Inteligencia Artificial
Unidad 3
Redes Neuronales

INTRODUCCIÓN

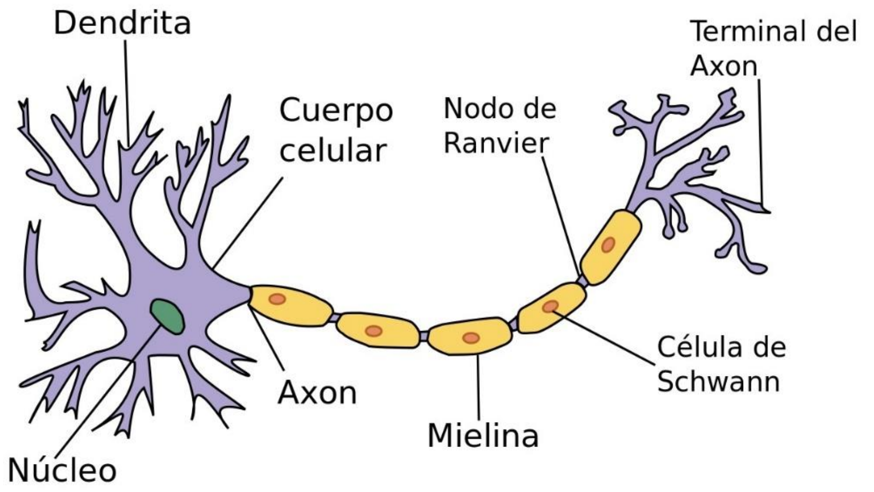
- Hugo David Calderón
- Heider Sanchez Enriquez

Introducción

Introducción

Las redes neuronales artificiales emulan el comportamiento de las neuronas humanas.

- **Las neuronas** son las células del sistema nervioso encargadas de transmitir información mediante impulsos nerviosos.
- **Una red neuronal** es la unión entre dos o más neuronas, estas se relacionan entre si y pueden transmitir la información de manera eficaz.



Modelo Biológico de una neurona

Características de una red neuronal

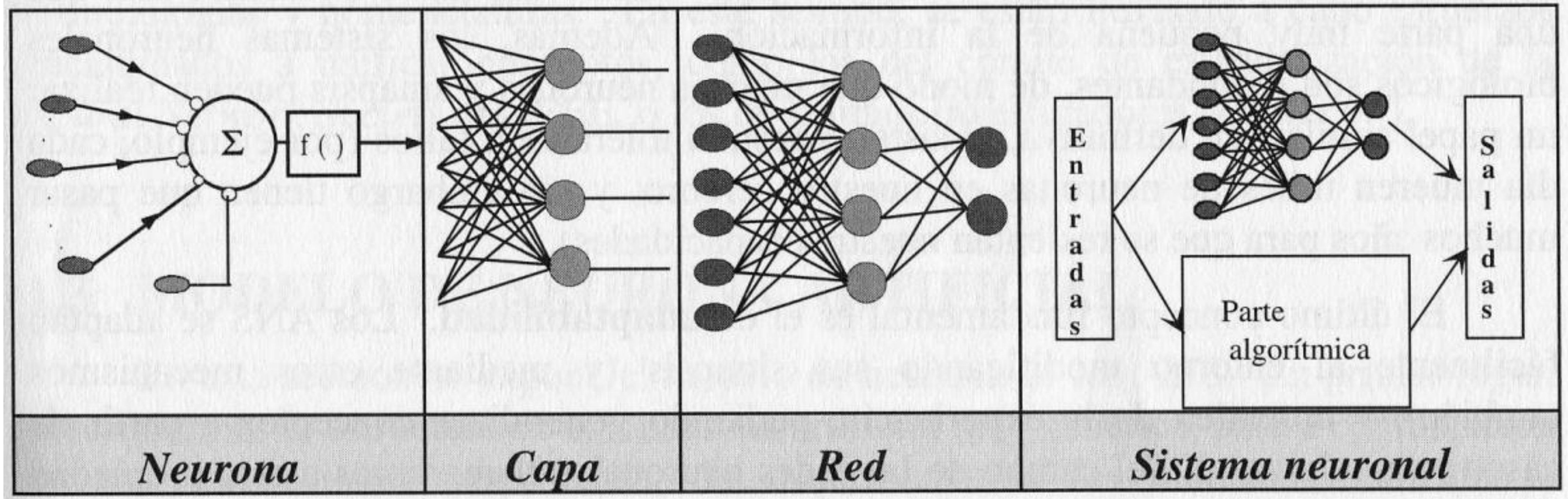
- Método general y práctico para aprendizaje supervisado.
- Aprenden una función $f : X \rightarrow Y$, donde no es necesario conocer apriori su "forma".
- No tienen una interpretabilidad clara, funcionan como cajas negras.
- Requiere muchos recursos computacionales para ser entrenadas.
- La aplicación del modelo es eficiente y requiere pocos recursos computacionales.
- Funcionan con datos de entrada ruidosos y complejos.

Características de una red neuronal

- La estructura de una red neuronal es paralela, por lo cual si esto es implementado sobre un cluster de computadoras, se pueden obtener respuestas en tiempo real.
- Aplicaciones Comunes:
  - Reconocimiento de fonemas en señales de voz.
  - Reconocimiento de caracteres desde escritura manual.
  - Clasificación de imágenes.
  - Predicción financiera.

Estructura de una red neuronal

- En términos generales, una red consiste en un gran número de unidades simples de proceso, denominas neuronas, que actúan en paralelo, están agrupadas en capas y están conectadas mediante vínculos ponderados. Finalmente una red neuronal junto con los interfaces de entrada y salida constituyen el sistema global de proceso.



