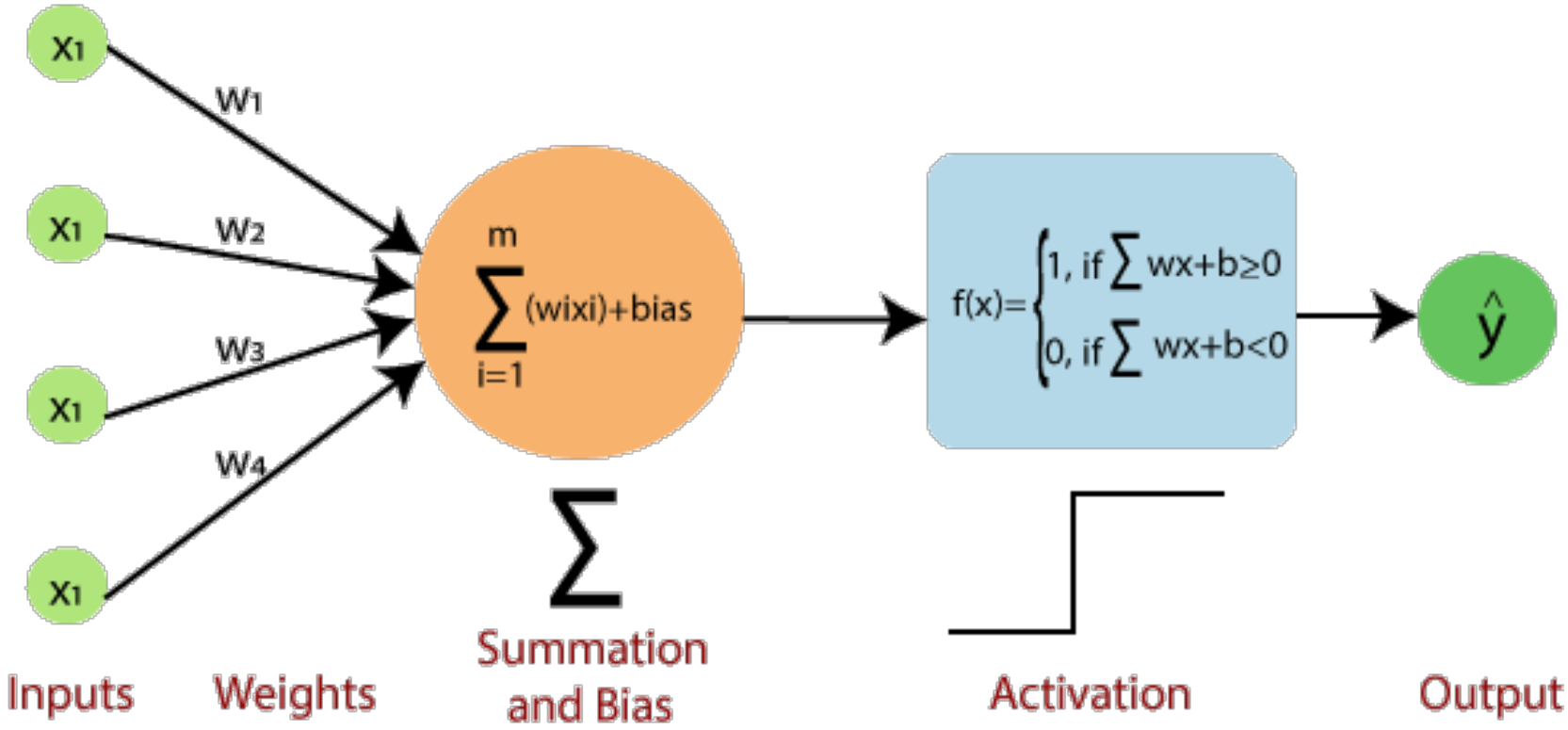
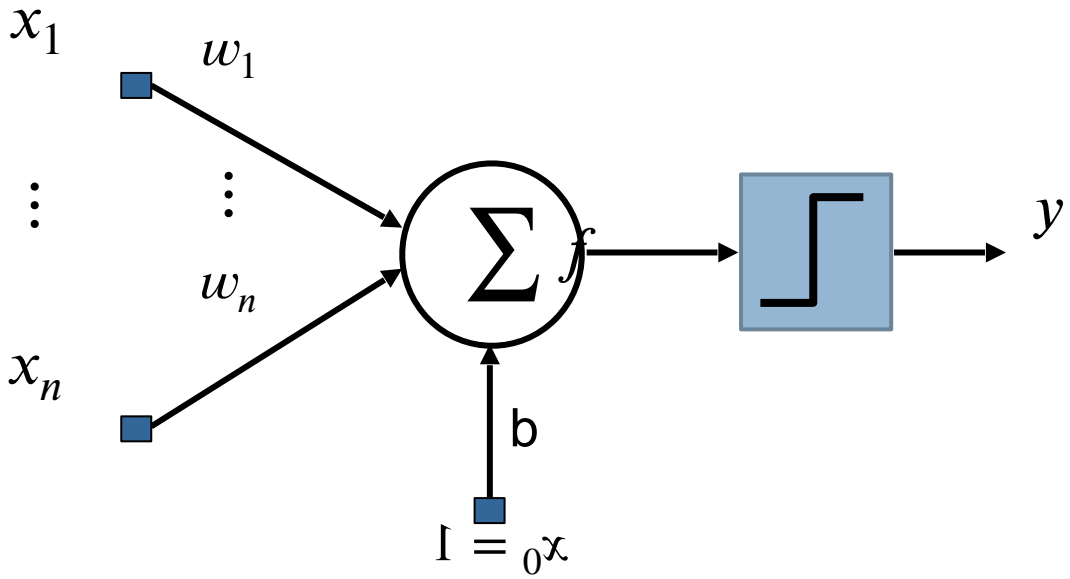
El Perceptrón



