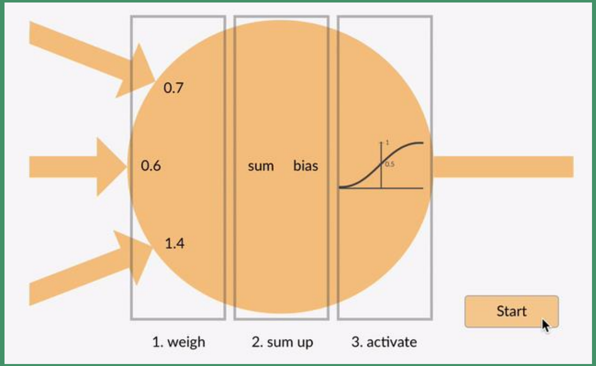


Inteligencia Artificial

Unidad 2: Redes Neuronales

TEMA 3: Algoritmos de IA Moderna-I

Módulo 2: Desde el Perceptrón a las Redes Neuronales

Unidad 2

Redes Neuronales

TEMA 3: Algoritmos de IA Moderna-I

Sesión 11-12

MÓDULO 2:

Desde el Perceptrón a las Redes Neuronales

Contenido

1. Perceptrón vs. Red Neuronal Artificial
2. Estructura de una Red Neuronal Artificial
3. Modelo Estándar de una Red Neuronal Artificial
4. El Perceptrón Simple
5. Redes ADALINE
6. Ventajas / Limitaciones

Preguntas

Aprendizaje supervisado

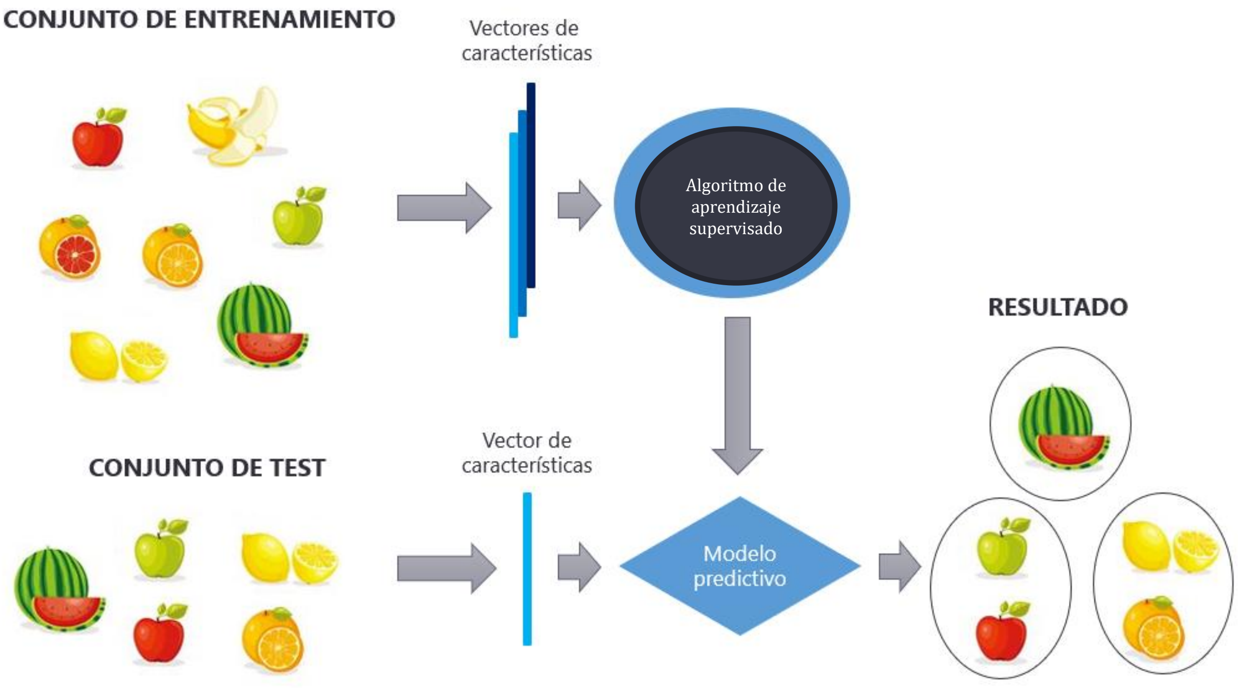
- El aprendizaje supervisado es una técnica aplicado en aprendizaje automático y minería de datos.
- El objetivo del aprendizaje supervisado es clasificar o predecir el valor correspondiente a cualquier objeto de entrada válida después de haber visto una serie de ejemplos denominado datos de entrenamiento. Para ello, tiene que generalizar a partir de los datos presentados a las situaciones no vistas previamente denominado test o prueba.

Clasificación vs Predicción

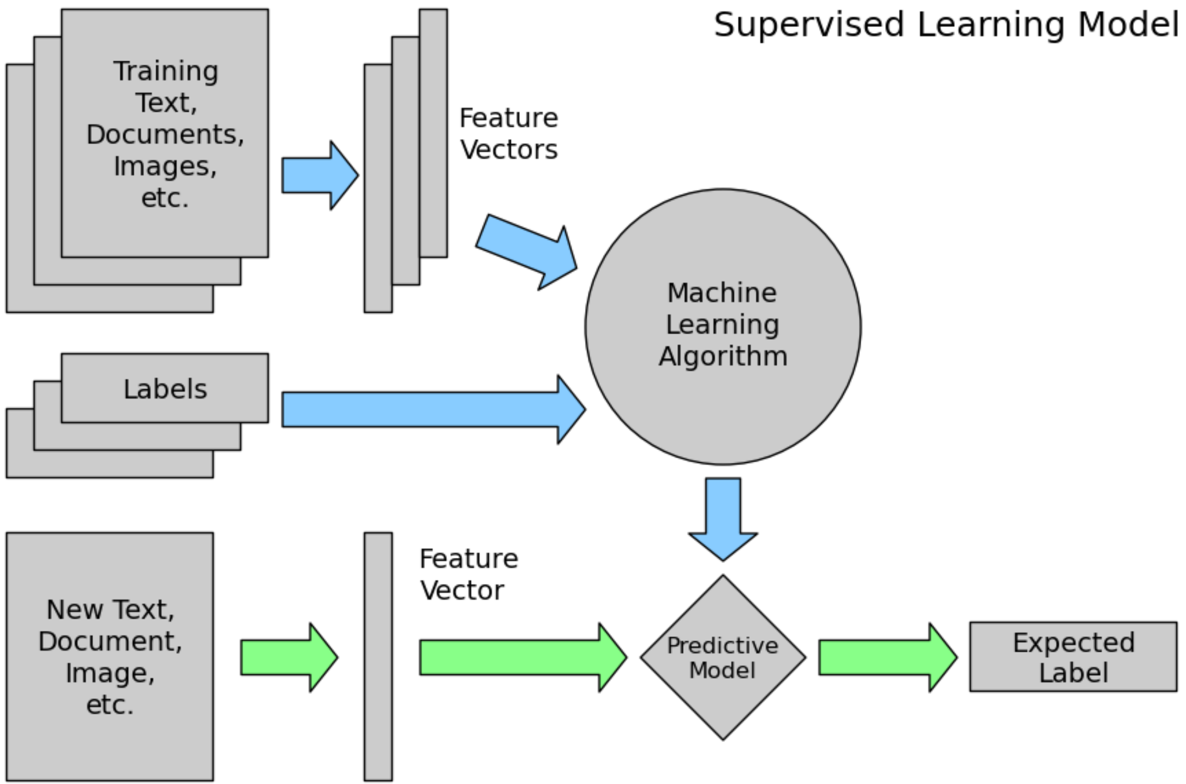
- Clasificación
  - Usado para predecir el valor de un atributo categórico (discreto o nominal)