El modelo estándar de neurona artificial

- Se va a introducir el denominado modelo estándar de neurona artificial según los principios descritos en Rumelhart y McClelland (1986) y McClelland y Rumelhart (1986).

- Siguiendo dichos principios, la i-ésima neurona artificial estándar consiste en:

1. Un ***conjunto de entradas*** x_j y unos pesos sinápticos $w_{i,j}$, con $j = 1, \dots, n$

2. Una ***regla de propagación*** hi definida a partir del conjunto de entradas y los pesos sinápticos. Es decir:

$$h_i(x_1, \dots, x_n, w_{i,1}, \dots, w_{i,n})$$

   - La regla de propagación más comúnmente utilizada consiste en combinar linealmente las entradas y los pesos sinápticos, obteniéndose:

$$h_i\left(x_1, \dots, x_n, w_{i,1}, \dots, w_{i,n}\right) = \sum_{j=1}^{n} w_{i,j}\, x_j$$

El modelo estándar de neurona artificial

- Suele ser habitual añadir al conjunto de pesos de la neurona un parámetro adicional θ_i, que se denomina umbral, el cual se acostumbra a restar al potencial pos-sináptico. Es decir:

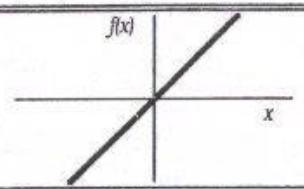
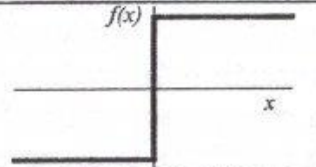
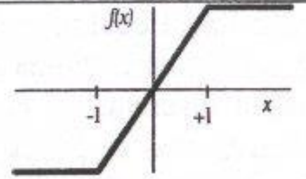
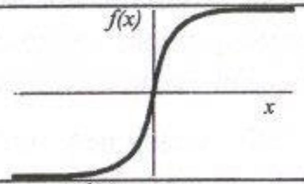
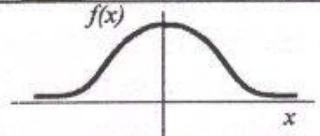
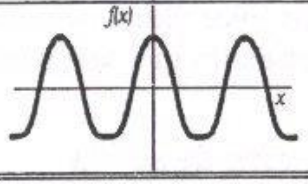
$$h_i(x_1, \dots, x_n, w_{i,1}, \dots, w_{i,n}) = \sum_{j=1}^{n} w_{i,j}\, x_j - \theta_i$$

3. Una **función de activación**, la cual representa simultáneamente la salida de la neurona y su estado de activación. Si denotamos por y_i dicha función de activación, se tiene:

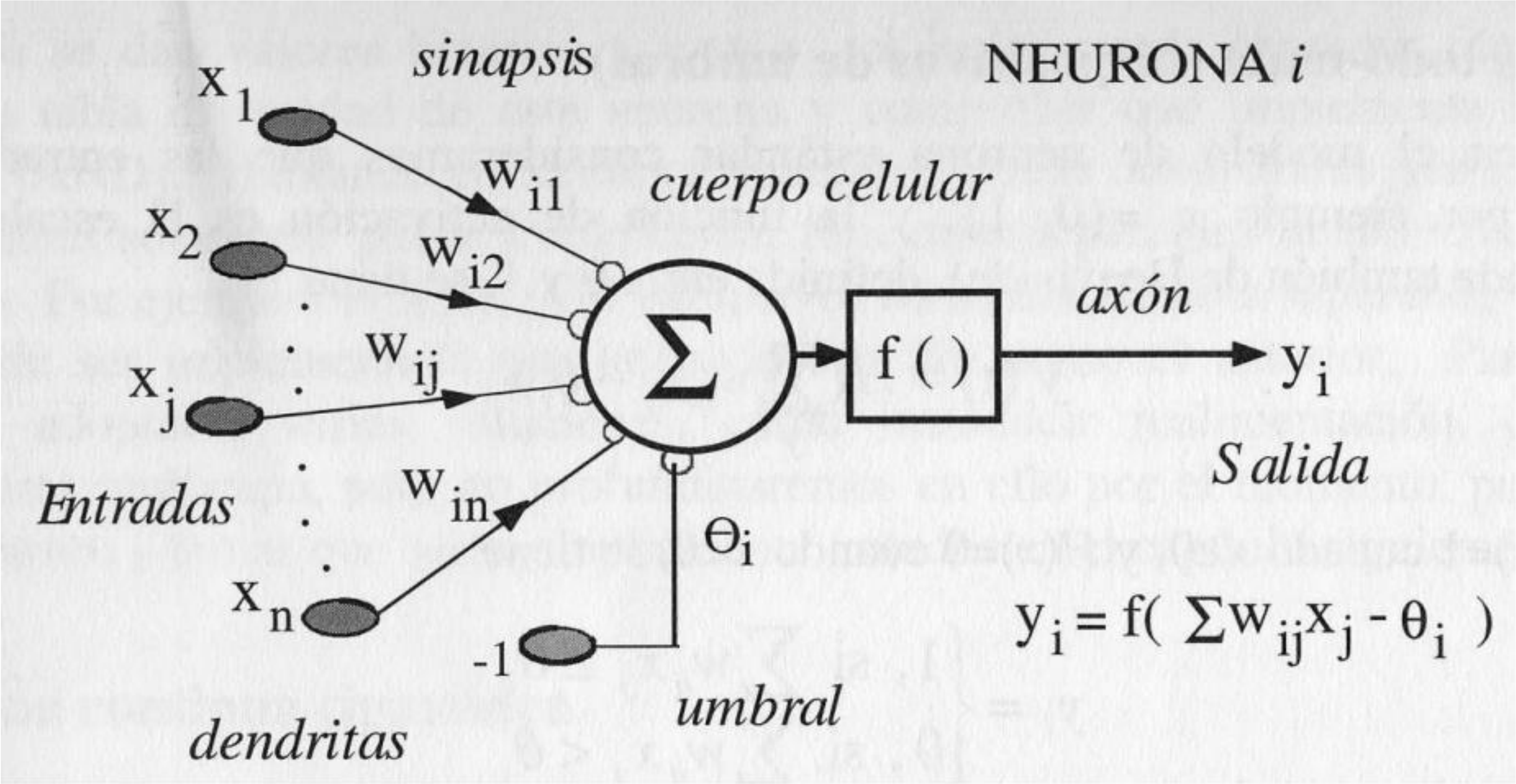
$$y_i = f(h_i) = f\left(\sum_{j=1}^{n} w_{i,j}\, x_j - \theta_i\right)$$