El Perceptrón

- Estructura del Perceptrón
- El Perceptrón como clasificador de dos clases
- El Perceptrón como operador lógico
- Limitaciones del Perceptrón
- Regla de aprendizaje
- Ejemplos

Estructura del perceptrón



$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

El umbral *b (bias)* puede verse como un peso entre la unidad de entrada y una señal ficticia de valor $x_0 = 1$

Estructura del perceptrón

La suma pesada de las entradas se aplica a la función de activación signo o escalón.

$$sgn(x) = \begin{cases} 1 & if\ x \geq 0 \\ -1 & otherwise \end{cases}$$

$$step(x) = \begin{cases} 1 & if\ x \geq 0 \\ 0 & otherwise \end{cases}$$

Modelo matemático del perceptrón

- Salida de la neurona

$$y = hardlim\left(\sum_{i=1}^{n} w_i x_i + b \right)$$

- Forma vectorial

b incluida

$$y = hardlim\left(w^T x \right)$$

El Perceptron Clasificador de dos Clases

Clasificador de dos clases

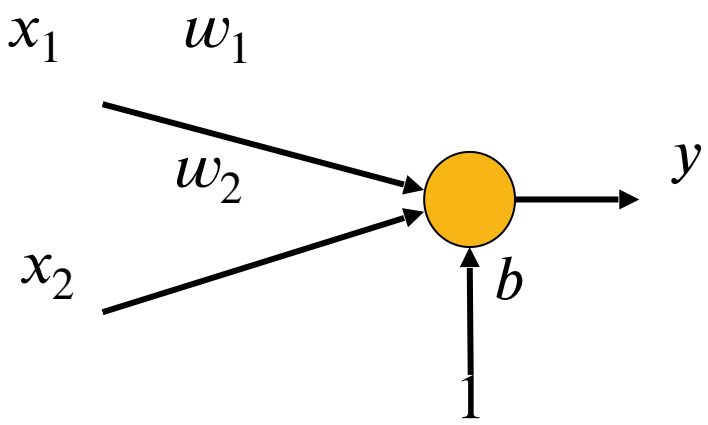
- El propósito del perceptrón es clasificar las entradas, $x_1, x_2, \ldots, x_n$ en una de dos clases, digamos A_1 y A_2.

- Los patrones de entrada pertenecen a una de dos clases.

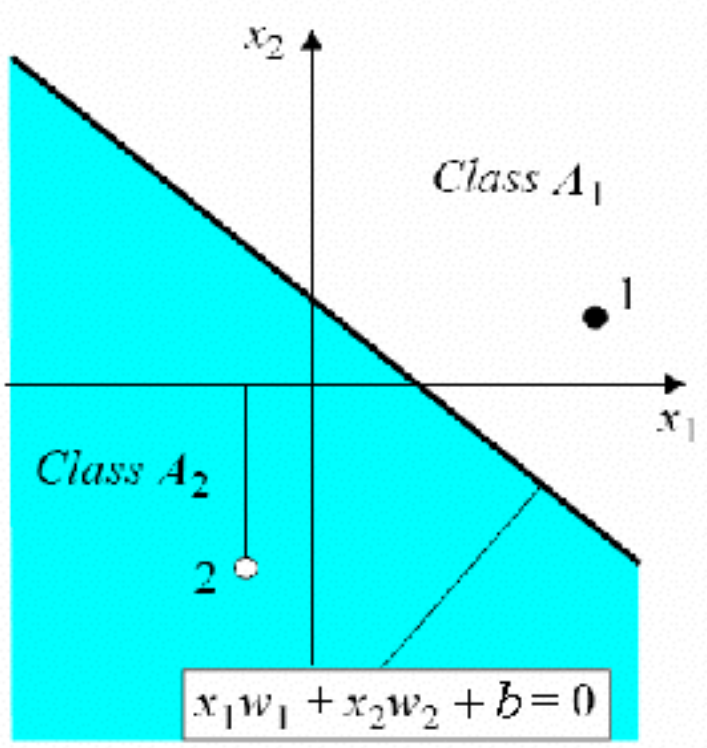
Esto sólo puede suceder cuando ellos son linealmente separables