- Predicción
  - Usado para modelar funciones que toman valore continuos
  - Predecir valores numéricos desconocidos

Aprendizaje Supervisado



Modelo supervisado

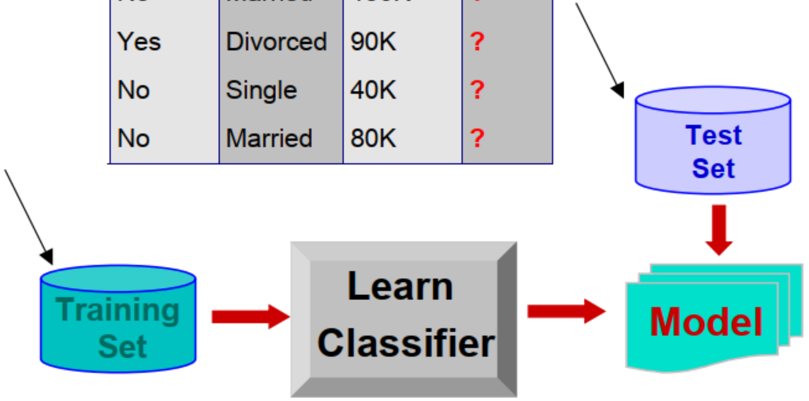


Tarea de clasificación

Ejemplo: predicción de engaños al fisco

Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	**Marital Status**	**Taxable Income**	**Cheat**
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



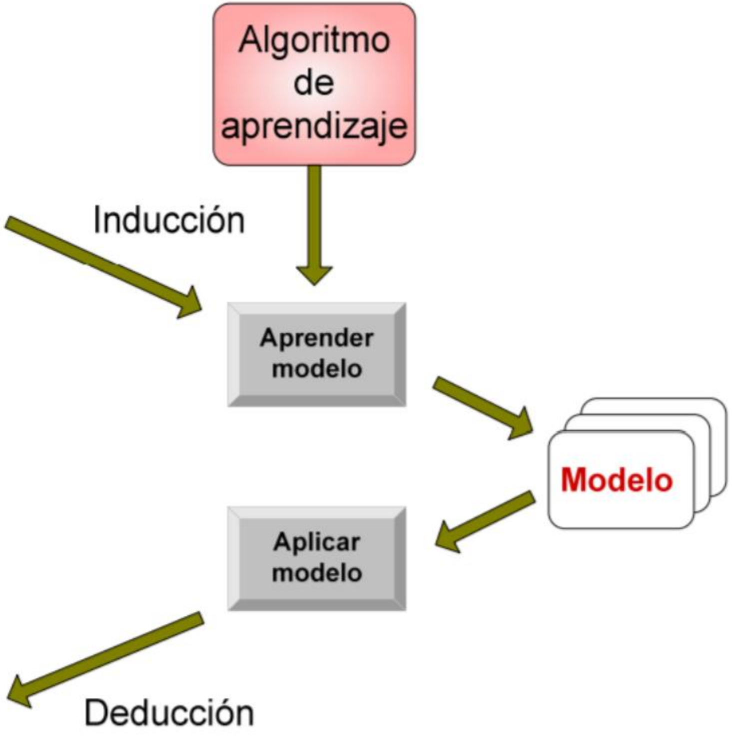
Modelo de clasificación o modelo de predicción

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

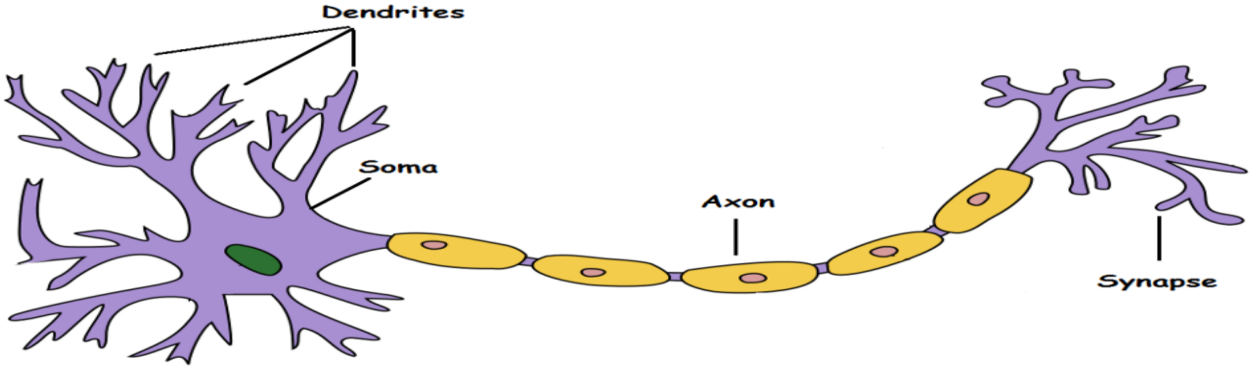
Conjunto de entrenamiento

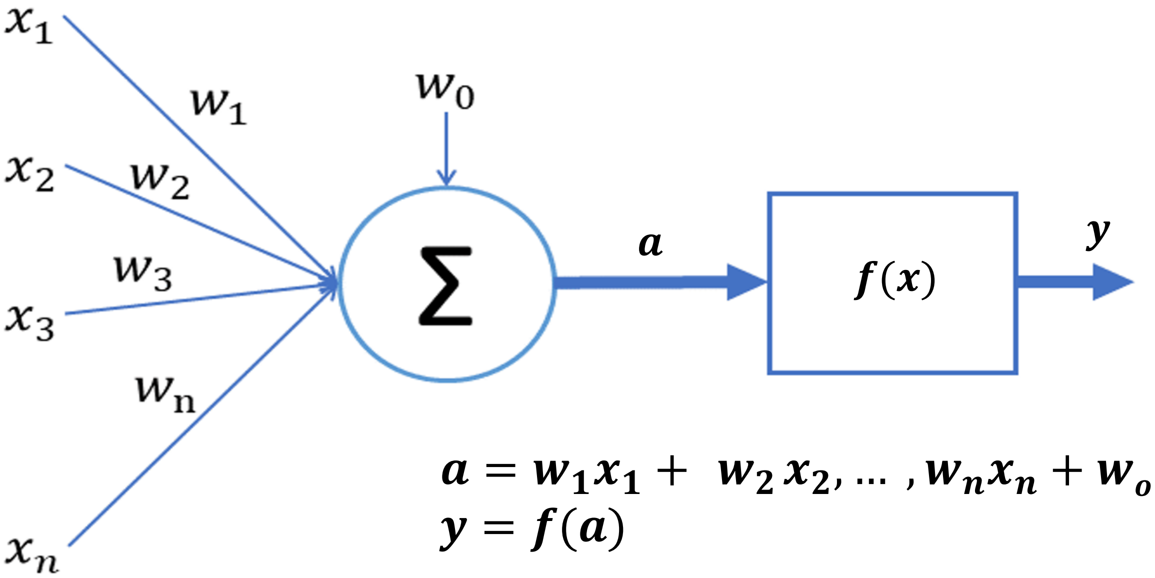
Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Conjunto de prueba



