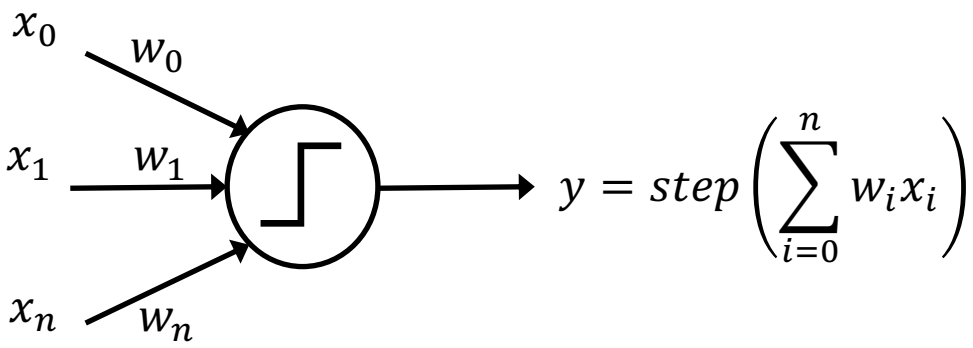
Tema 2: Entrenamiento del perceptrón

1. Perceptrón, función de activación $step()$.
2. Cálculo de los pesos del perceptrón, entrenamiento.
3. Diseño de un perceptrón que aprende la función OR lógica.
4. Patrones no separables linealmente: función lógica XOR.

Perceptrón con función $step()$:

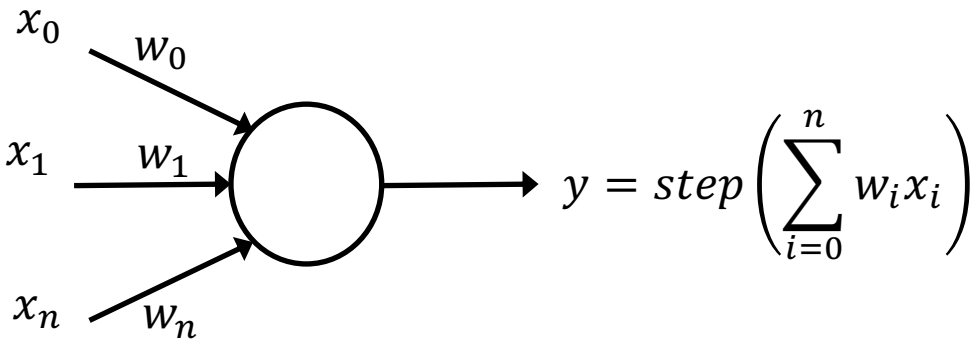
- Devuelve 1 si la suma ponderada de las entradas es mayor que un umbral.
  - $y = 1 \Rightarrow \sum_{i=0}^{n} w_i x_i > t.$
- Devuelve 0 si la suma ponderada de las entradas es menor o igual que un umbral.
  - $y = 0 \Rightarrow \sum_{i=0}^{n} w_i x_i \leq t.$



Perceptrón:

- ¿Cómo calculamos los pesos del perceptrón?
  - Entrenándolo con un conjunto de datos de los que conocemos sus clasificaciones.

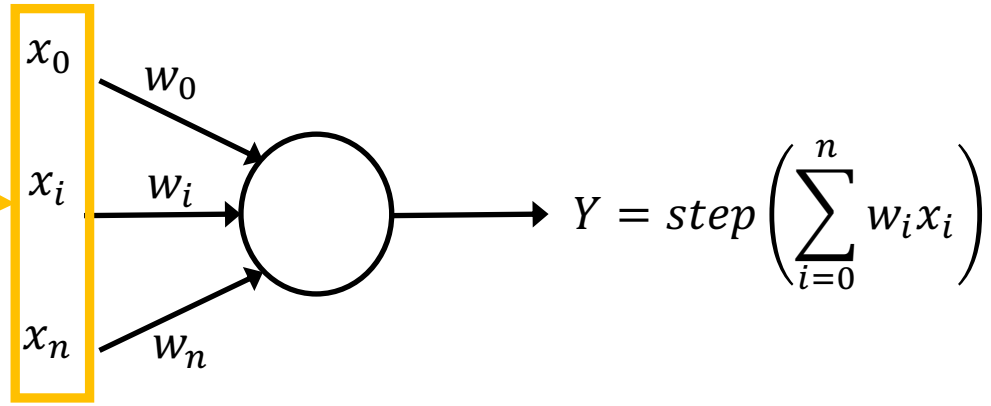
	x_0	x_1	$\cdots$	x_n	y_d
Ejemplo1	3	5	$\cdots$	7	1
Ejemplo2	0	3	$\cdots$	4	0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
EjemploN	-4	2	$\cdots$	0	1