Perceptrón con dos entradas

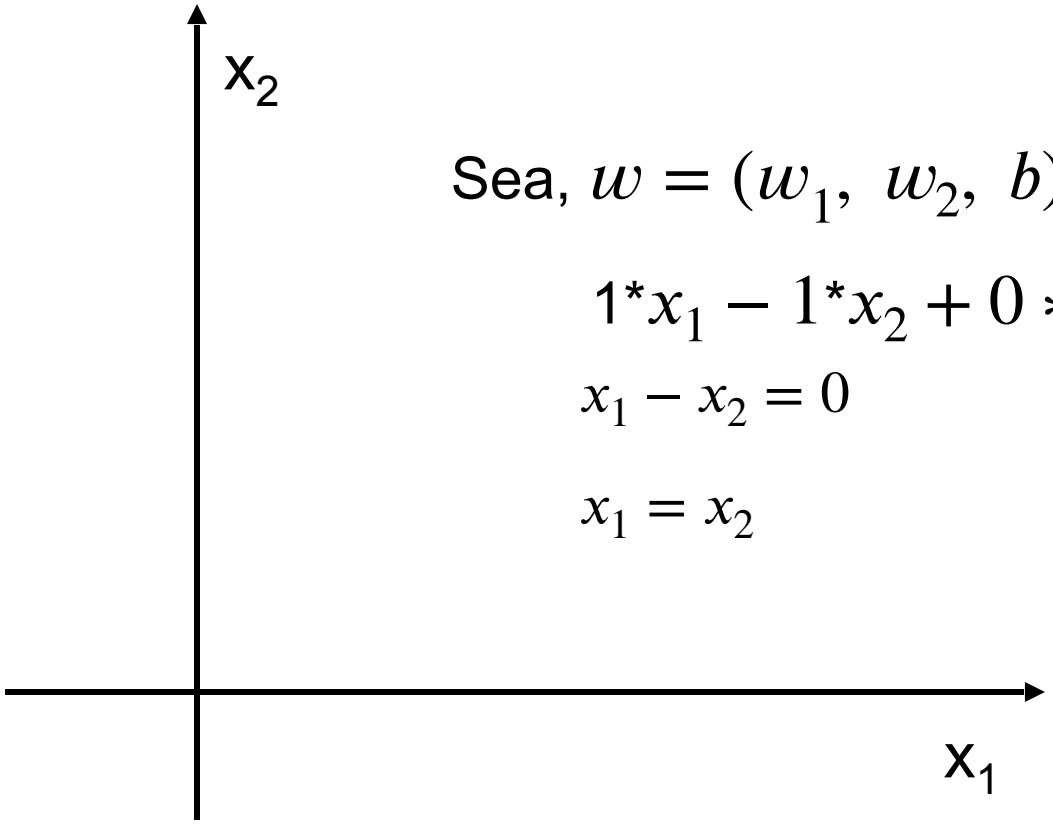
La frontera de decisión esta determinada por,



$$y = f(w_1 x_1 + w_2 x_2 + b)$$



La frontera de decisión en el perceptrón



Sea, $w = (w_1,\ w_2,\ b)$ = (1, -1, 0)