Arquitectura de la neurona artificial



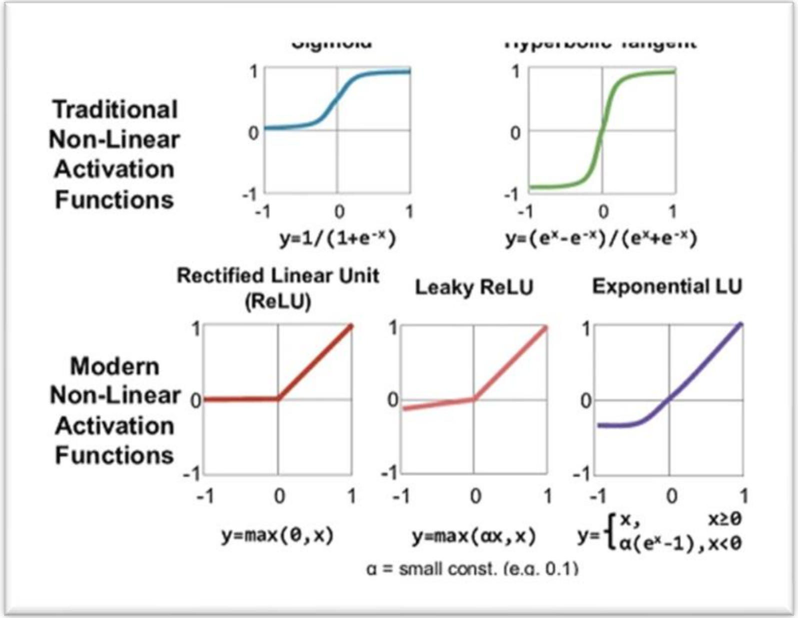


$$a = w_1x_1 + \ w_2\,x_2, \ldots\ , w_nx_n + w_o$$
$$y = f(a)$$

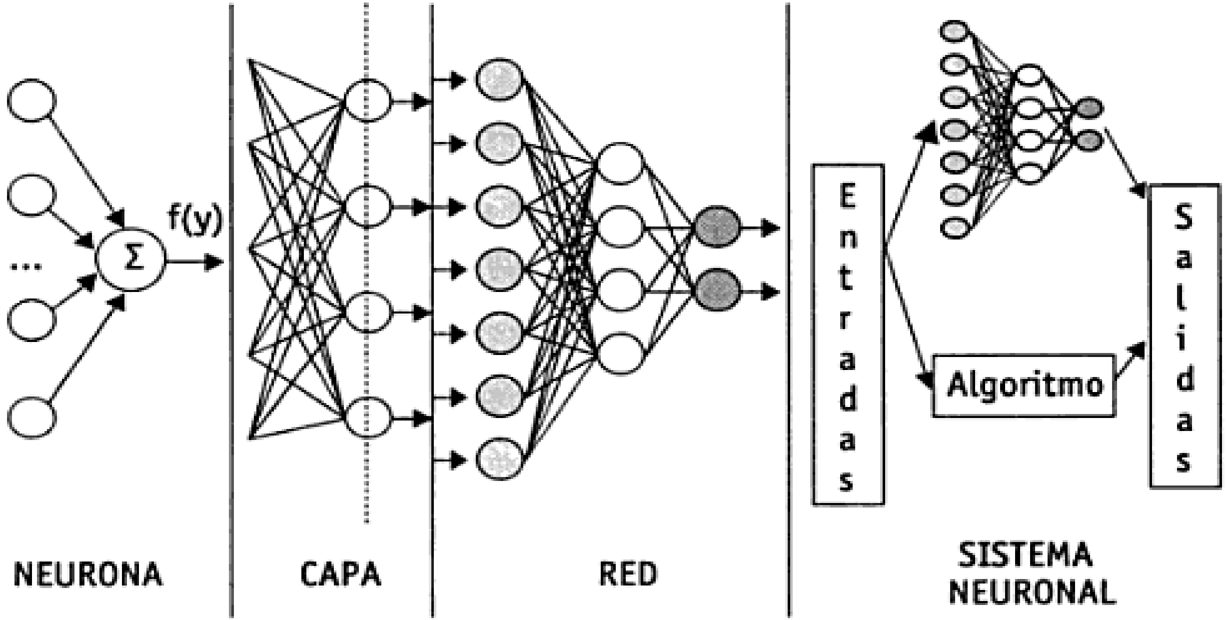
Funciones de activación para la neurona

Nombre	Relación Entrada /Salida	Icono
Limitador Fuerte	$a = 0 \quad n < 0$ $a = 1 \; n \geq 0$	
Limitador Fuerte Simétrico	$a = -1 \quad n < 0$ $a = +1 \quad n \geq 0$	
Lineal Positiva	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n$	
Lineal	$a = n$	
Lineal Saturado	$a = 0 \quad n < 0$ $a = n \quad 0 \leq n < 1$ $a = 1 \; n > 1$	

Nombre	Relación Entrada /Salida	Icono
Lineal Saturado Simétrico	$a = -1 \quad n < -1$ $a = n \quad -1 \leq n \leq 1$ $a = +1 \; n > 1$	
Sigmoidal Logarítmico	$a = \dfrac{1}{1 + e^{-n}}$	
Tangente Sigmoidal Hiperbólica	$a = \dfrac{e^n - e^{-n}}{e^n + e^{-n}}$	
Competitiva	a $= 1$ *Neurona con n max* a $= 0$ *el resto de neuronas*	C