Perceptrón:

- ¿Cómo calculamos los pesos del perceptrón?
  - Entrenándolo con un conjunto de datos de los que conocemos sus clasificaciones.

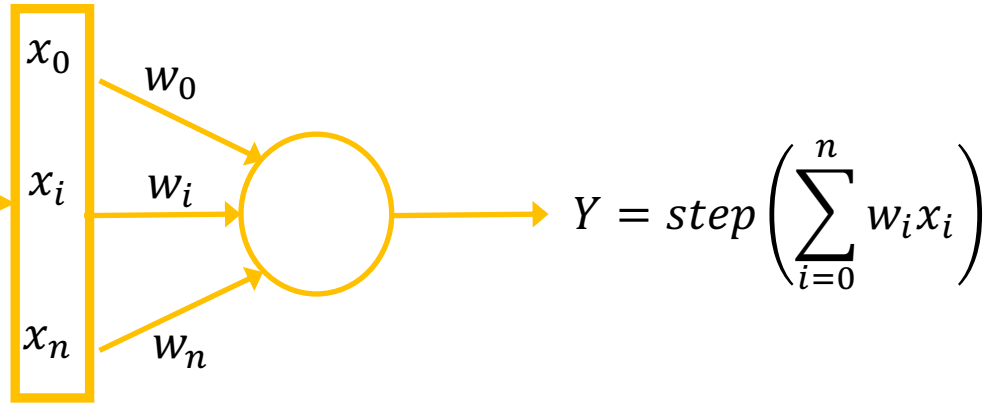
	x_0	x_1	$\cdots$	x_n	Y_d
Ejemplo1	3	5	$\cdots$	7	1
Ejemplo2	0	3	$\cdots$	4	0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
EjemploN	-4	2	$\cdots$	0	1



Perceptrón:

- ¿Cómo calculamos los pesos del perceptrón?
  - Entrenándolo con un conjunto de datos de los que conocemos sus clasificaciones.

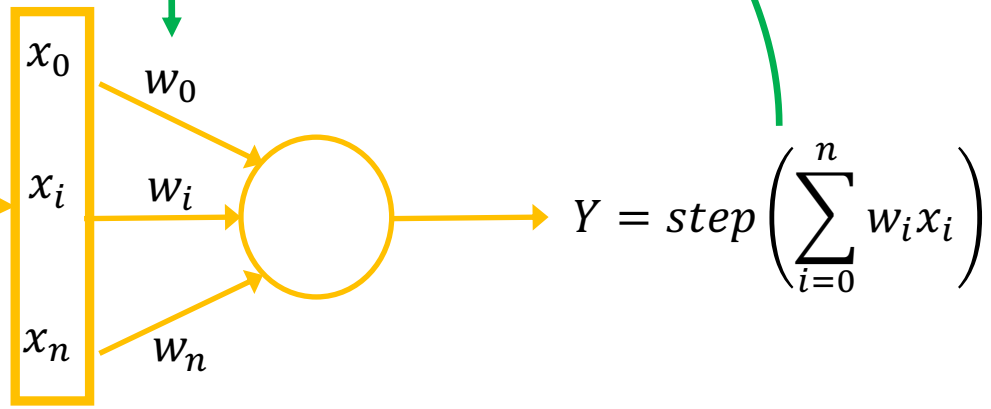
	x_0	x_1	⋯	x_n	Y_d
Ejemplo1	3	5	⋯	7	1
Ejemplo2	0	3	⋯	4	0
⋯	⋯	⋯	⋯	⋯	⋯
EjemploN	-4	2	⋯	0	1



Perceptrón:

- ¿Cómo calculamos los pesos del perceptrón?
  - Entrenándolo con un conjunto de datos de los que conocemos sus clasificaciones.