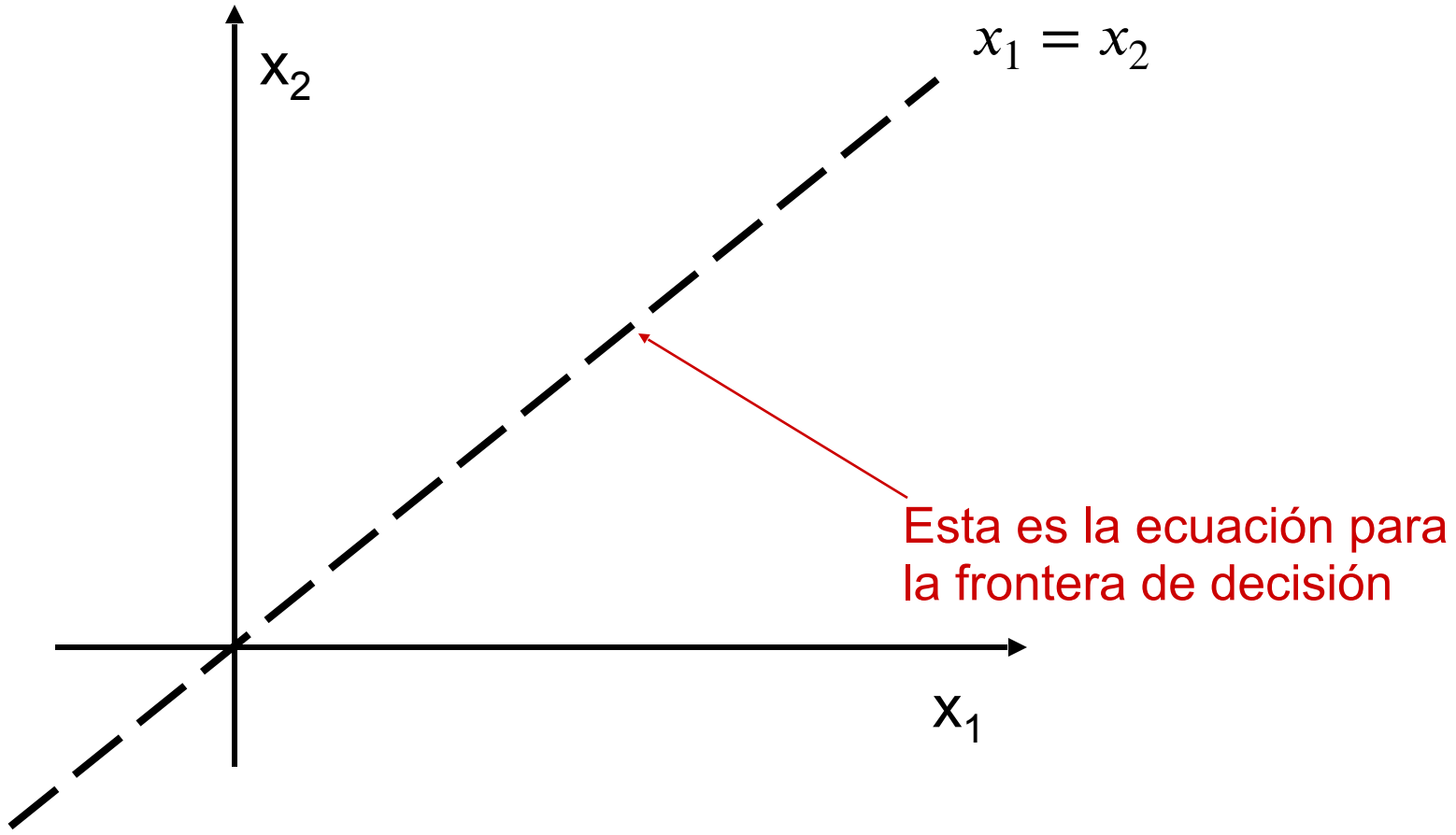
$$1^{*}x_1 - 1^{*}x_2 + 0 * 1 = 0$$

$$x_1 - x_2 = 0$$

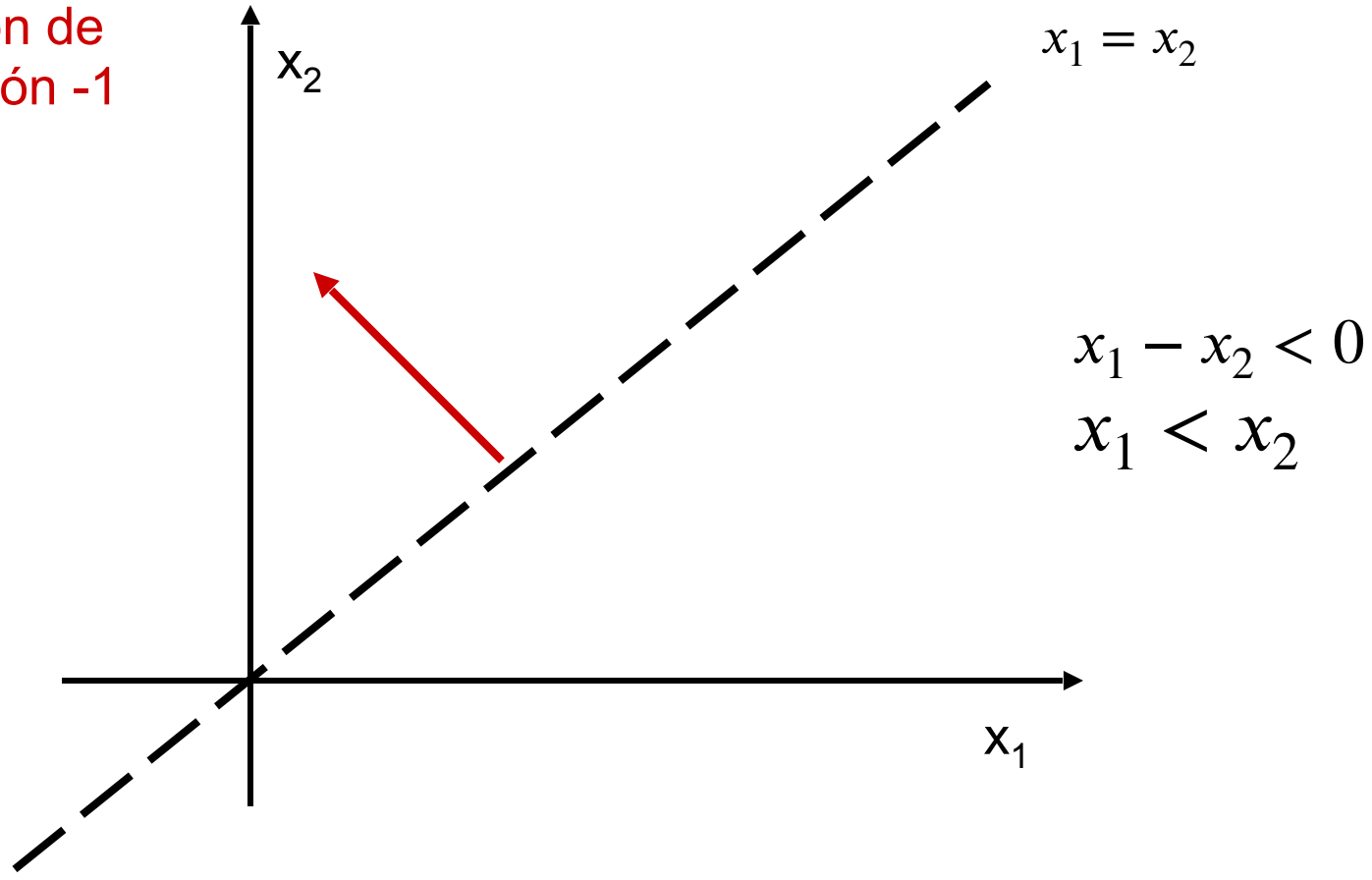