El modelo estándar de neurona artificial

Ejemplo de funciones de activación

	Función	Rango	Gráfica
Identidad	$y = x$	[-∞, +∞]	
Escalón	$y = sign(x)$ $y = H(x)$	{-1, +1} {0, +1}	
Lineal a tramos	$y = \begin{cases} -1, & si\ x < -l \\ x, & si\ +l \le x \le -l \\ +1, & si\ x > +l \end{cases}$	[-1, +1]	
Sigmoidea	$y = \dfrac{1}{1+e^{-x}}$ $y = tgh(x)$	[0, +1] [-1, +1]	
Gaussiana	$y = Ae^{-Bx^2}$	[0,+1]	
Sinusoidal	$y = A\,\mathrm{sen}(\omega x + \varphi)$	[-1,+1]	

El modelo estándar de neurona artificial



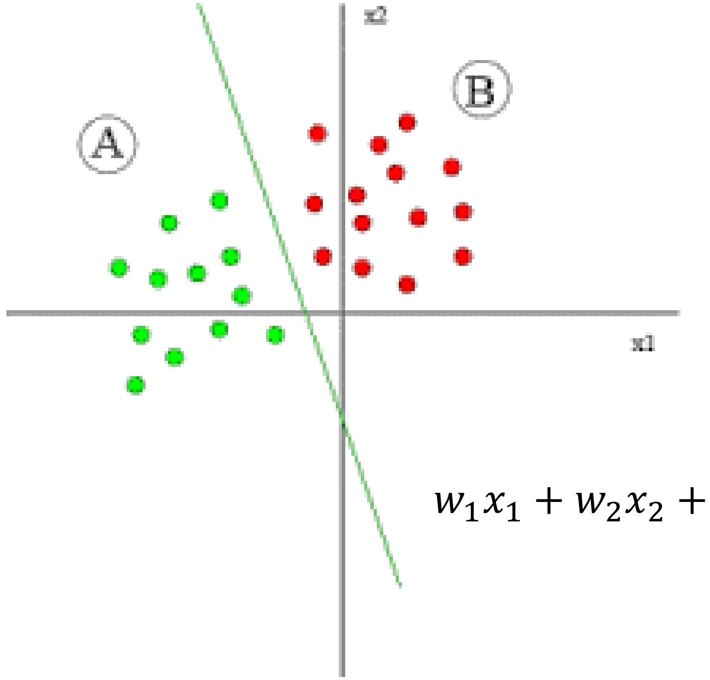
Tipos de Redes Neuronales

PERCEPTRÓN SIMPLE

Perceptron Simple

- Fue introducido por Rosenblatt en 1962.
- Se concibió como un sistema capaz de realizar tareas de clasificación de forma automática.

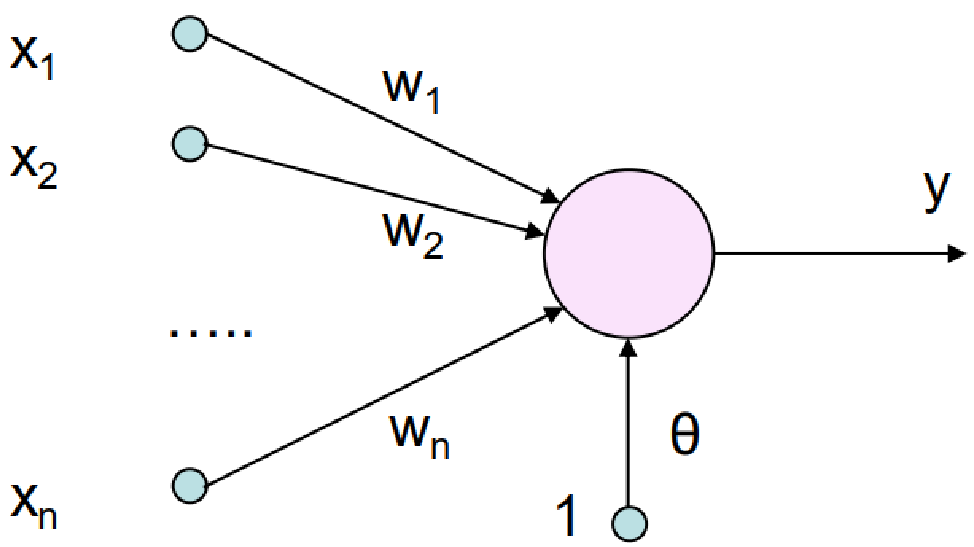
A partir de un número de elementos etiquetados, el sistema determina la ecuación del hiperplano discriminante.



$$w_1x_1 + w_2x_2 + \theta = 0$$

Perceptron Simple: Arquitectura

- Es un modelo unidireccional compuesto por dos capas de neuronas, una de entrada y otra de salida.