Funciones de activación para la neurona



Algoritmo de Perceptron unicapa

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

1. Iniciar valores aleatorios para pesos y umbral
2. Modificación de los pesos hasta encontrar el hiperplano discriminante
   a) Seleccionar un ejemplo x del conjunto de entrenamiento
   b) Se calcula la salida de la red:

$$\boldsymbol{a = w_1 x_1 + \ w_2 x_2, \dots, w_n x_n + w_o}$$

$$\boldsymbol{y = f(a)}$$

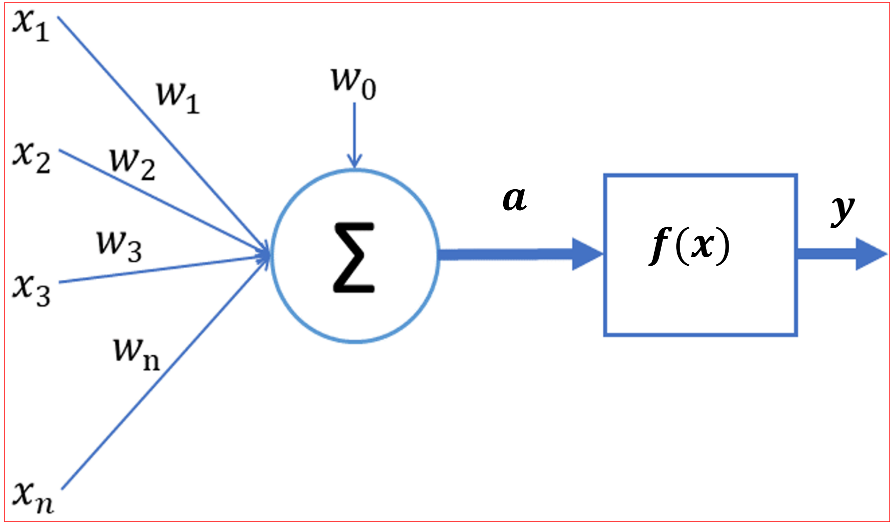
   c) Si $y \neq \ d(x)$ se modifican los pesos:

$$\boldsymbol{w_i(t+1) = w_i\,(t) + \alpha(d(x) - y) * x_i}$$

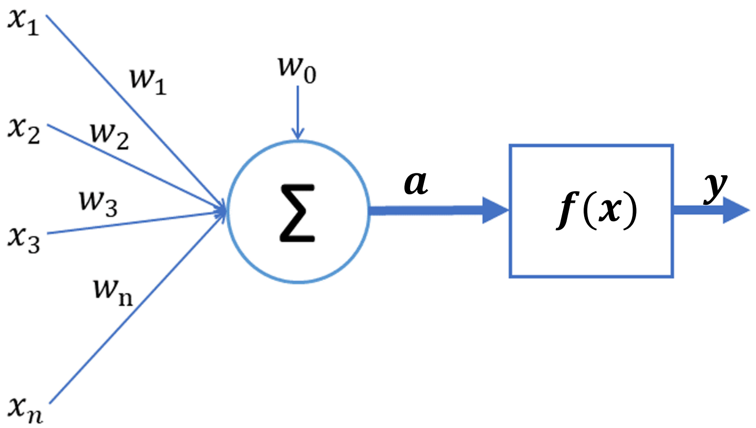
   d) Repetir épocas desde el paso 2.a hasta cumplir un criterio de parada.

Donde:

- $\alpha \quad es\ factor\ de\ aprendizaje$
- $d(x) \ \ clase\ o\ etiqueta\ esperada$



Objetivo de Perceptrón Unicapa



Usa una función de activación ejemplo limitador fuerte simétrico

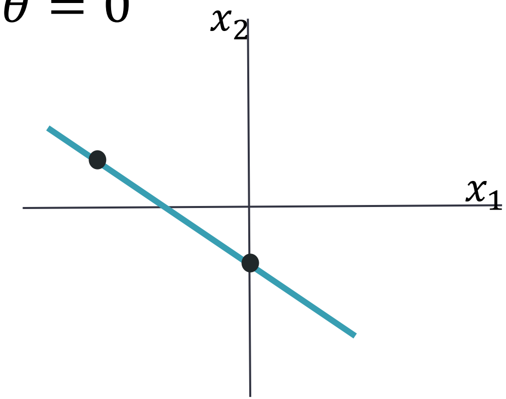
$$\boxed{y = \begin{cases} +1, & si\ a > 0 \\ -1, & si\ a \leq 0 \end{cases}}$$

La ecuación del hiperplano es: $w_1 x_1 + \ w_2 x_2 + \theta = 0$

Pendiente de la recta

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

Punto de corte



Perceptrón Simple: Ejemplo

Considerar el conjunto de datos

Factor de aprendizaje $\alpha = 0.3$

Atributos

	X_1	X_2	Clase
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	1
..	…	….	…
n	….	….	….

1. Diseñar la Red Neuronal que corresponde
2. Presentar el hiperplano discriminante de clasificación previo a entrenamiento
3. Realizar el entrenamiento
4. Presentar el hiperplano discriminante de clasificación posterior al entrenamiento

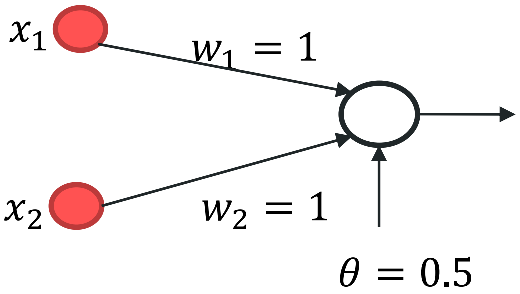
Perceptrón Simple: Ejemplo- Solución

1) Diseñar la Red Neuronal

Conjunto de datos

Atributos

	X_1	X_2	Clase
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	1
..	…	….	…
n	….	….	….



Perceptrón Simple: Ejemplo- Solución

Hallamos un punto

Asumimos $x_1 = -1$

$$(1)(-1) + (1)x_2 + 0.5 = 0$$

x_2 = 1 – 0.5

x_2 = 0.5

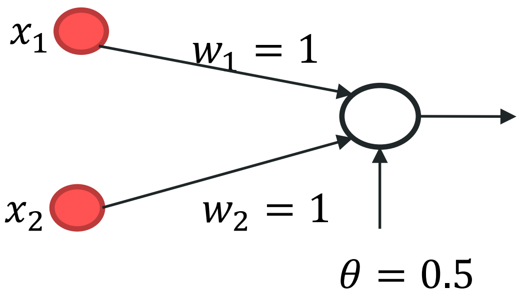
2) Presentar el hiperplano discriminante de clasificación previo a entrenamiento

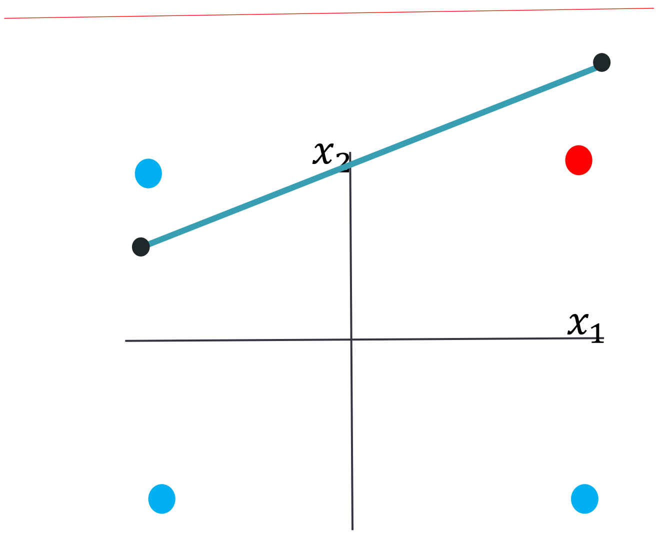
Conjunto de datos

	Atributos		
	X_1	X_2	Clase
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	1
..	…	….	…
n	….	….	….