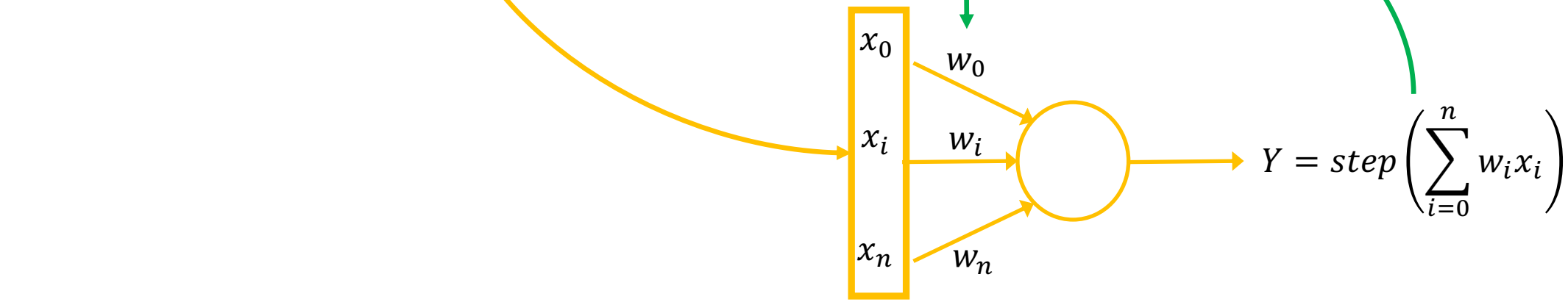
	x_0	x_1	⋯	x_n	Y_d
Ejemplo1	3	5	⋯	7	1
Ejemplo2	0	3	⋯	4	0
⋯	⋯	⋯	⋯	⋯	⋯
EjemploN	-4	2	⋯	0	1



Perceptrón:

- ¿Cómo calculamos los pesos del perceptrón?
  - Entrenándolo con un conjunto de datos de los que conocemos sus clasificaciones.

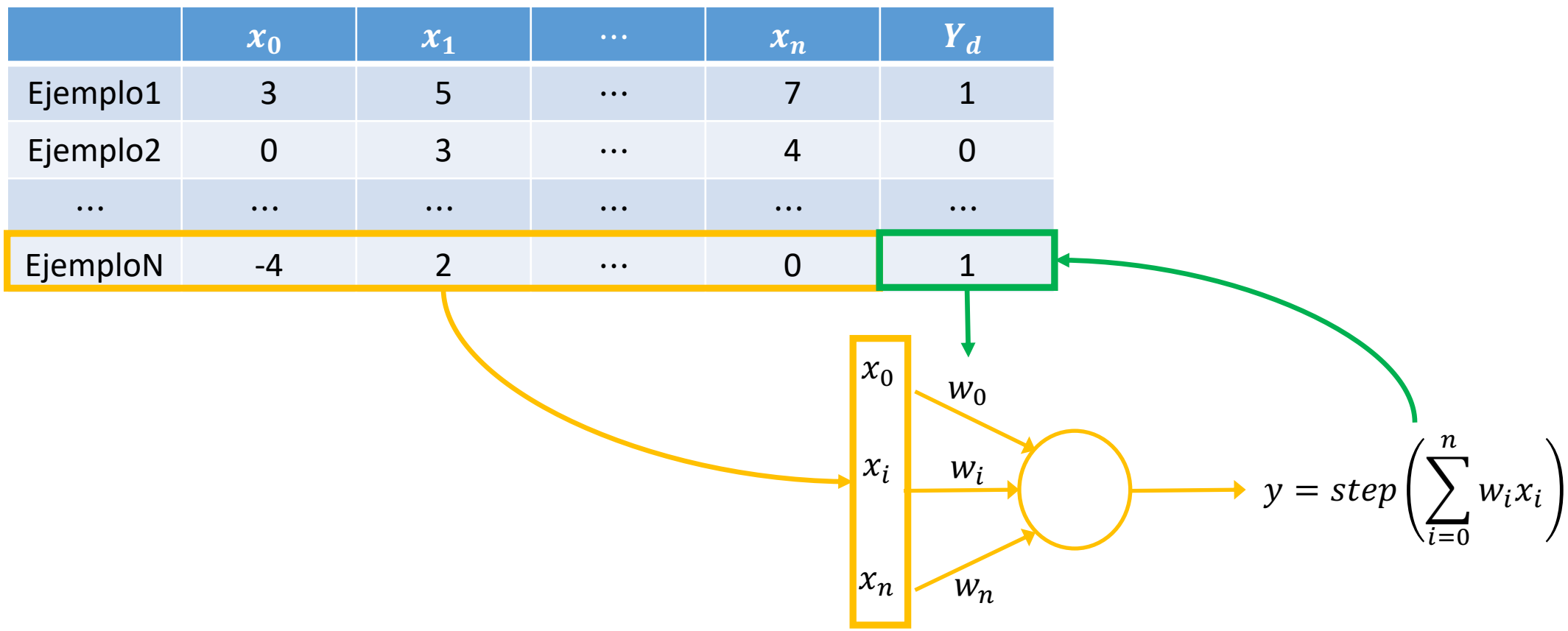
	x_0	x_1	⋯	x_n	Y_d
Ejemplo1	3	5	⋯	7	1
Ejemplo2	0	3	⋯	4	0
⋯	⋯	⋯	⋯	⋯	⋯
EjemploN	-4	2	⋯	0	1



