

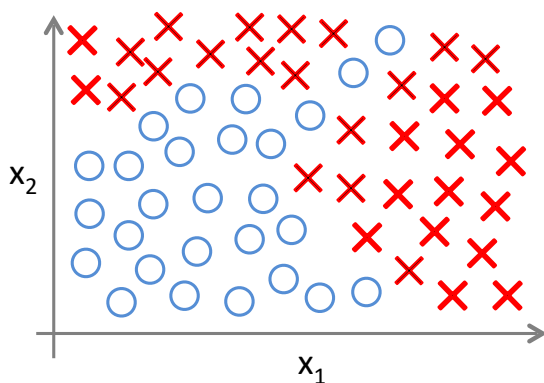
# Representación de Redes neuronales: Hipótesis no lineales



Machine Learning Aplicado

1

## Clasificación no lineal



$x_1$  = size  
 $x_2$  = # bedrooms  
 $x_3$  = # floors  
 $x_4$  = age  
...  
 $x_{100}$

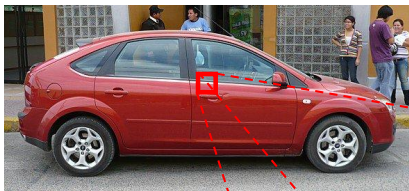
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \dots)$$

Machine Learning Aplicado

2

# No todo es lo que parece

Si usted ve esto:

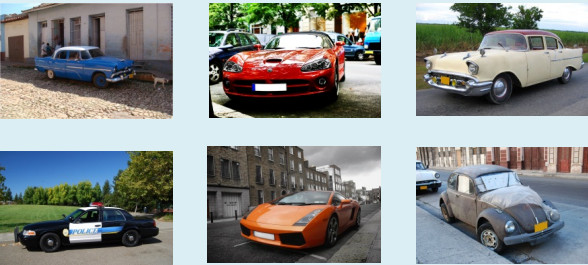


La cámara ve esto:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Machine Learning Aplicado