$$x_1 = x_2$$

$$w_1 x_1 + w_2 x_2 + b = 0$$

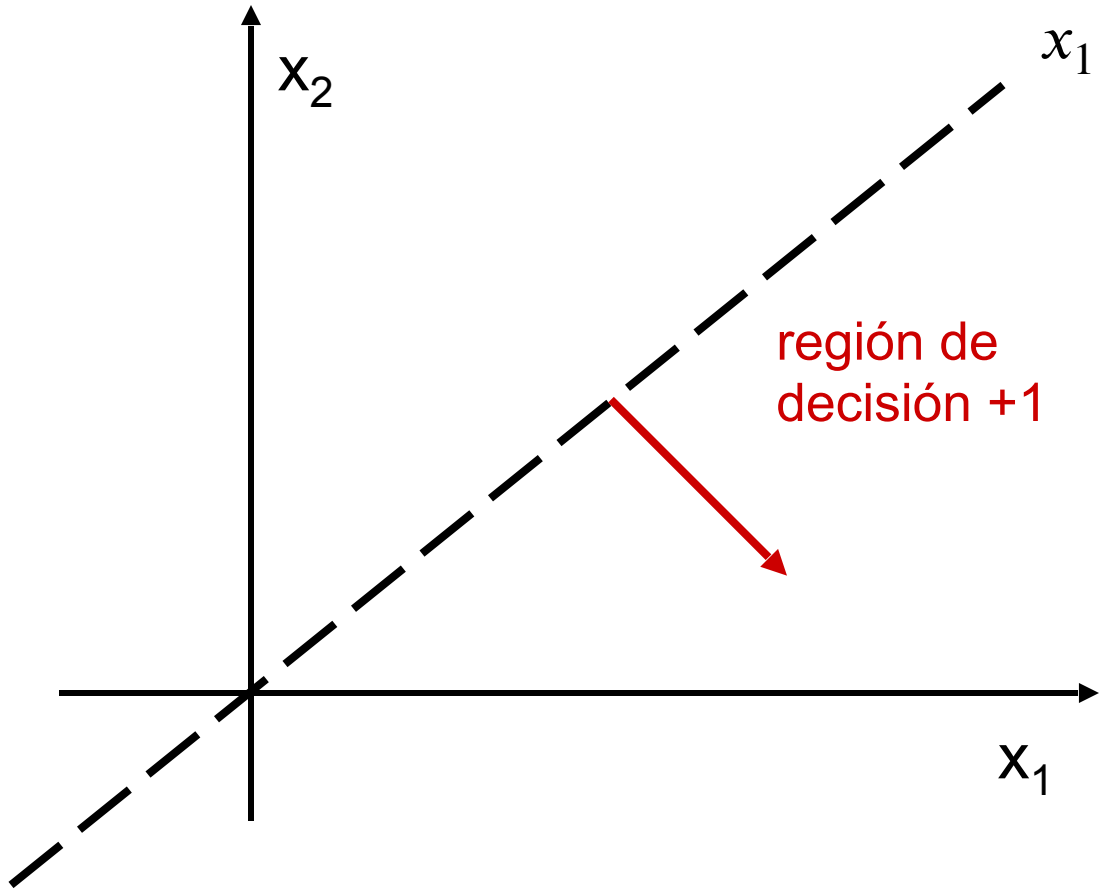
La frontera de decisión en el perceptrón



