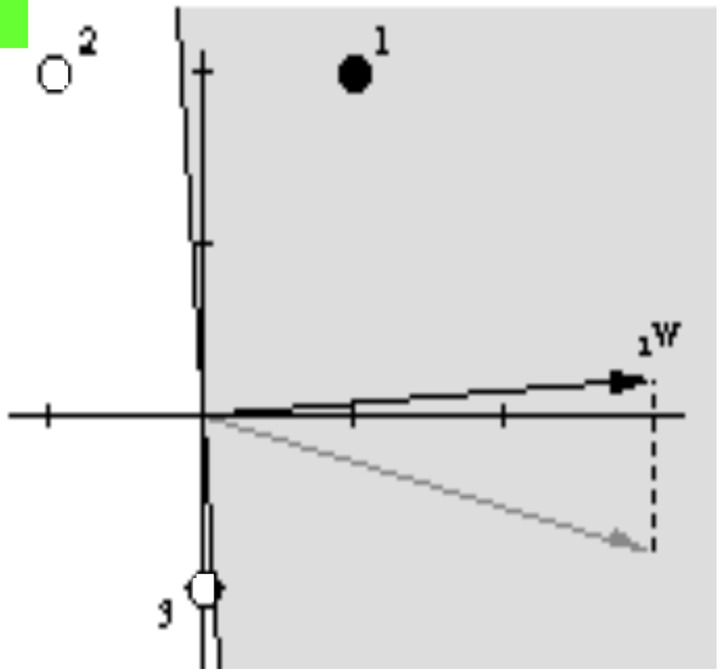
Tercer Vector de entrada

$$a = hardlim({}_1\mathbf{w}^T \mathbf{p}_3) = hardlim\left(\begin{bmatrix} 3.0 & -0.8 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix}\right)$$

$$\mathrm{a} = \mathrm{hardlim}(0.8) = 1$$

$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}_3 = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix} - \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 3.0 \\ 0.2 \end{bmatrix}$$

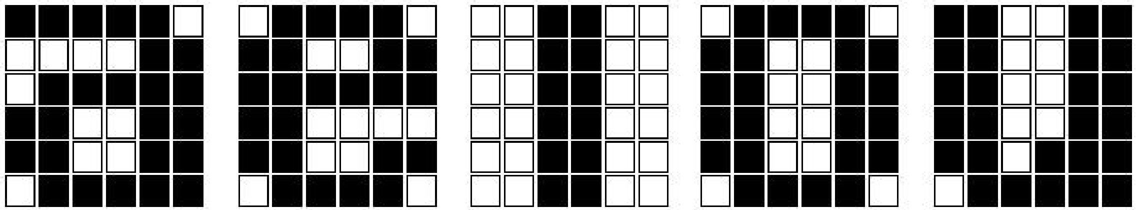
$$\text{If } t = a, \text{then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old}.$$