Perceptrón:

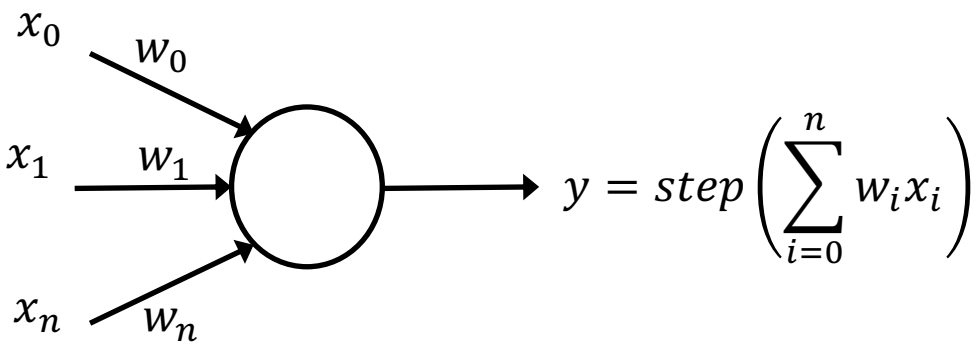
- ¿Cómo calculamos los pesos del perceptrón?
  - Entrenándolo con un conjunto de datos de los que conocemos sus clasificaciones.

	x_0	x_1	$\cdots$	x_n	Y_d
Ejemplo1	3	5	$\cdots$	7	1
Ejemplo2	0	3	$\cdots$	4	0
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
EjemploN	-4	2	$\cdots$	0	1



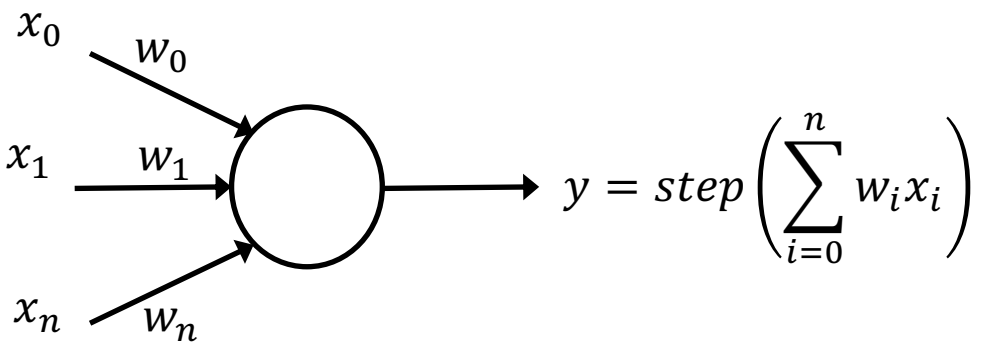
Perceptrón:

- Los pesos se calculan entrenando al perceptron datos conocidos de entrenamiento.
  - Proceso de aprendizaje:
    1. Se asignan pesos aleatorios a las entradas, típicamente $w_i \in (-0.5,\ 0.5)$.



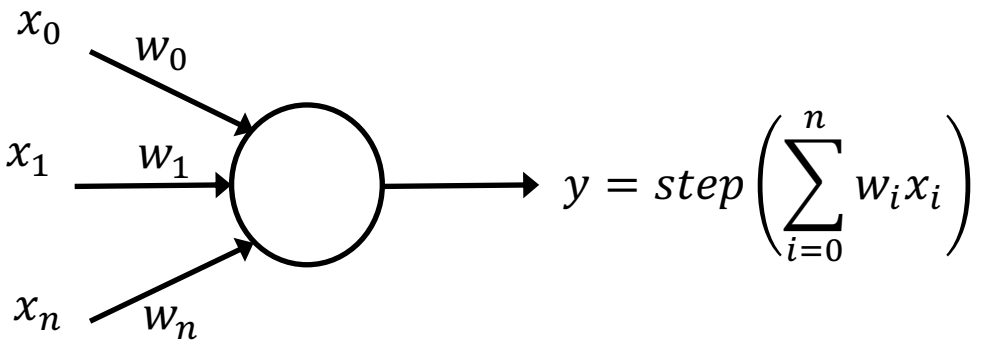
Perceptrón:

- Los pesos se calculan entrenando al perceptron datos conocidos de entrenamiento.
  - Proceso de aprendizaje:
    1. Se asignan pesos aleatorios a las entradas, típicamente $w_i \in (-0.5,\ 0.5)$.
    2. Se presenta el siguiente conjunto de datos de entrenamiento en las entradas y se calcula la salida.



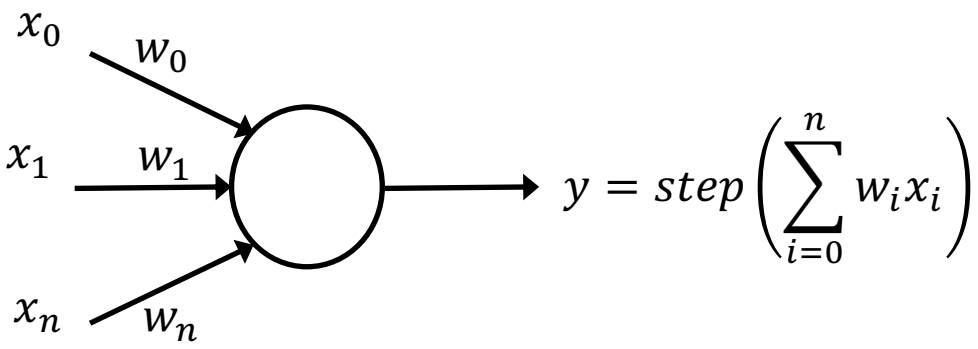
Perceptrón:

- Los pesos se calculan entrenando al perceptron datos conocidos de entrenamiento.
  - Proceso de aprendizaje:
    1. Se asignan pesos aleatorios a las entradas, típicamente $w_i \in (-0.5,\ 0.5)$.
    2. Se presenta el siguiente conjunto de datos de entrenamiento en las entradas y se calcula la salida.
    3. Si la salida es incorrecta se cambian los valores de los pesos con el fin de conseguir una salida correcta.