$$y = \begin{cases} +1, & si\ \ w_1x_1 + \cdots + w_nx_n + \theta > 0 \\ -1, & si\ \ w_1x_1 + \cdots + w_nx_n + \theta \leq 0 \end{cases}$$

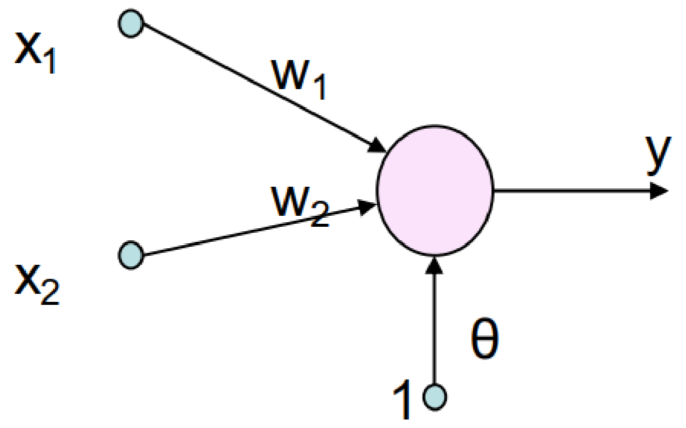
Perceptron Simple: Arquitectura

- El perceptron equivale a un hiperplano de dimensión $n - 1$ capaz de separar las clases
  - Si la salida del perceptron es +1, la entrada pertenecerá a una clase (estará situada a un lado del hiperplano)
  - Si la salida es -1, la entrada pertenecerá a la clase contraria (estará situada al otro lado del hiperplano)

- La ecuación del hiperplano es:

$$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + \theta = 0$$

Perceptron Simple: Ejemplo 2 dimensiones



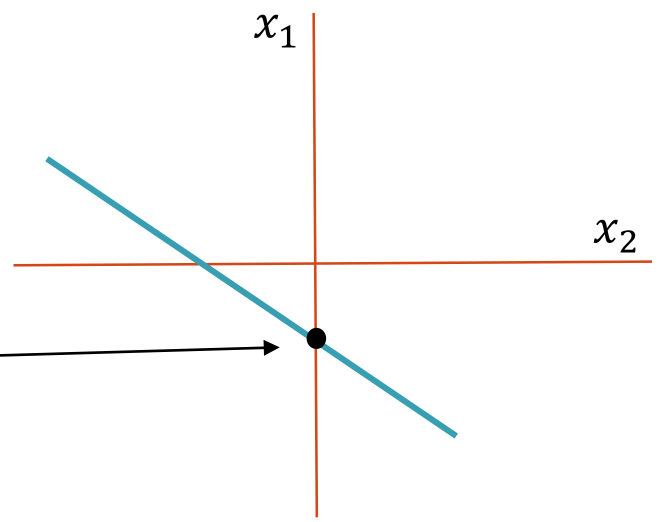
$$y = \begin{cases} +1, & si\ \ w_1x_1 + w_2x_2 + \theta > 0 \\ -1, & si\ \ w_1x_1 + w_2x_2 + \theta \leq 0 \end{cases}$$

La ecuación del hiperplano es: $w_1x_1 + \ w_2x_2 + \theta = 0$

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

Pendiente de la recta

Punto de corte



Perceptron Simple: Aprendizaje

- Se dispone de un conjunto de observaciones (patrones, ejemplos, datos) de los que se sabe su categoría o clase.
- Los ejemplos o datos son puntos en un espacio multidimensional:

$$\Re^n : (x_1, \ldots, x_n)$$

- Hay que determinar la ecuación del hiperplano que deja a un lado los ejemplos de una clase y a otro lado los de la otra clase.
- La ecuación del hiperplano se deduce a partir de los ejemplos o datos.
- Proceso iterativo supervisado.

Perceptron Simple: Aprendizaje

- Dado:
  - Conjunto de patrones
  - Vector de entrada: $(x_1, \dots, x_n)$
  - Salida: $d(x)$

$$d(x) = +1 \quad si\ x \in A$$

$$d(x) = -1 \quad si\ x \in B$$

- Hiperplano discriminante:

$(w_1, \dots, w_n, \theta)$ tales que

$$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + \theta = 0$$

Separe las clases A y B.

Perceptron Simple: Aprendizaje

1. Comenzar con valores aleatorios para pesos y umbral
2. Modificación de los pesos y umbral hasta encontrar el hiperplano discriminante
   a. Seleccionar un ejemplo x del conjunto de entrenamiento
   b. Se calcula la salida de la red: $y = f(w_1 x_1 + \ w_2 x_2, \dots, w_n x_n + \theta)$
   c. Si $y \neq \ d(x)$ (clasificación incorrecta) se modifican los pesos y el umbral:

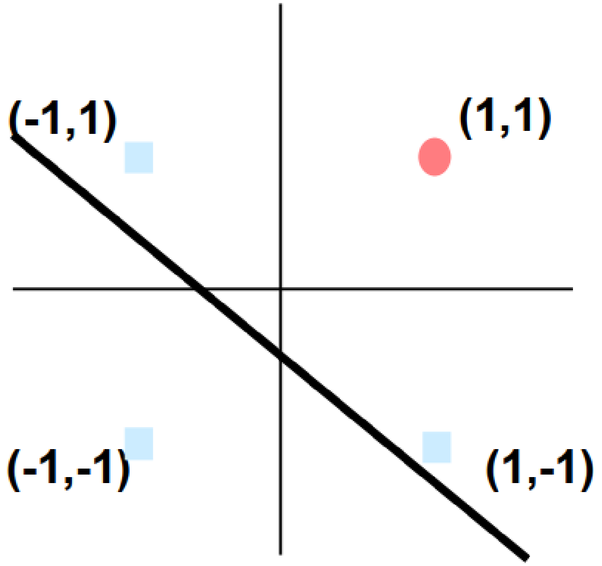
$$w_i(t+1) = \ w_i\,(t) + d(x)\ * x_i$$
$$\theta(t+1) = \theta(t) + d(x)$$

   d. Repetir desde el paso 2.a hasta completar el conjunto de patrones de entrenamiento o hasta alcanzar el criterio de parada.