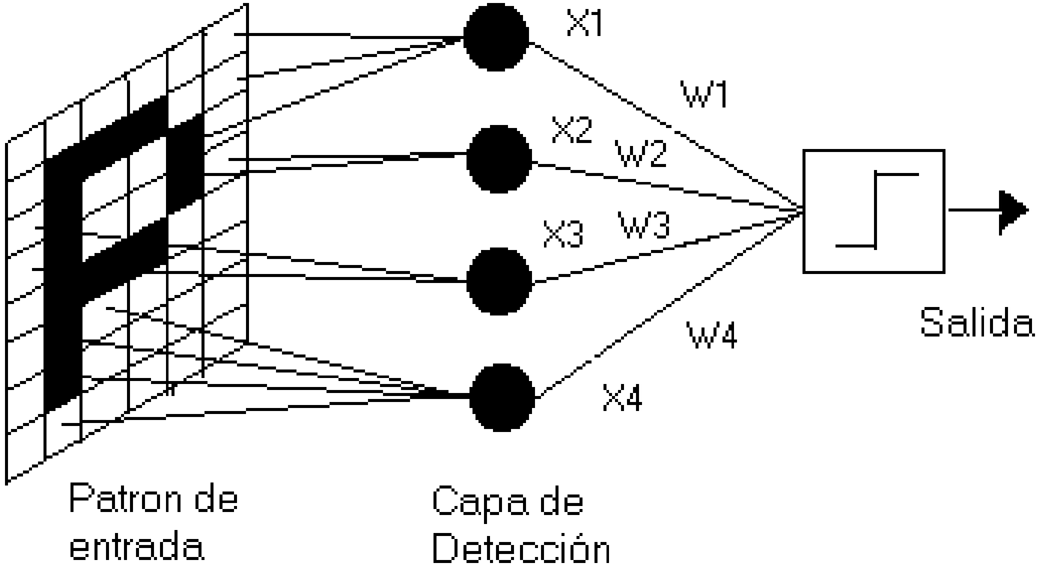


Aplicación

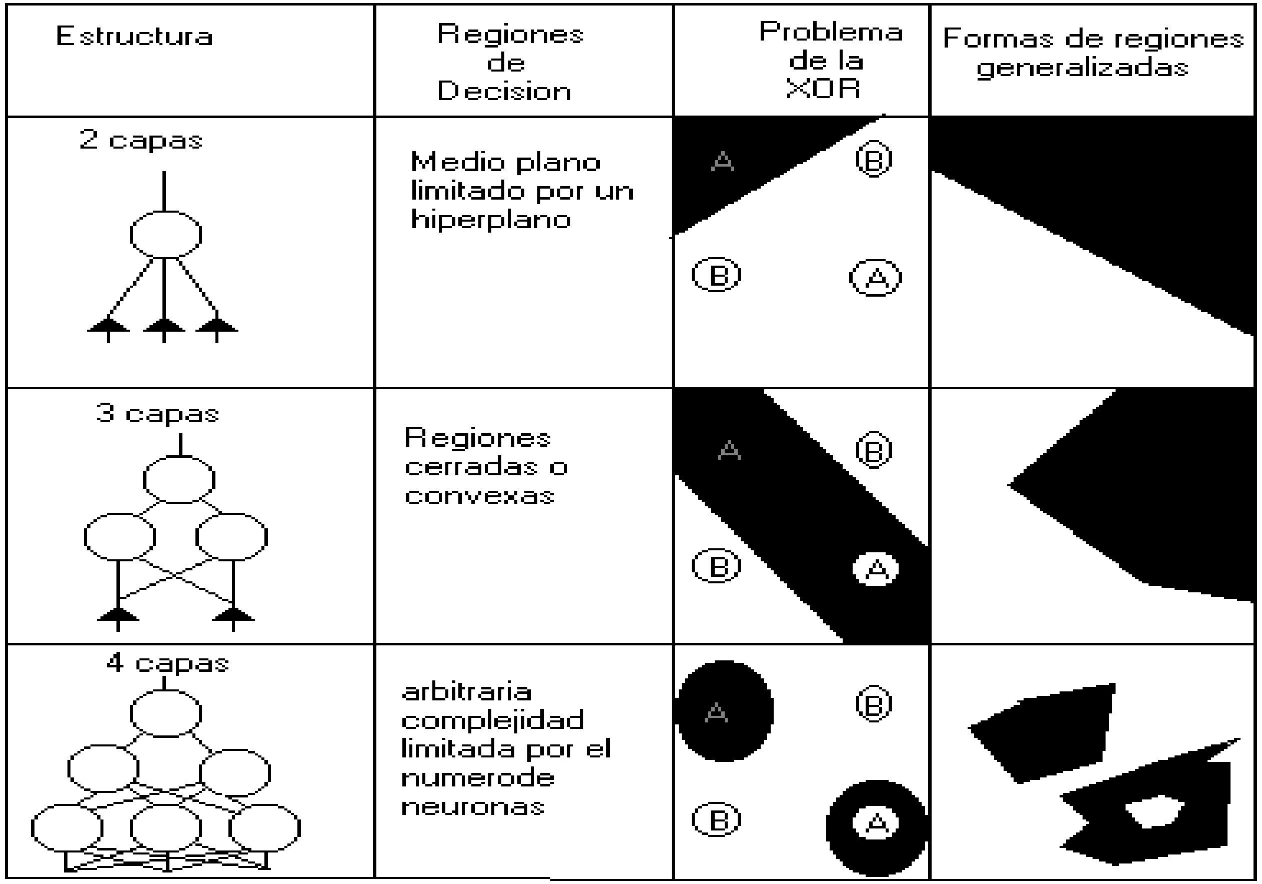
Reconocimiento de patrones linealmente separables

OCR





Regiones de Clasificación con el Perceptrón Multicapa



Estructura	Regiones de Decision	Problema de la XOR	Formas de regiones generalizadas
2 capas	Medio plano limitado por un hiperplano	A B B A	
3 capas	Regiones cerradas o convexas	A B B A	
4 capas	arbitraria complejidad limitada por el numerode neuronas	A B B A	

Distintas formas de las regiones generadas por un perceptron multinivel.

Links de interés

https://rpubs.com/Joaquin_AR/530875