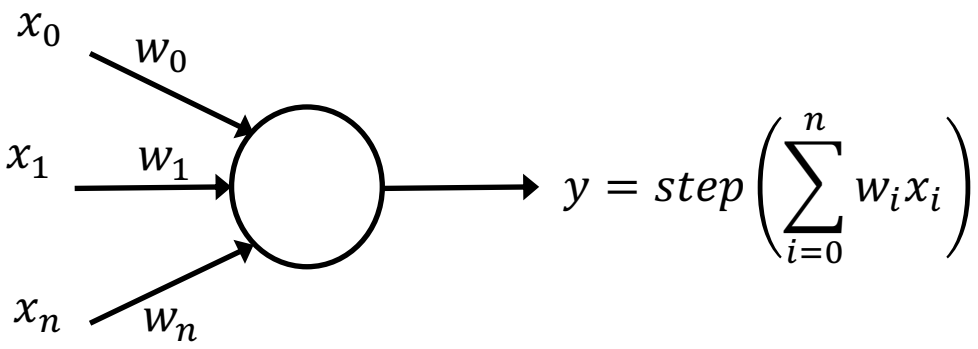
Perceptrón:

- Los pesos se calculan entrenando al perceptron datos conocidos de entrenamiento.
  - Proceso de aprendizaje:
    4. Regla de entrenamiento del perceptron:
       - $w_i \leftarrow w_i + (\alpha \times (y_d - y) \times x_i)$, α es el ritmo de aprendizaje, con valor entre 0 y 1.



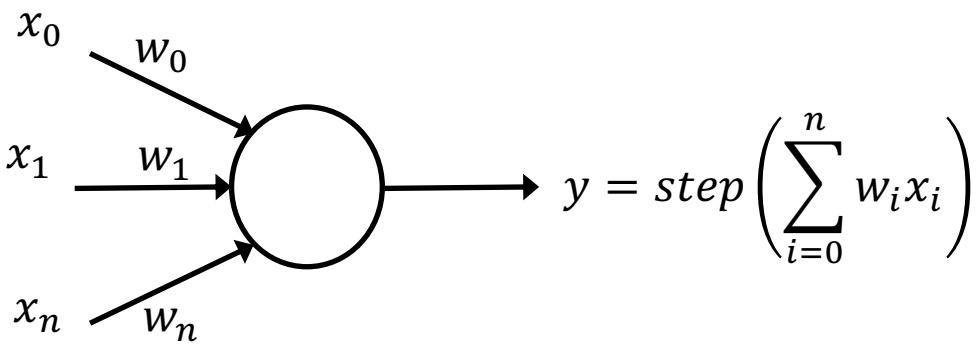
Perceptrón:

- Los pesos se calculan entrenando al perceptron datos conocidos de entrenamiento.
  - Proceso de aprendizaje:
    4. Regla de entrenamiento del perceptron:
       - $w_i \leftarrow w_i + (\alpha \times (y_d - y) \times x_i)$, α es el ritmo de aprendizaje, con valor entre 0 y 1.
    5. Una vez se actualizan los pesos analizamos los siguientes datos de entrenamiento y procedemos de la misma manera.



Perceptrón:

- Los pesos se calculan entrenando al perceptron datos conocidos de entrenamiento.
  - Proceso de aprendizaje:
    4. Regla de entrenamiento del perceptron:
       - $w_i \leftarrow w_i + (\alpha \times (y_d - y) \times x_i)$, α es el ritmo de aprendizaje, con valor entre 0 y 1.
    5. Una vez se actualizan los pesos analizamos los siguientes datos de entrenamiento y procedemos de la misma manera.
    6. A cada iteración completa del conjunto de datos se le llama época (época).



Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $a = 0.2$.

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $a = 0.2$.

	x_0	x_1	y_d
Ejemplo1	0	0	0
Ejemplo2	0	1	1
Ejemplo3	1	0	1
Ejemplo4	1	1	1

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $a = 0.2$.
  1. Inicializamos los pesos $w_0 = -0.2$ y $w_1 = 0.4$.

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $a = 0.2$.
  1. Inicializamos los pesos $w_0 = -0.2$ y $w_1 = 0.4$.
  2. 1er ejemplo: $x_0 = 0$ y $x_1 = 0 \Rightarrow$ Valor esperado: $(y_d = x_0$ OR $x_1 = 0)$.
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 0 + 0.4 \cdot 0) = 0$.
     - Los coeficientes no cambian , ya que $(y_d - y) = 0 - 0 = 0$.