La frontera de decisión en el perceptrón



La frontera de decisión en el perceptrón

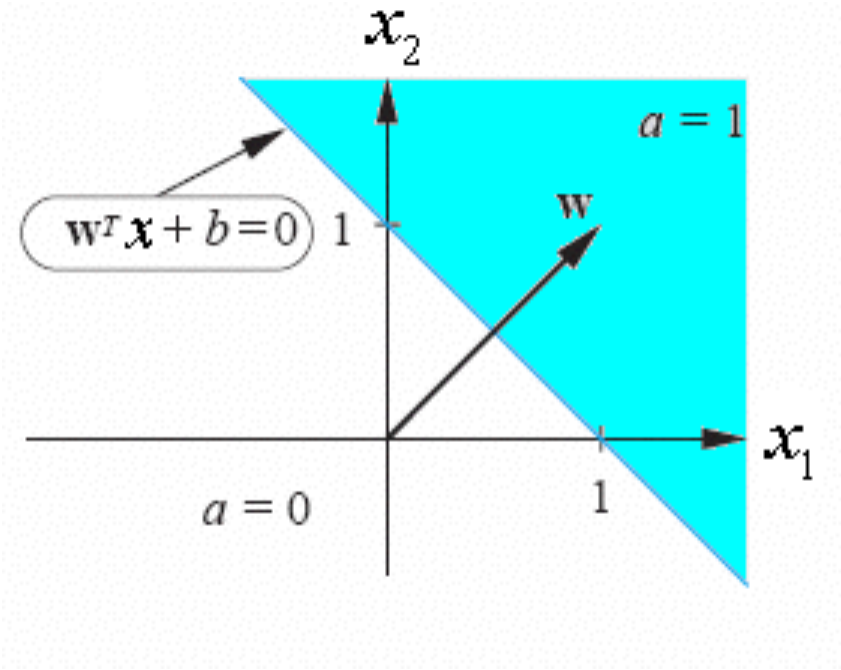


$$x_1 - x_2 > 0$$
$$x_1 > x_2$$

Otro Ejemplo numérico

- La frontera de decisión con

$$w_1 = 1$$
$$w_2 = 1$$
$$b = -1$$

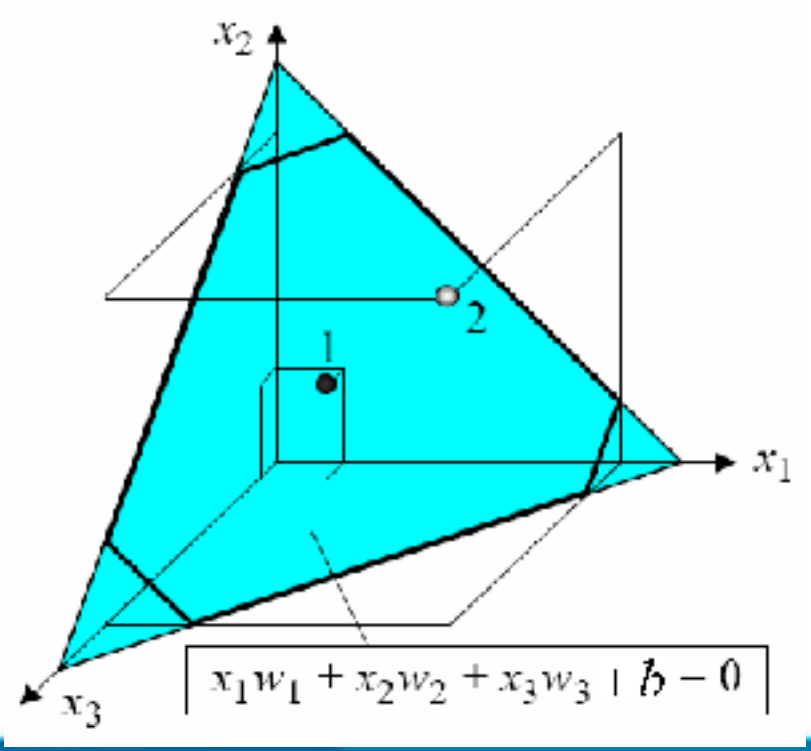


$$w_1 x_1 + w_2 x_2 + b = 0$$

El perceptrón con tres entradas

- La frontera de decisión esta determinada por,

$$w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 0$$



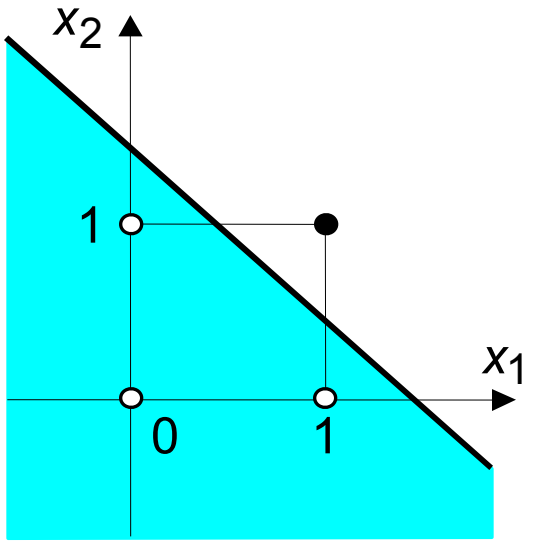
El Perceptrón como un Operador Lógico

El perceptrón como operador lógico

- El perceptrón como una AND lógica