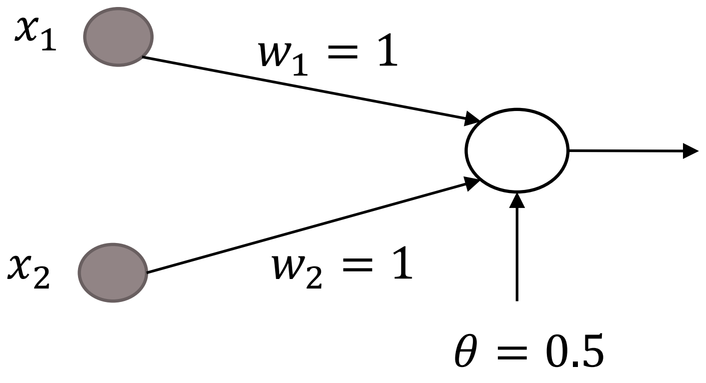
Perceptron Simple: Ejemplo

- Función lógica AND

X_1	X_2	AND
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1



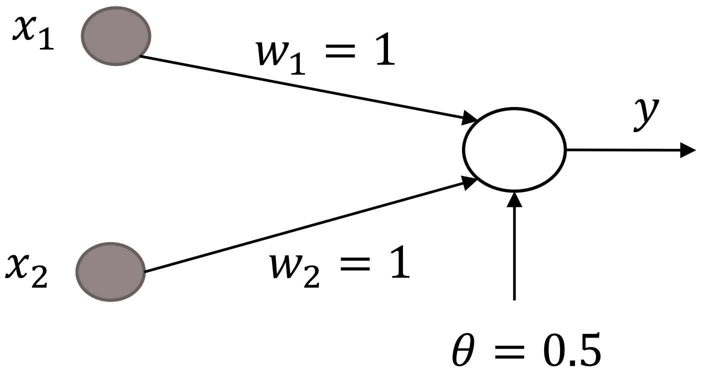
Inicialmente al azar



Perceptron Simple: Ejemplo

- Función lógica AND

x_1	x_2	AND
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1



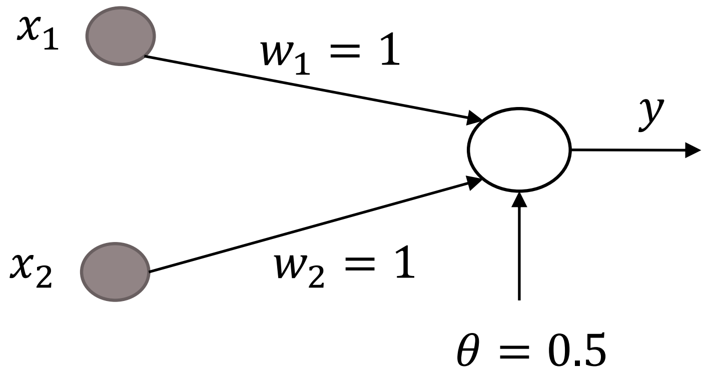
$$y = \begin{cases} +1, & si\ \ w_1x_1 + w_2x_2 + \theta > 0 \\ -1, & si\ \ w_1x_1 + w_2x_2 + \theta \le 0 \end{cases}$$

$\boldsymbol{x = (-1, -1),\ \ d(x) = -1}$ → $\boldsymbol{y = f(-1.5)\ = -1}$ → **Bien clasificado**

Perceptron Simple: Ejemplo

- Función lógica AND

x_1	x_2	AND
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1



$$y = \begin{cases} +1, & si\ \ w_1 x_1 + w_2 x_2 + \theta > 0 \\ -1, & si\ \ w_1 x_1 + w_2 x_2 + \theta \leq 0 \end{cases}$$

$\boldsymbol{x = (1, -1),\ \ d(x) = -1}$ → $\boldsymbol{y = f(0.5) = 1}$ → **Mal clasificado (nuevos pesos)**

$$w_1 = w_1 + d(x) * x_1 = 1 - 1 = 0$$

$$w_2 = w_2 + d(x) * x_2 = 1 + 1 = 2$$

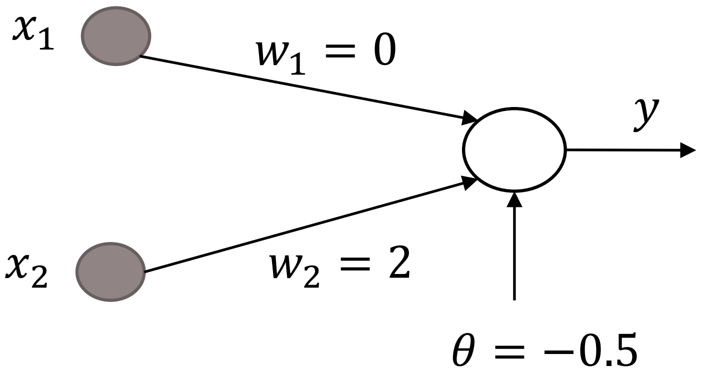
$$\theta = \theta + d(x) = 0.5 - 1 = -0.5$$