Considerar la ecuación:

$$w_1 x_1 + w_2 x_2 + \theta = 0$$





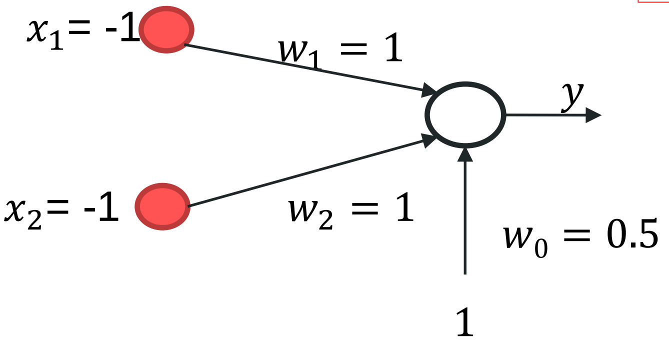
Solución Perceptrón Simple Solución

3) Realizar el entrenamiento con el primer patrón

Conjunto de datos

	X_1	X_2	Clase
a) 1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	1
..	…	….	…
n	….	….	….

$$\boldsymbol{f(x)} \Rightarrow \quad y = \begin{cases} +1, & si\ \ w_1x_1 + w_2x_2 + \theta > 0 \\ -1, & si\ \ w_1x_1 + w_2x_2 + \theta \leq 0 \end{cases}$$



$$\boldsymbol{x = (-1, -1),}$$
$$\boldsymbol{d(x) = -1}$$

b)

$$a = w_1x_1 + w_2x_2 + w_0x_0$$
$$a = (1)(-1) + (1)(-1) + (0.5)(1)$$
$$a = -1.5$$

$$\boldsymbol{y = d(x) = ?}$$
$$\boldsymbol{y = f(-1.5) = -1}$$

c) **Bien clasificado**
No modificamos pesos

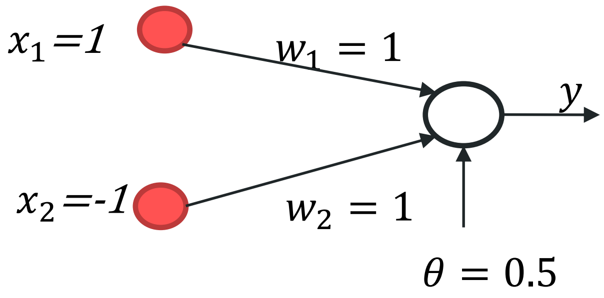
Entrenando Perceptrón Simple - Solución

4) Realizar el entrenamiento con el segundo patrón

Conjunto de datos

	X_1	X_2	Clase
1	-1	-1	-1
a) 2	1	-1	-1
3	-1	1	-1
4	1	1	1
..	…	….	…
n	….	….	….

$$\boldsymbol{f(x)} \quad \Rightarrow \quad y = \begin{cases} +1, & si \;\; w_1x_1 + w_2x_2 + \theta > 0 \\ -1, & si \;\; w_1x_1 + w_2x_2 + \theta \leq 0 \end{cases}$$



Factor de aprendizaje $\alpha = 0.3$

b)
$$\begin{cases} a = w_1x_1 + w_2x_2 + w_0x_0 \\ a = (1)(1) + (1)(-1) + (0.5)(1) \\ a = 0.5 \\ \boldsymbol{y = f(0.5) \;\; = 1} \\ \boldsymbol{y = d(x) = \text{F}} \end{cases}$$

Mal clasificado (nuevos pesos)

c)
$$\begin{cases} w_i(t+1) = \; w_i(t) + \alpha(d(x) - y) \; * x_i \\ w_1 \; = 1 + 0.3(-1 - 1)1 = \; 0.4 \\ w_2 \; = 1 + \; 0.3(-1 - 1) - 1 = 1.6 \\ \theta \; = 0.5 + 0.3(-1 - 1)1 = \; -0.1 \end{cases}$$

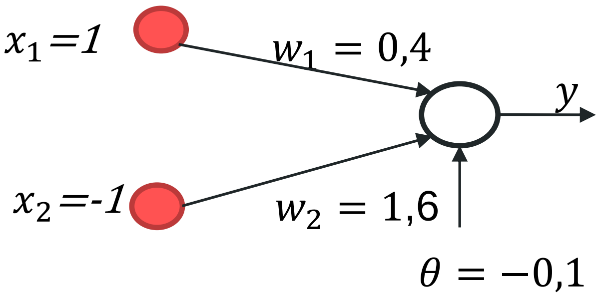
Entrenando Perceptrón Simple - Solución

4) Realizar el entrenamiento con el segundo patrón

Conjunto de datos

		X_1	X_2	Clase
	1	-1	-1	-1
a)	2	1	-1	-1
	3	-1	1	-1
	4	1	1	1
	..	…	….	…
	n	….	….	….

$$f(\boldsymbol{x}) \quad \Rightarrow \quad y = \begin{cases} +1, & si \;\; w_1x_1 + w_2x_2 + \theta > 0 \\ -1, & si \;\; w_1x_1 + w_2x_2 + \theta \leq 0 \end{cases}$$



Factor de aprendizaje $\alpha = 0.3$

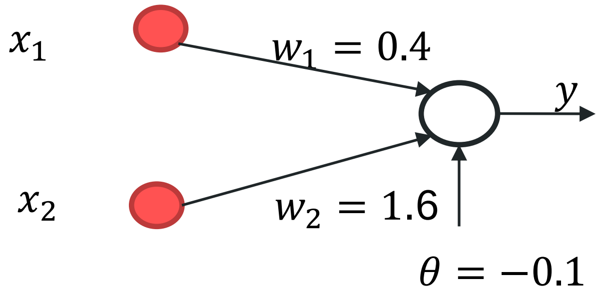
b)

$$a = w_1x_1 + w_2x_2 + w_0x_0$$
$$a = (0,4)(1) + (1,6)(\text{-}1) + (\text{-}0,1)(1)$$
$$a = 0,4\text{-}1,6\text{-}0,1$$
$$\boldsymbol{y = f(-1,3) = -1}$$
$$\boldsymbol{y = d(x) = \mathrm{V}}$$

c) **Bien clasificado**
No modificamos pesos

Resultado de entrenamiento

Después de entrenar 2 patrones



Considerar la ecuación:

$$w_1 x_1 + \; w_2 x_2 + \theta = 0$$

Hallamos un punto

Asumimos $\quad x_1 = -1$

$$(0.4)(-1) + (1.6)x_2 - 0.1 = 0$$

x_2 = (0.4 +0.1)/1.6

x_2 = 0.313

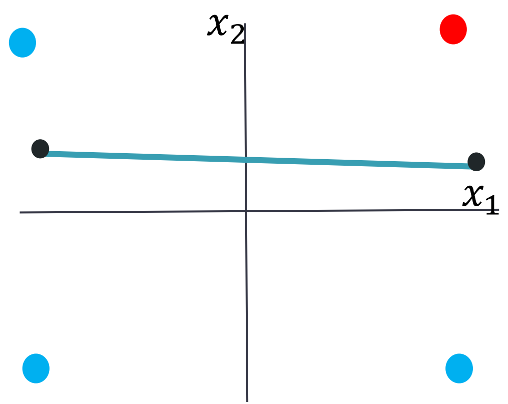
Hallamos el otro punto

Asumimos $\quad x_1 = 1$

$$(0.4)(1) + (1.6)x_2 - 0.1 = 0$$

x_2 = (-0.4 +0.1)/1.6

x_2 = 0.186



Entrenando Perceptrón Simple - Solución

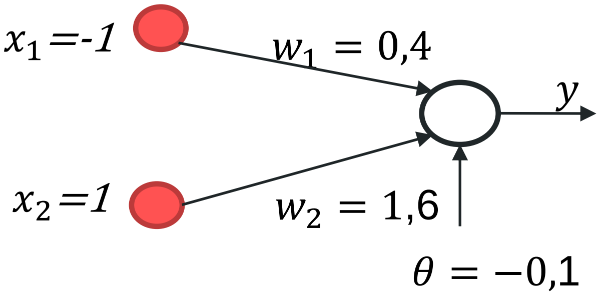
4) Realizar el entrenamiento con el tercer patrón

Conjunto de datos

	X_1	X_2	Clase
1	-1	-1	-1
2	1	-1	-1
a) 3	-1	1	-1
4	1	1	1
..	…	….	…
n	….	….	….

$f(x) \Rightarrow$ $f(x) \Rightarrow$

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



Factor de aprendizaje $\alpha = 0.3$

b)

$$a = w_1x_1 + w_2x_2 + w_0x_0$$
$$a = (0,4)(-1) + (1,6)(1) + (-0,1)(1)$$
$$a = 1,1$$
$$\boldsymbol{y = f(0.5) = 1}$$
$$\boldsymbol{y = d(x) =} \textbf{F}$$

Mal clasificado (nuevos pesos)

c)

$$w_i(t+1) = w_i(t) + \alpha(d(x) - y) * x_i$$
$$w_1 = 0,4 + 0.3(-1-1)(-1 =)\ \ 1$$
$$w_2 = 1,6 + 0.3(-1-1)(+1) = 1$$
$$\theta = -0,1 + 0.3(-1-1)1 = -0.7$$

Entrenando Perceptrón Simple - Solución

4) Realizar el entrenamiento con el tercer patrón

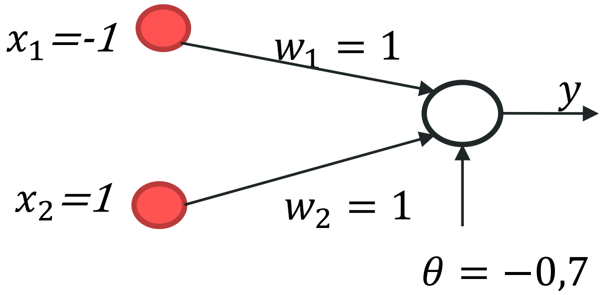
Conjunto de datos

	X_1	X_2	Clase
1	-1	-1	-1
2	1	-1	-1
a) ➡ 3	-1	1	-1
4	1	1	1
..	…	….	…
n	….	….	….

$f(\boldsymbol{x}) \Rightarrow$

$f(\boldsymbol{x}) \Rightarrow$

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



Factor de aprendizaje $\alpha = 0.3$

b)

$$\begin{cases} a = w_1x_1 + w_2x_2 + w_0x_0 \\ a = (1)(-1) + (1)(1) + (-0{,}7)(1) \\ a = -0{,}7 \\ \boldsymbol{y = f(-0,7) = -1} \\ \boldsymbol{y = d(x) = } \mathbf{V} \end{cases}$$

c) **Bien clasificado**
No modificamos pesos

Entrenando Perceptrón Simple - Solución

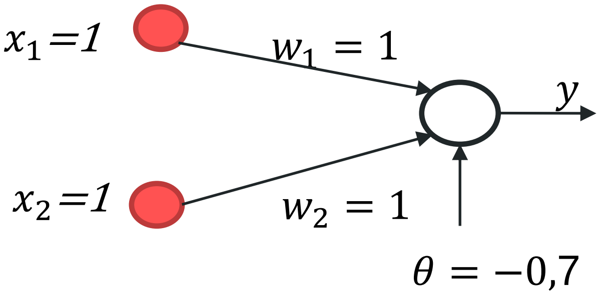
4) Realizar el entrenamiento con el cuarto patrón

Conjunto de datos

	X_1	X_2	Clase
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
a) 4	1	1	1
..	…	….	…
n	….	….	….

$f(x) \Rightarrow$
$f(x) \Rightarrow$

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



Factor de aprendizaje $\alpha = 0.3$

b)

$$a = w_1x_1 + w_2x_2 + w_0x_0$$
$$a = (1)(1) + (1)(1) + (\text{-}0,7)(1)$$
$$a = 1,3$$
$$\boldsymbol{y = f(1.3) = 1}$$
$$\boldsymbol{y = d(x) = \mathrm{V}}$$

c) **Bien clasificado**
No modificamos pesos

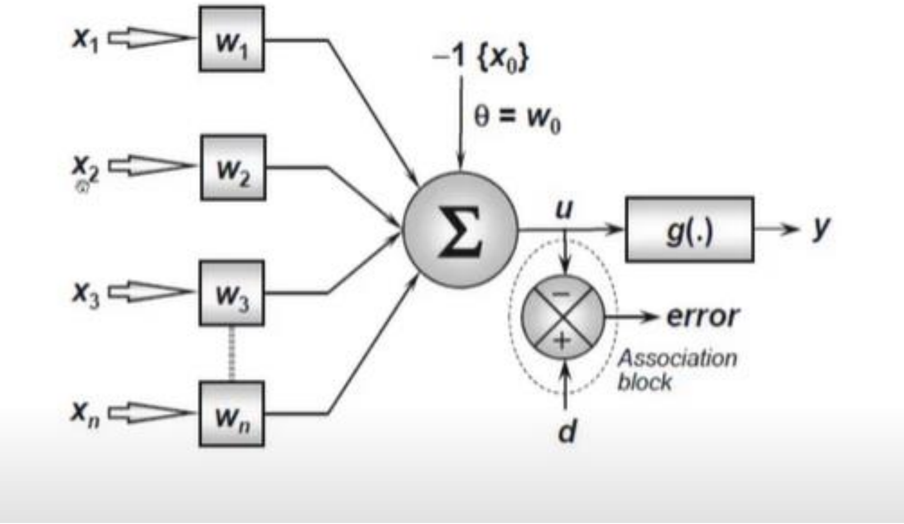
5. Redes ADALINE

ADALINE (**ADA**ptive **LI**near **NE**uron) fue desarrollado en 1960 por Widrow y Hoff.