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $a = 0.2$.
  1. Inicializamos los pesos $w_0 = -0.2$ y $w_1 = 0.4$.
  2. 1er ejemplo: $x_0 = 0$ y $x_1 = 0 \Rightarrow$ Valor esperado: ($y_d = x_0$ OR $x_1 = 0$).
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 0 + 0.4 \cdot 0) = 0$.
     - Los coeficientes no cambian , ya que $(y_d - y) = 0 - 0 = 0$.
  3. 2º ejemplo: $x_0 = 0$ y $x_1 = 1 \Rightarrow$ Valor esperado: ($y_d = x_0$ OR $x_1 = 1$).
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 0 + 0.4 \cdot 1) = 1$.
     - Los coeficientes no cambian , ya que $(y_d - y) = 1 - 1 = 0$.

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $a = 0.2$.
  1. Inicializamos los pesos $w_0 = -0.2$ y $w_1 = 0.4$.
  2. 1er ejemplo: $x_0 = 0$ y $x_1 = 0 \Rightarrow$ Valor esperado: $y_d = (x_0 \text{ OR } x_1) = 0$.
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 0 + 0.4 \cdot 0) = 0$.
     - Los coeficientes no cambian , ya que $(y_d - y) = 0 - 0 = 0$.
  3. 2º ejemplo: $x_0 = 0$ y $x_1 = 1 \Rightarrow$ Valor esperado: $y_d = (x_0 \text{ OR } x_1) = 1$.
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 0 + 0.4 \cdot 1) = 1$.
     - Los coeficientes no cambian , ya que $(y_d - y) = 1 - 1 = 0$.

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $\alpha = 0.2$.
  4. 3º ejemplo: $x_0 = 1$ y $x_1 = 0 \Rightarrow$ Valor esperado: $y_d = (x_0 \text{ OR } x_1) = 1$.
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 1 + 0.4 \cdot 0) = 0$.
     - Hay que recalcular los coeficientes, ya que $(y_d - y) = 1 - 0 = 1$.

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $\alpha = 0.2$.
  4. 3º ejemplo: $x_0 = 1$ y $x_1 = 0 \Rightarrow$ Valor esperado: $y_d = (x_0 \text{ OR } x_1) = 1$.
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 1 + 0.4 \cdot 0) = 0$.
     - Hay que recalcular los coeficientes, ya que $(y_d - y) = 1 - 0 = 1$.
       - $w_i \leftarrow w_i + \alpha \times (y_d - y) \times x_i$.
       - $w_0 \leftarrow -0.2 + (0.2 \times 1 \times 1) = 0$.
       - $w_1 \leftarrow 0.4 + (0.2 \times 0 \times 1) = 0.4$. w_1 no ha contribuido a este error por eso no se ajusta.

Perceptrón:

- Ejemplo: utilizar un perceptron para representar la función lógica OR con dos entradas, $t = 0$ y $\alpha = 0.2$.
  4. 3º ejemplo: $x_0 = 1$ y $x_1 = 0 \Rightarrow$ Valor esperado: $y_d = (x_0 \text{ OR } x_1) = 1$.
     - Obtenemos: $y = Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 1 + 0.4 \cdot 0) = 0$.
     - Hay que recalcular los coeficientes, ya que $(y_d - y) = 1 - 1 = 0$.
       - $w_i \leftarrow w_i + \alpha \times (y_d - y) \times x_i$.
       - $w_0 \leftarrow -0.2 + (0.2 \times 1 \times 1) = 0$.
       - $w_1 \leftarrow 0.4 + (0.2 \times 0 \times 1) = 0.4$. w_1 no ha contribuido a este error por eso no se ajusta.
  5. 4º ejemplo : $x_0 = 1$ y $x_1 = 1 \Rightarrow$ Valor esperado: $y_d = (x_0 \text{ OR } x_1) = 1$
     - Obtenemos: y $= Step(\sum_{i=0}^{n} w_i x_i) = Step(-0.2 \cdot 1 + 0.4 \cdot 1) = 1$.
     - Los coeficientes no cambian , ya que $(y_d - y) = 1 - 1 = 0$.