→ $\boldsymbol{y = f(-2.5) = -1}$ → **Bien clasificado**

Perceptron Simple: Ejemplo

- Función lógica AND

x_1	x_2	AND
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1



$$y = \begin{cases} +1, & si \;\; w_1x_1 + w_2x_2 + \theta > 0 \\ -1, & si \;\; w_1x_1 + w_2x_2 + \theta \leq 0 \end{cases}$$

$\boldsymbol{x = (-1, 1), \;\; d(x) = -1}$ → $\boldsymbol{y = f(1.5) \; = 1}$ → **Mal clasificado (nuevos pesos)**

$$w_1 = w_1 + d(x) * x_1 = 0 + 1 = 1$$

$$w_2 = w_2 + d(x) * x_2 = 2 - 1 = 1$$

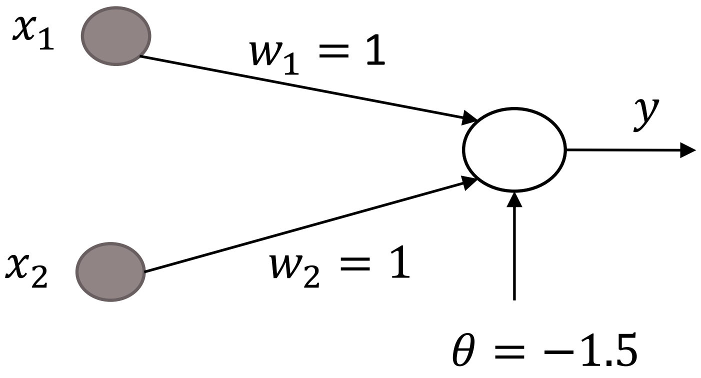
$$\theta = \theta + d(x) = -0.5 - 1 = \; -1.5$$

→ $\boldsymbol{y = f(-1.5) = -1}$ → **Bien clasificado**

Perceptron Simple: Ejemplo

- Función lógica AND

x_1	x_2	AND
-1	-1	-1
1	-1	-1
-1	1	-1
1	1	1



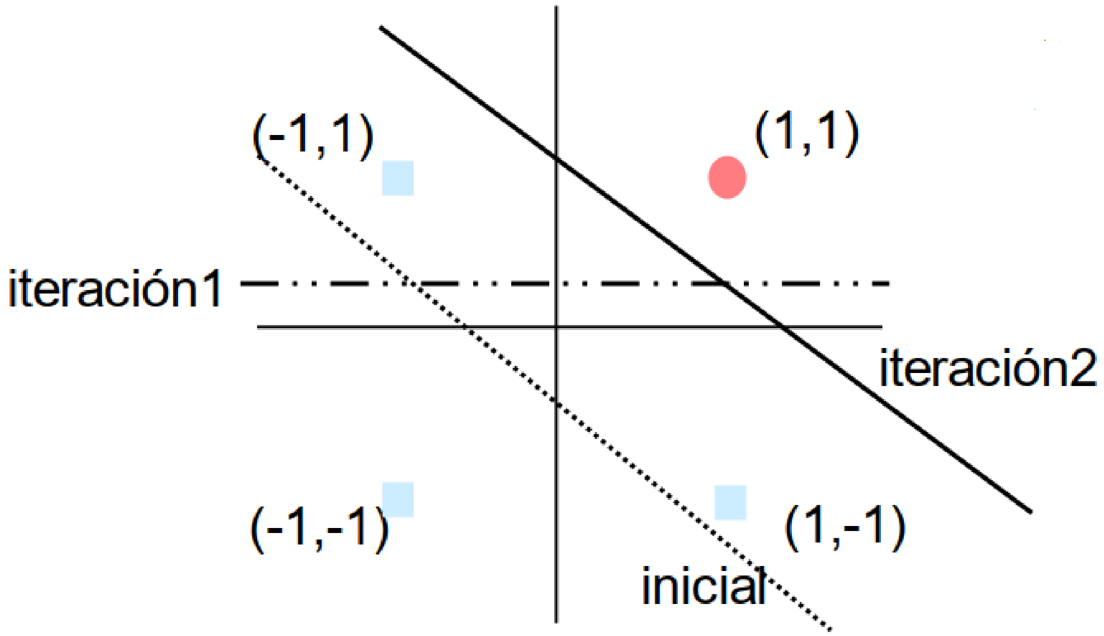
$$y = \begin{cases} +1, & si \;\; w_1x_1 + w_2x_2 + \theta > 0 \\ -1, & si \;\; w_1x_1 + w_2x_2 + \theta \leq 0 \end{cases}$$

$\boldsymbol{x = (1, 1), \;\; d(x) = 1}$ → $\boldsymbol{y = f(2) \; = 1}$ → **Bien clasificado**