Dr. Bernard Widrow

PhD. Marcian Hoff

- Las entradas pueden ser continuas y se utiliza una neurona similar a la del Perceptrón Simple, igualmente, resulta un caso de respuesta lineal.



$$u = \sum_{i=1}^{n} w_i * x_i - \theta \longleftrightarrow u = \sum_{i=0}^{n} w_i * x_i$$

5. Redes ADALINE

ADALINE (ADAptive **LI**near **NE**uron) fue desarrollado en 1960 por Widrow y Hoff.

Dr. Bernard Widrow

PhD. Marcian Hoff

- La salida de la Red Adaline se puede dar mediante la ecuación:

$$u = \sum_{i=1}^{n} w_i * x_i - \theta \longleftrightarrow u = \sum_{i=0}^{n} w_i * x_i$$

$$y = g(u)$$

- x_i son las señales de entrada a la red ADALINE.
- w_i son los pesos sinápticos asociados a cada entrada.
- θ es el umbral de activación o bias.
- $g()$ es la función de activación.
- u es el potencial de activación.

5. Redes ADALINE

APRENDIZAJE: DIFERENCIAS ENTRE PERCEPTRON Y REDES ADALINE

PERCEPTRON

- Utiliza la salida de la función umbral (binaria) para el aprendizaje.
- Sólo se tiene en cuenta si se ha equivocado o no (no cuanto se equivocó).
- Valores de respuesta no continuos.

ADALINE

- Utiliza directamente la salida de la red (real) teniendo en cuenta **cuánto se ha equivocado**.
- Considera el error entre la salida lograda $\boldsymbol{y}$ versus la salida deseada $\boldsymbol{d}$ (etiqueta): $$|d^p - y^p|$$
- Esta regla se conoce como **REGLA DELTA**. La constante α se denomina **TASA DE APRENDIZAJE**. $$\Delta w_i = \alpha |d^p - y^p| x_i$$
- Valores de respuesta continuos.
- Se busca minimizar la desviación de la red para todos los patrones (características) de entrada, eligiendo una medida del error global.
- Normalmente se utiliza el error cuadrático medio:

$$E = \frac{1}{2} \sum_{p=1}^{m} (d^p - y^p)^2$$

Error cuadrático medio, compara un valor predicho y un valor observado o conocido.

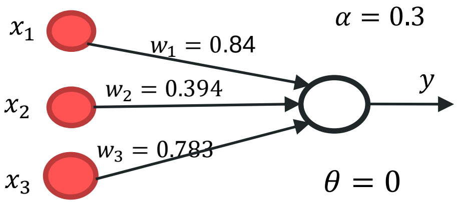
5. Redes ADALINE

EJEMPLO: Decodificador binario a decimal

Datos

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**

Paso #1: Asignamos valores aleatorios para pesos de las entradas y umbral



Paso #2: Seleccionar un ejemplo x del conjunto de entrenamiento: x_1 =0, x_2 =0, x_3 =1

Paso #3: Calcular la salida de la red: $y = f(w_1 x_1 + \cdots + w_n x_n + \theta)$ y obtener la diferencia.

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

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

Paso #4: Para todos los pesos y para el umbral, calcular:

$$w_1 = w_1 + \alpha * E * x_1 = 0.84 + 0.3 * 0.217 * 0 = \mathbf{0.840}$$

$$w_2 = w_2 + \alpha * E * x_2 = 0.394 + 0.3 * 0.217 * 0 = \mathbf{0.394}$$

$$w_3 = w_3 + \alpha * E * x_3 = 0.783 + 0.3 * 0.217 * \quad \mathbf{0.8481}$$

$$1 =$$

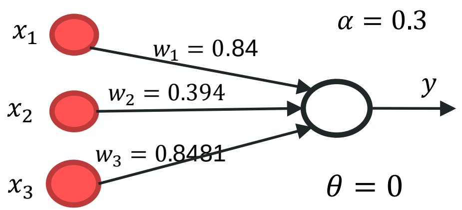
5. Redes ADALINE

EJEMPLO: Decodificador binario a decimal

Datos

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**

Paso #5: Modificar los pesos y el umbral y repetimos desde el **paso #2** hasta cumplir el criterio de parada.



Paso #2: Seleccionar un ejemplo x del conjunto de entrenamiento: $x_1 = 0$, $x_2 = 1$, $x_3 = 0$

Paso #3: Calcular la salida de la red: $y = f(w_1x_1 + \cdots + w_nx_n + \theta)$ y obtener la diferencia.

$$\boldsymbol{y = 0.84 * 0 + 0.394 * 1 + 0.8481 * 0 = 0.394}$$

$$E = |d^p - y^p| = |2 - 0.394| = 1.606$$

Paso #4: Para todos los pesos y para el umbral, calcular:

$$w_1 = w_1 + \alpha * E * x_1 = 0.84 + 0.3 * 1.606 * 0 = \mathbf{0.840}$$

$$w_2 = w_2 + \alpha * E * x_2 = 0.394 + 0.3 * 1.606 * 1 = \mathbf{0.876}$$

$$w_3 = w_3 + \alpha * E * x_3 = 0.8481 + 0.3 * 1.606 * \mathbf{0} = \mathbf{0.8481}$$

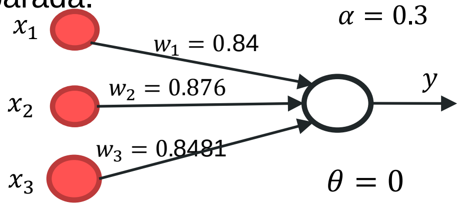
5. Redes ADALINE

EJEMPLO: Decodificador binario a decimal

Datos

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**

Paso #5: Modificar los pesos y el umbral y repetimos desde el **paso #2** hasta cumplir el criterio de parada.



Paso #2: Seleccionar un ejemplo x del conjunto de entrenamiento: x_1 =0, x_2 =1, x_3 = 1

Paso #3: Calcular la salida de la red: $y = f(w_1x_1 + \cdots + w_nx_n + \theta)$ y obtener la diferencia.

$$\boldsymbol{y = 0.84 * 0 + 0.876 * 1 + 0.8481 * 1 = 1.7241}$$

$$E = |d^p - y^p| = |3 - 0.394| = 1.2759$$

Paso #4: Para todos los pesos y para el umbral, calcular:

$$w_1 = w_1 + \alpha * E * x_1 = 0.84 + 0.3 * 1.2759 * 0 = \mathbf{0.840}$$

$$w_2 = w_2 + \alpha * E * x_2 = 0.876 + 0.3 * 1.2759 * 1 = \mathbf{1.259}$$

$$w_3 = w_3 + \alpha * E * x_3 = 0.8481 + 0.3 * 1.2759 * \mathbf{1} = \mathbf{1.231}$$

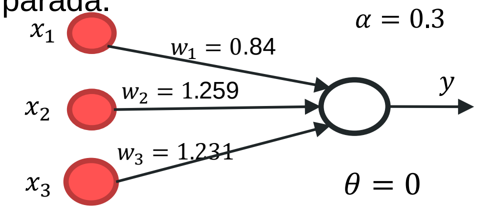
5. Redes ADALINE

EJEMPLO: Decodificador binario a decimal

Datos

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**

Paso #5: Modificar los pesos y el umbral y repetimos desde el **paso #2** hasta cumplir el criterio de parada.