Perceptron:

1ª Época

Época	x_0	x_1	y_d	y	$y_d - y$	w_0	w_1
1	0	0	0	0	0	-0.2	0.4
1	0	1	1	1	0	-0.2	0.4
1	1	0	1	0	1	0	0.4
1	1	1	1	1	0	0	0.4
2	0	0	0	0	0	0	0.4
2	0	1	1	1	0	0	0.4
2	1	0	1	0	1	0.2	0.4
2	1	1	1	1	0	0.2	0.4
3	0	0	0	0	0	0.2	0.4
3	0	1	1	1	0	0.2	0.4
3	1	0	1	1	0	0.2	0.4
3	1	1	1	1	0	0.2	0.4

Perceptron:

Época	x_0	x_1	y_d	y	$y_d - y$	w_0	w_1
1	0	0	0	0	0	-0.2	0.4
1	0	1	1	1	0	-0.2	0.4
1	1	0	1	0	1	0	0.4
1	1	1	1	1	0	0	0.4
2	0	0	0	0	0	0	0.4
2	0	1	1	1	0	0	0.4
2	1	0	1	0	1	0.2	0.4
2	1	1	1	1	0	0.2	0.4
3	0	0	0	0	0	0.2	0.4
3	0	1	1	1	0	0.2	0.4
3	1	0	1	1	0	0.2	0.4
3	1	1	1	1	0	0.2	0.4

2ª Época

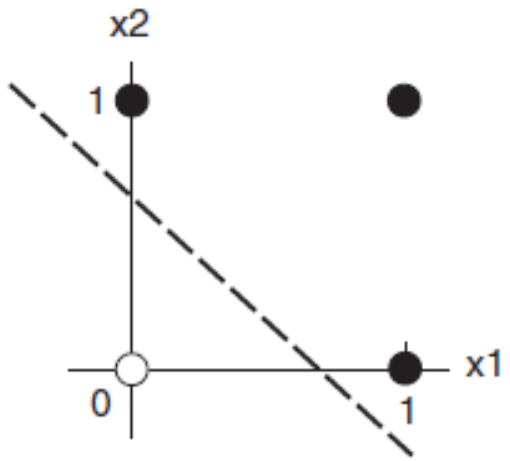
Perceptron:

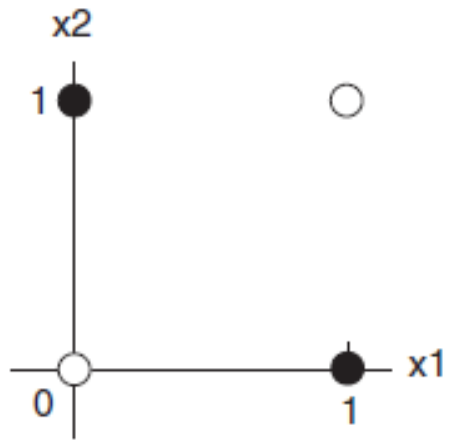
Época	x_0	x_1	y_d	y	$y_d - y$	w_0	w_1
1	0	0	0	0	0	-0.2	0.4
1	0	1	1	1	0	-0.2	0.4
1	1	0	1	0	1	0	0.4
1	1	1	1	1	0	0	0.4
2	0	0	0	0	0	0	0.4
2	0	1	1	1	0	0	0.4
2	1	0	1	0	1	0.2	0.4
2	1	1	1	1	0	0.2	0.4
3	0	0	0	0	0	0.2	0.4
3	0	1	1	1	0	0.2	0.4
3	1	0	1	1	0	0.2	0.4
3	1	1	1	1	0	0.2	0.4

3ª Época

Perceptron:

- Un perceptron se podría utilizar también para aprender una función AND pero no una XOR, la razón es que la función AND es separable linealmente y la XOR no.





Bibliografía:

- Artificial Intelligence A Modern Approach. Stuart Russell and Peter Norvig. Third Edition.
- Artificial Intelligence Illuminated. Ben Coppin. First Edition.