Un hiperplano solución es: $\boldsymbol{x_1 + x_2 - 1.5 = 0}$

Perceptron Simple: Ejemplo

- Función lógica AND



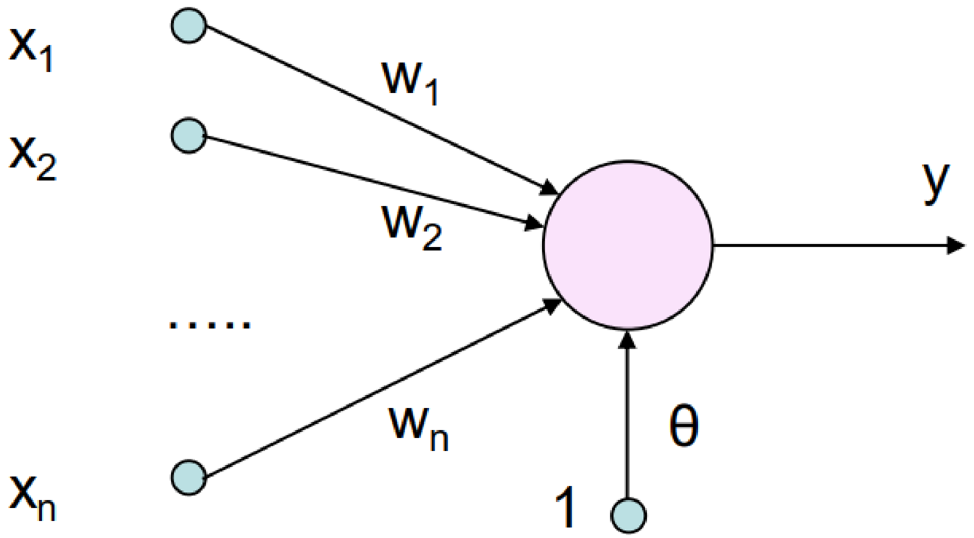
El hiperplano se mueve de una iteración a otra para clasificar correctamente los patrones.

Tipos de Redes Neuronales

REDES ADALINE

ADALINE

- ADALINE (ADAptive LInear NEuron): Desarrollado en 1960 por Widrow y Hoff
- Las entradas pueden ser continuas y se utiliza una neurona similar a la del Perceptron Simple, pero en este caso de respuesta lineal.



$$y = w_1 x_1 + \cdots + w_n x_n + \theta$$

ADALINE

- La diferencia con el perceptron es la manera de utilizar la salida en la regla de aprendizaje.
  - El perceptron utiliza la salida de la función umbral (binaria) para el aprendizaje. Sólo se tiene en cuenta si se ha equivocado o no.
  - En Adaline se utiliza directamente la salida de la red (real) teniendo en cuenta **cuánto se ha equivocado**.
- La regla de aprendizaje de ADALINE considera el error entre la salida lograda $\boldsymbol{y}$ versus la salida deseada $\boldsymbol{d}$: $|d^p - y^p|$
- Esta regla se conoce como REGLA DELTA. La constante α se denomina TASA DE APRENDIZAJE.

$$\Delta w_i = \alpha |d^p - y^p| x_i$$

ADALINE : Aprendizaje

1. Inicializar los pesos y umbral de forma aleatoria
2. Seleccionar un ejemplo x del conjunto de entrenamiento
3. Calcular la salida de la red: $y = f(w_1 x_1 + \cdots + w_n x_n + \theta)$

   y obtener la diferencia $|d^p - y^p|$
4. Para todos los pesos y para el umbral, calcular

$$\Delta w_i = \alpha |d^p - y^p| x_i \qquad \Delta \theta_i = \alpha |d^p - y^p|$$

5. Modificar los pesos y el umbral del siguiente modo

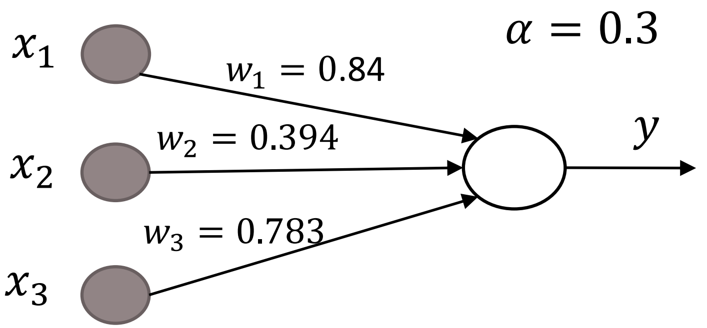
$$w_i(t+1) = w_i(t) + \Delta w_i$$
$$\theta(t+1) = \theta(t) + \Delta \theta_i$$

6. Repetir para todos los patrones de entrenamiento hasta cumplir el criterio de parada.

ADALINE : Ejemplo

- Decodificador binario a decimal

x_1	x_2	x_3	d(X)
0	0	1	**1**
0	1	0	**2**
0	1	1	**3**
1	0	0	**4**
1	0	1	**5**
1	1	0	**6**
1	1	1	**7**



$$\boldsymbol{y = 0.84 * 0 + 0.394 * 0 + 0.783 * 1 = 0.783}$$

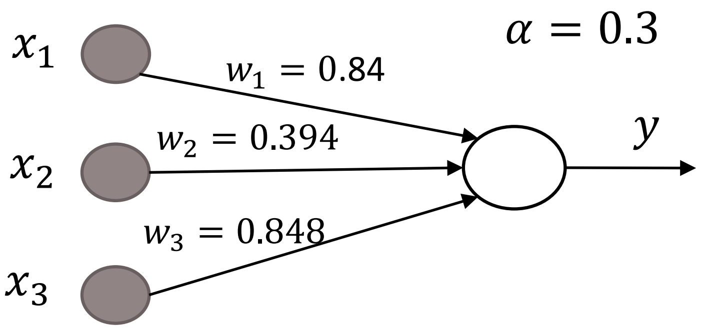
$$E = |d^p - y^p| = |1 - 0.783| = 0.217$$

$$w_1 = w_1 + \alpha * E * x_1 = 0.84 + 0.3 * 0.217 * 0 = \mathbf{0.840}$$
$$w_2 = w_2 + \alpha * E * x_2 = \mathbf{0.394}$$
$$w_3 = w_3 + \alpha * E * x_3 = \mathbf{0.848}$$