$$w = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$b = -3$$



x	y	x & y
0	0	0
0	1	0
1	0	0
1	1	1

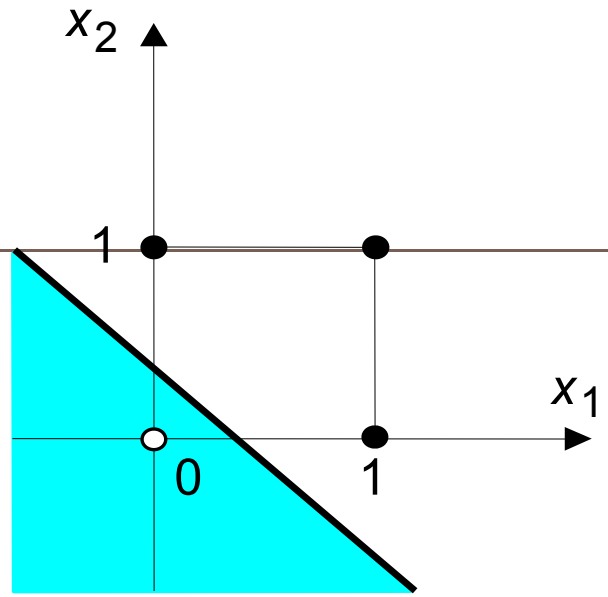
$$w_1 x_1 + w_2 x_2 + b = 0$$

El perceptrón como operador lógico

- El perceptrón como un OR lógica

$$w = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$b = -1$$



x	y	x | y
0	0	0
0	1	1
1	0	1
1	1	1

$$w_1 x_1 + w_2 x_2 + b = 0$$

Limitaciones

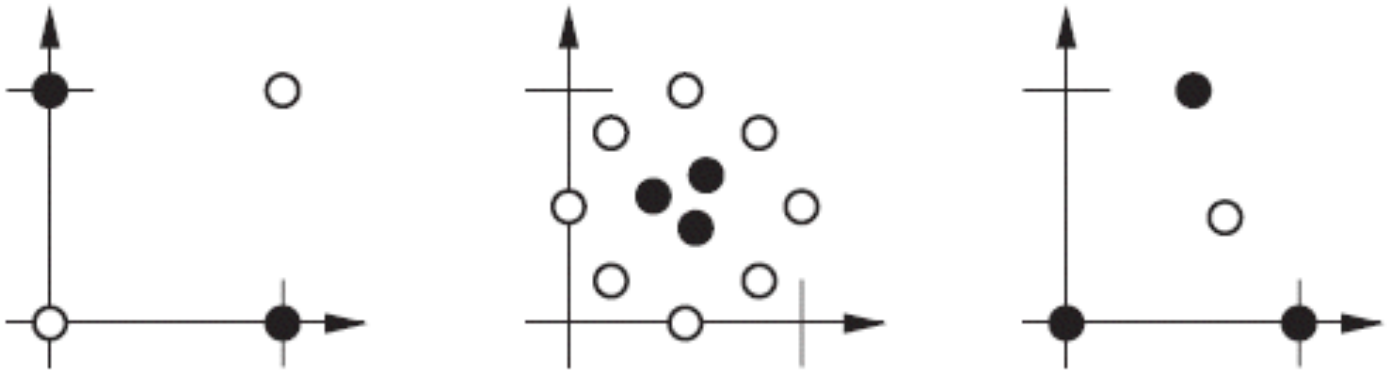
- Las dos clases deben ser linealmente separables.

- Se garantiza la convergencia de la regla de aprendizaje del perceptron a una solución en un numero finito de pasos, siempre que exista una solución.