Paso #2: Seleccionar un ejemplo x del conjunto de entrenamiento: $x_1 = 1$, $x_2 = 0$, $x_3 = 0$

Paso #3: Calcular la salida de la red: $y = f(w_1 x_1 + \cdots + w_n x_n + \theta)$ y obtener la diferencia.

$$\boldsymbol{y = 0.84 * 1 + 0.876 * 0 + 0.8481 * 0 =}\ \mathbf{0.84}$$

$$E = |d^p - y^p| = |4 - 0.84| = 3.16$$

Paso #4: Para todos los pesos y para el umbral, calcular:

$$w_1 = w_1 + \alpha * E * x_1 = 0.84 + 0.3 * 3.16 * 1 = \mathbf{1.788}$$

$$w_2 = w_2 + \alpha * E * x_2 = 1.259 + 0.3 * 3.16 * 0 = \mathbf{1.259}$$

$$w_3 = w_3 + \alpha * E * x_3 = 1.231 + 0.3 * 3.16 * \mathbf{0} = \mathbf{1.231}$$

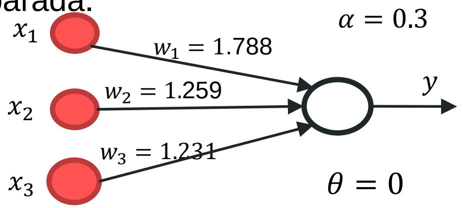
5. Redes ADALINE

EJEMPLO: Decodificador binario a decimal

Datos

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**

Paso #5: Modificar los pesos y el umbral y repetimos desde el **paso #2** hasta cumplir el criterio de parada.



Paso #2: Seleccionar un ejemplo x del conjunto de entrenamiento: $x_1 = 1$, $x_2 = 0$, $x_3 = 1$

Paso #3: Calcular la salida de la red: $y = f(w_1x_1 + \cdots + w_nx_n + \theta)$ y obtener la diferencia.

$$\boldsymbol{y = 1.788 * 1 + 1.259 * 0 + 1.231 * 1 =} \ \mathbf{3.019}$$

$$E = |d^p - y^p| = |5 - 3.019| = \ 1.981$$

Paso #4: Para todos los pesos y para el umbral, calcular:

$$w_1 = w_1 + \alpha * E * x_1 = 1.788 + 0.3 * 1.981 * 1 = \ \mathbf{2.382}$$

$$w_2 = w_2 + \alpha * E * x_2 = 1.259 + 0.3 * 1.981 * 0 \ \ \mathbf{1.259}$$

$$w_3 = w_3 + \alpha * E * x_3 = 1.231 + 0.3 * 1.981 * \mathbf{1} = \ \mathbf{1.825}$$

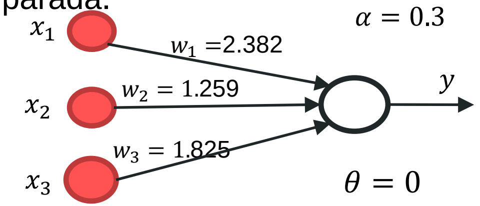
5. Redes ADALINE

EJEMPLO: Decodificador binario a decimal

Datos

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**

Paso #5: Modificar los pesos y el umbral y repetimos desde el **paso #2** hasta cumplir el criterio de parada.



Paso #2: Seleccionar un ejemplo x del conjunto de entrenamiento:

x_1 =1, x_2 =1, x_3 = 0

Paso #3: Calcular la salida de la red: $y = f(w_1x_1 + \cdots + w_nx_n + \theta)$ y obtener la diferencia.

$$\boldsymbol{y = 2.382 * 1 + 1.259 * 1 + 1.825 * 0 =} \ \ \mathbf{3.641}$$

$$E = |d^p - y^p| = |6 - 3.641| = 2.359$$

Paso #4: Para todos los pesos y para el umbral, calcular:

$$w_1 = w_1 + \alpha * E * x_1 = 2.382 + 0.3 * 2.359 * 1 = \mathbf{3.09}$$

$$w_2 = w_2 + \alpha * E * x_2 = 1.259 + 0.3 * 2.359 * 1 = \mathbf{1.967}$$

$$w_3 = w_3 + \alpha * E * x_3 = 1.825 + 0.3 * 2.359 * \mathbf{0} = \ \ \mathbf{1.825}$$

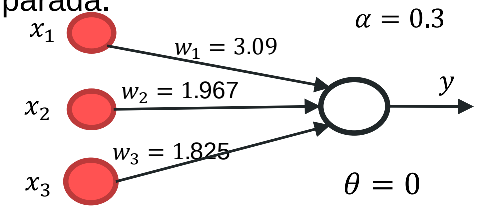
5. Redes ADALINE

EJEMPLO: Decodificador binario a decimal

Datos

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**

Paso #5: Modificar los pesos y el umbral y repetimos desde el **paso #2** hasta cumplir el criterio de parada.



Paso #2: Seleccionar un ejemplo x del conjunto de entrenamiento: $x_1 = 1$, $x_2 = 1$, $x_3 = 1$

Paso #3: Calcular la salida de la red: $y = f(w_1 x_1 + \cdots + w_n x_n + \theta)$ y obtener la diferencia.

$$\boldsymbol{y = 3.09 * 1 + 1.967 * 1 + 1.825 * 1 = 6.882}$$

$$E = |d^p - y^p| = |7 - 6.882| = 0.118$$

Paso #4: Para todos los pesos y para el umbral, calcular:

Resultado después de la primera iteración del entrenamiento

$$\begin{cases} w_1 = w_1 + \alpha * E * x_1 = 3.09 + 0.3 * 0.118 * 1 = \mathbf{3.125} \\ w_2 = w_2 + \alpha * E * x_2 = 1.967 + 0.3 * 0.118 * 1 = \mathbf{2.002} \\ w_3 = w_3 + \alpha * E * x_3 = 1.825 + 0.3 * 0.118 * 1 = \mathbf{1.860} \end{cases}$$

5. Redes ADALINE

<u>EJEMPLO</u>: Decodificador binario a decimal

Visualización de los pesos según iteraciones

Iteración	w1	w2	w3
1	3.13	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.

5. Redes ADALINE

APRENDIZAJE: Seudocódigo

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.

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.

6. Ventajas / Limitaciones

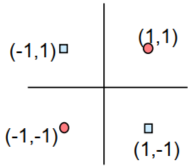
VENTAJAS

- 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 (características o entradas).
- El uso del Perceptrón o de las redes ADALINE permite que cualquier sistema (caja negra), se puede representar por una red.

LIMITACIONES

- Estas técnicas poseen grandes limitaciones.
- Sólo pueden resolver sistemas donde los ejemplos o características (entradas) son linealmente separables (por ejemplo no resolverá el XOR Exclusivo para el cual 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

PREGUNTAS

Dudas y opiniones

Preguntas?