# Computer Vision: Detección



Autos



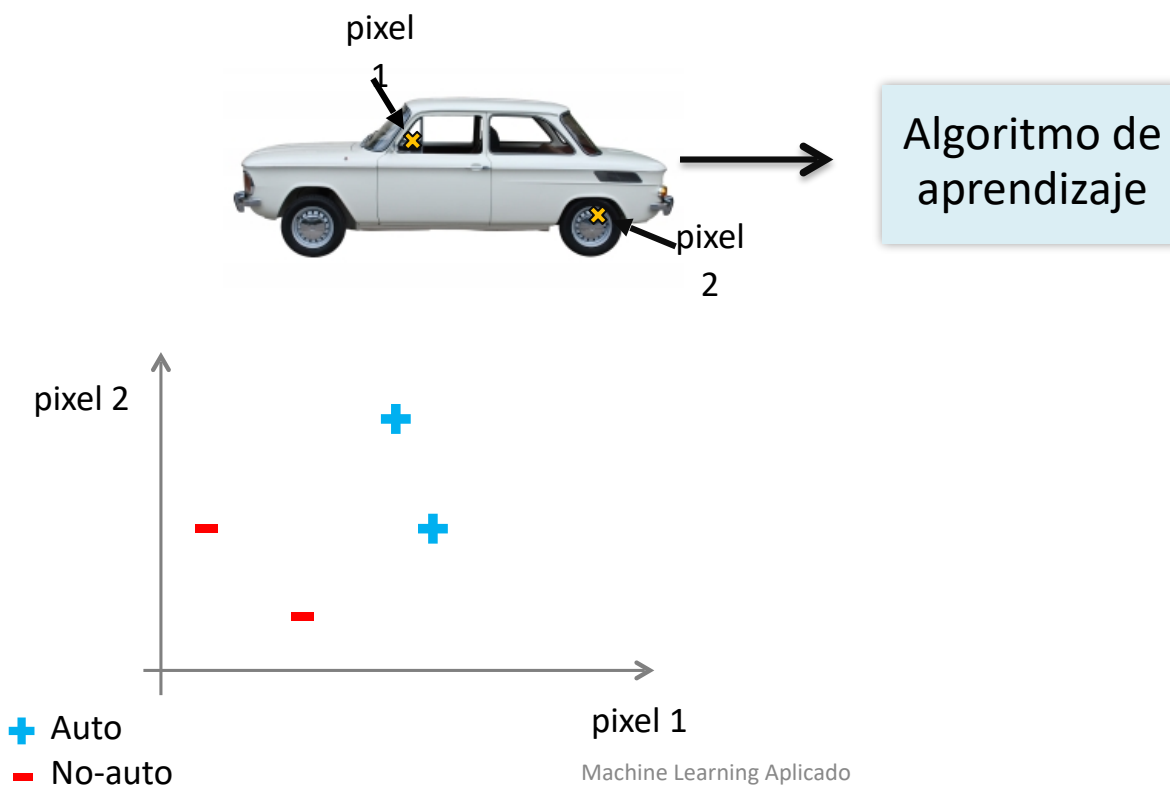
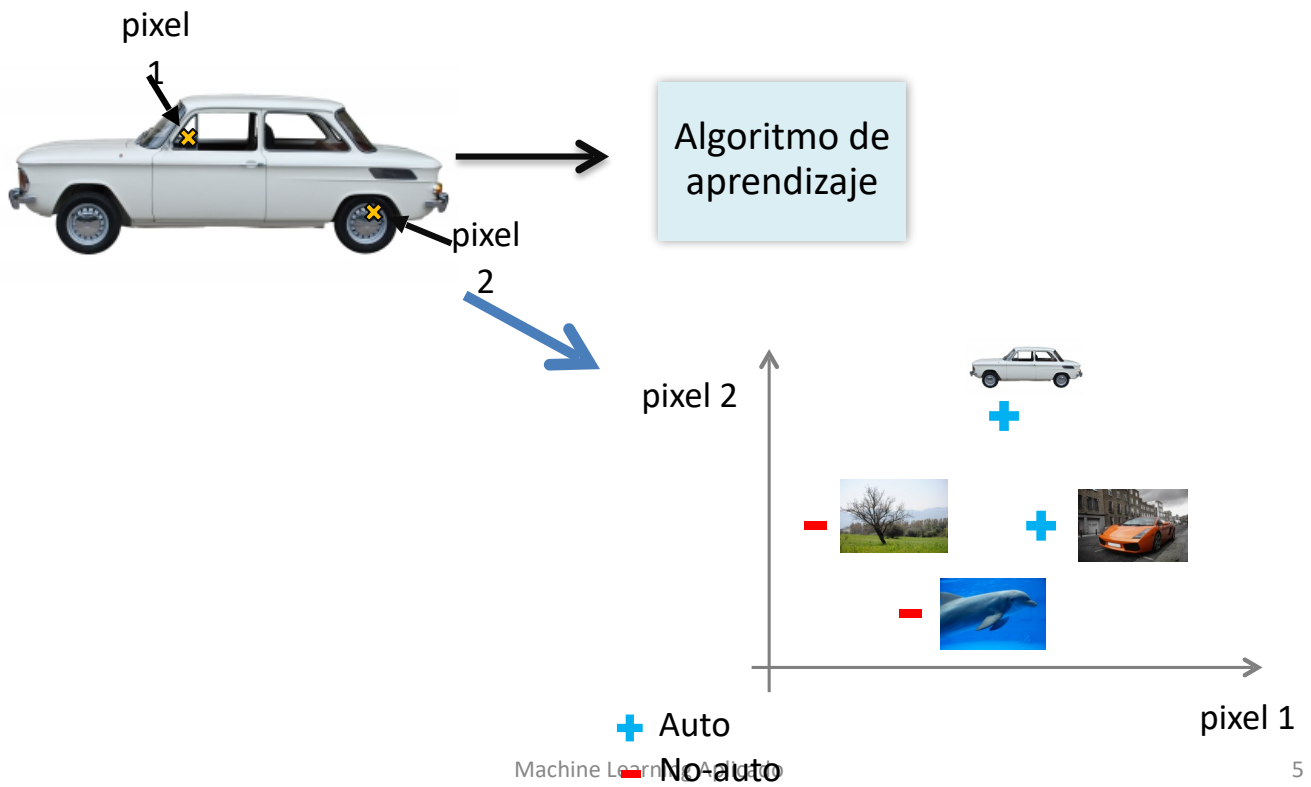
No un auto

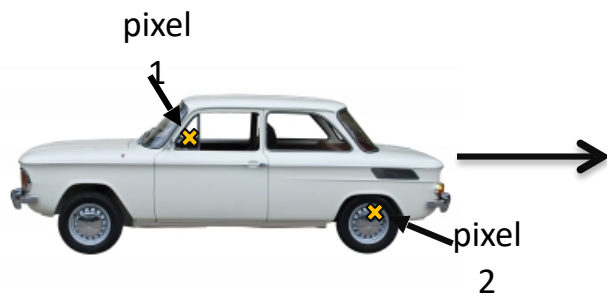
Testeo:



¿Que es esto?