ADALINE : Ejemplo

- Decodificador binario a decimal

x_1	x_2	x_3	d(X)
0	0	1	**1**
0	1	0	**2**
0	1	1	**3**
1	0	0	**4**
1	0	1	**5**
1	1	0	**6**
1	1	1	**7**



$$\boldsymbol{y = 0.84 * 0 + 0.394 * 1 + 0.848 * 0 = 0.394}$$

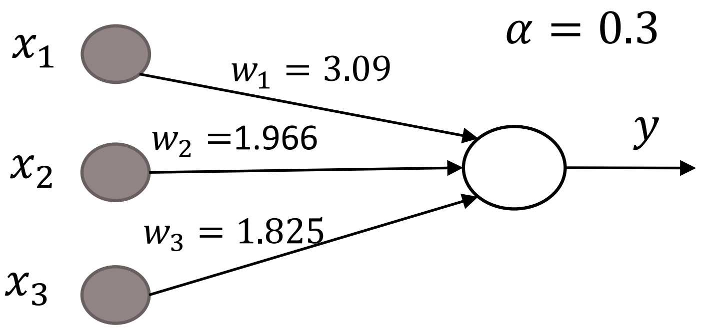
$$E = |d^p - y^p| = |2 - 0.394\,| = 1.61$$

$$w_1 = w_1 + \alpha * E * x_1 = 0.84 + 0.3 * 0.161 * 0 = \mathbf{0.840}$$
$$w_2 = w_2 + \alpha * E * x_2 = \mathbf{0.876}$$
$$w_3 = w_3 + \alpha * E * x_3 = \mathbf{0.848}$$

ADALINE : Ejemplo

- Decodificador binario a decimal

x_1	x_2	x_3	d(X)
0	0	1	**1**
0	1	0	**2**
0	1	1	**3**
1	0	0	**4**
1	0	1	**5**
1	1	0	**6**
1	1	1	**7**



$$\boldsymbol{y = 3.09 * 1 + 1.966 * 1 + 1.825 * 1 = 6.881}$$

$$E = |d^p - y^p| = |7 - 6.881| = 0.12$$

$$w_1 = w_1 + \alpha * E * x_1 = 3.09 + 0.3 * 0.12 * 1 = \mathbf{3.126}$$
$$w_2 = w_2 + \alpha * E * x_2 = \mathbf{2.002}$$
$$w_3 = w_3 + \alpha * E * x_3 = \mathbf{1.861}$$

Resultado después de la primera iteración del entrenamiento

ADALINE : Ejemplo

- Decodificador binario a decimal. Visualización de los pesos según iteraciones.

Iteración	w1	w2	w3
1	3.12	2.00	1.86
2	3.61	1.98	1.42
3	3.82	1.98	1.2
4	3.92	1.98	1.1
5	3.96	1.99	1.02
6	3.99	2.00	1.01
7	4.00	2.00	1.00
8	4.00	2.00	1.00
9	4.00	2.00	1.00
10	4.00	2.00	1.00

> La tasa de aprendizaje α también puede ser adaptativa.

> Por ejemplo al inicio el valor puede ser alto, para dar "grandes pasos" de corrección del error y para salir de mínimos locales.

> Sin embargo al final del entrenamiento debe disminuir para hacer correcciones finas.

Conclusión

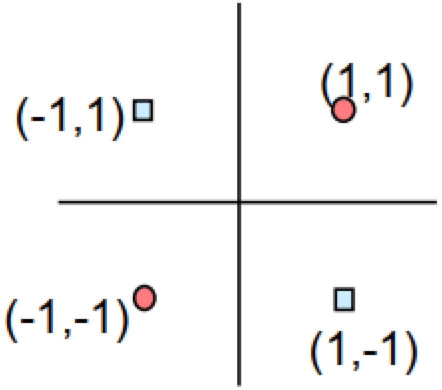
- El uso del Perceptrón o de las redes ADALINE permite aproximar de manera fácil, cualquier tipo de función o sistemas, sólo conociendo un conjunto de ejemplos.
- De esta manera cualquier sistema (caja negra), se puede representar por una red.
- Sin embargo, también se demostró que estas técnicas poseen grandes limitaciones.
  - Un ejemplo clásico es el OR Exclusivo
- En conclusión: éstas técnicas sólo pueden resolver sistemas donde los ejemplos son linealmente separables.

Conclusión

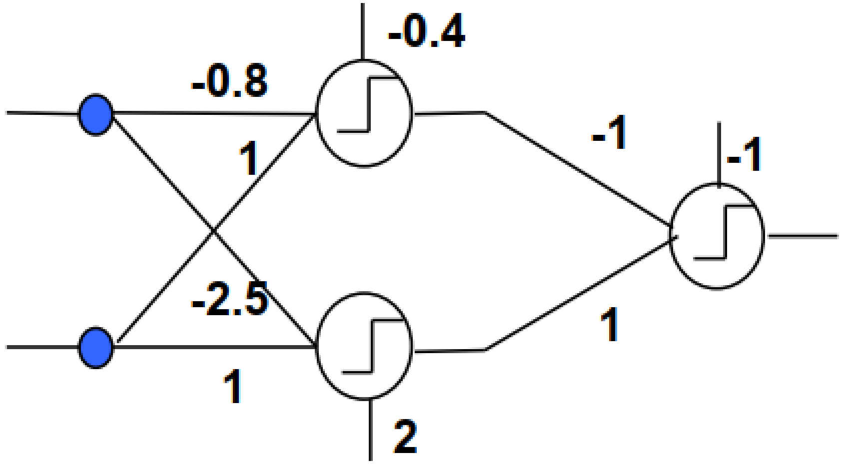
Problemas no linealmente separables

Ejemplo XOR: No existe un hiperplano.

x_1	x_2	d(x)
-1	-1	1
-1	1	-1
1	-1	-1
1	1	1



Solución: Combinar varios Perceptrones



Multicapas