Limitaciones del Perceptrón

Limitaciones

Problemas no linealmente separables



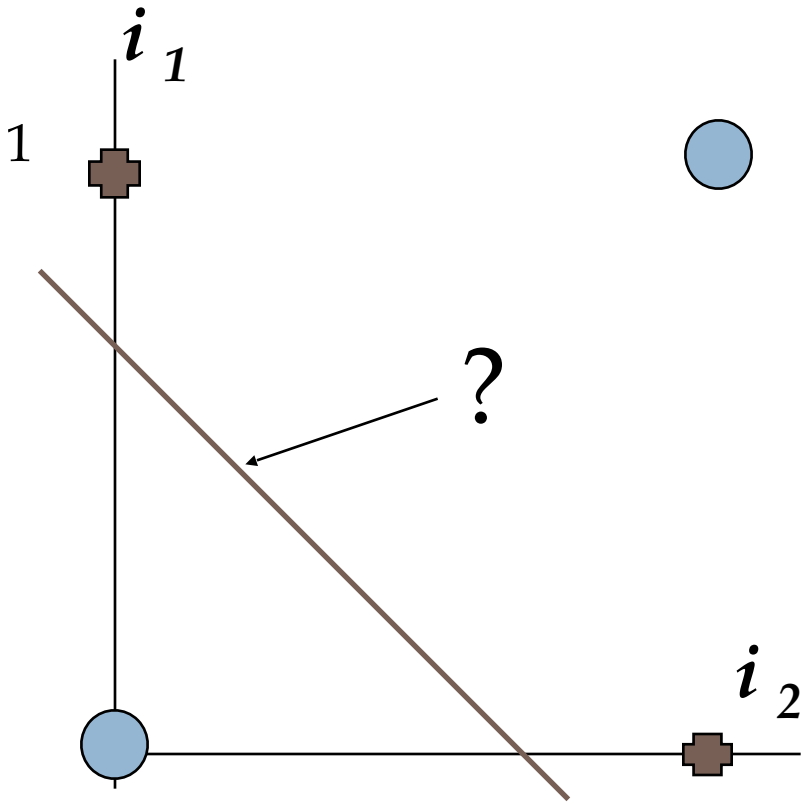
Trate de dibujar una línea recta entre vectores de dos clases

El problema de la OR exclusiva (XOR)

Propósito: clasificar un vector binario de entrada en la clase 0 si el vector tiene un numero par de 1's o 0's, en caso contrario asignarlo a la clase 1.

0	0	0
0	1	1
1	0	1
1	1	0

Objecion de Minsky-Papert



La simple operación de la ***OR exclusiva*** (XOR)
no puede ser resuelta usando un perceptron lineal con polarización.

Por lo tanto, probablemente problemas más importantes no pueden ser resueltos con el perceptron lineal.

El problema XOR resuelto

Conclusión:

Por medio de estructuras multicapas es posible clasificar problemas que no son linealmente separables.

Regla de Aprendizaje del Perceptrón

Aprendizaje en las RN artificiales

- Por regla de aprendizaje entendemos un procedimiento para modificar los pesos de una red.
- Este procedimiento se denomina también algoritmo de entrenamiento (supervisado, no supervisado).
- El propósito de la regla de aprendizaje es entrenar la red para que ejecute una tarea.

Aprendizaje supervisado

A la regla de aprendizaje se le proporciona un conjunto de ejemplos (el conjunto de entrenamiento) de la conducta apropiada de la red.

$$\{p_1, t_1\}, \{p_2, t_2\}, \cdots, \{p_n, t_n\}$$

p_k : Entrada a la red

t_k : Salida correcta (target)

Regla de aprendizaje del perceptrón

- Si en la iteración k, la salida de la red es $y(k)$ y la salida deseada es $t(k)$, entonces el error esta dado por:

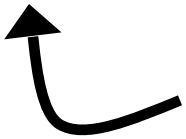
$$e(k) = t(k) - y(k)$$

- La iteración k se refiere aquí al k-esimo ejemplo de entrenamiento presentado al perceptrón.

Regla de aprendizaje del perceptrón

- La regla de aprendizaje se usa para ajustar los pesos de la red para mover las salidas de la red hacia las salidas correctas (targets).

$$w(k+1) = w(k) + \Delta w$$



Los pesos se ajustan de acuerdo al error

Regla de aprendizaje del perceptrón

$$w(k+1) = w(k) + \eta\, e(k)\, p(k)$$

- η: Velocidad de aprendizaje (eta)

- Si el error, $e(k) = t(k) - y(k)$, es positivo, se necesita incrementar la salida del Perceptrón $y(k)$, pero si es negativo, se necesita decrementar $y(k)$.

Algoritmo de entrenamiento

- Paso 1: Inicialización

  Seleccionar los pesos iniciales $w_1, w_2, \ldots, w_m$ y el umbral b con números aleatorios.

- Paso 2: Activación

  Presentar las entradas al Perceptrón $p_1(k), p_2(k), \ldots, p_m(k)$. Calcular la salida de la red en la iteración $k = 1$

$$y(k) = sgn\left[\sum_{i=1}^{m} p_i(k)w_i(k) + b\right]$$

Algoritmo de entrenamiento

- Paso 3: Actualización de los pesos

$$w(k+1) = w(k) + \eta\, e(k)\, p(k)$$

- Paso 4: Iteración

  Incrementar k en uno, volver al Paso 2 y repetir el proceso hasta la convergencia.

Ejemplo regla de aprendizaje

P		T
2	2	0
1	-2	1
-2	2	0
-1	1	1

$w(0) = [0 \quad 0]$ $\qquad$ $b(0) = 0$

Ejemplos

Un perceptrón en Matlab

- Crear un perceptron

  net = perceptron

- Entrenamiento de la red

  net = train(net, P, T);

- Simulación

  y = net(P)

¿ Preguntas ?