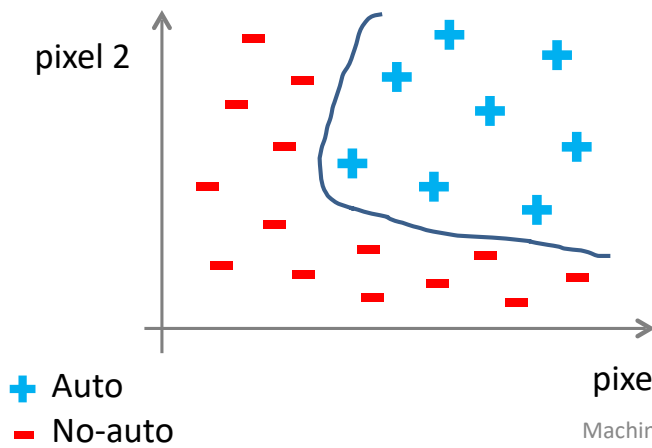
Machine Learning Aplicado





Algoritmo de aprendizaje

50 x 50 pixel images  $\rightarrow$  2500 pixels  
 $n = 2500$  (7500 if RGB)



$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix} \quad [0-255]$$

Features cuadráticos ( $x_i \times x_j$ ):  $\approx 3e6$  features

# Representación de Redes Neuronales

---

## Neuronas y el cerebro

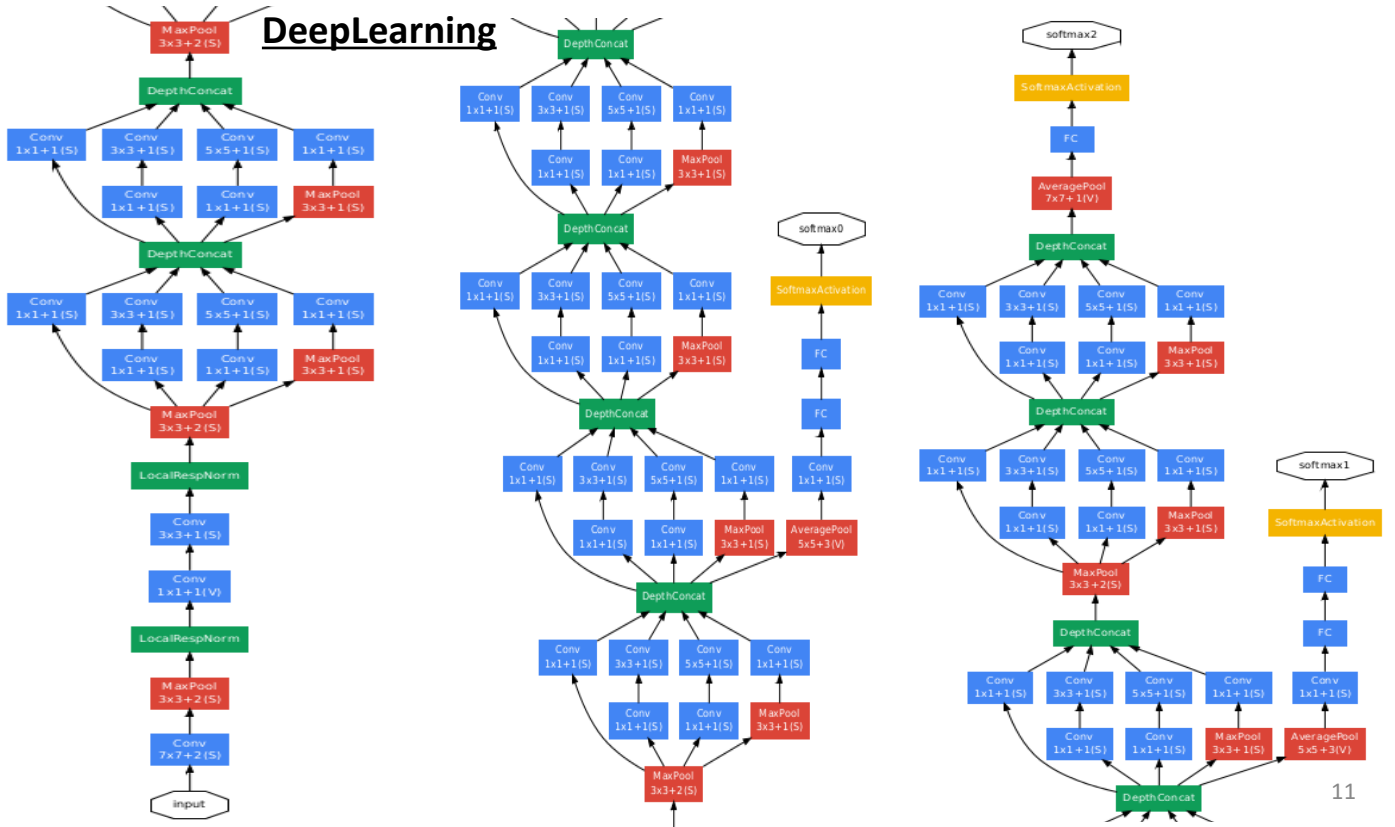
### **Redes neuronales**

**Origenes:** Algoritmos que tratan de imitar al cerebro.

Fue ampliamente usada en los 80s y en los 90s; su popularidad disminuyó a final de los 90s.

Contra-ataque: Estado de arte para muchas aplicaciones

# DeepLearning



11

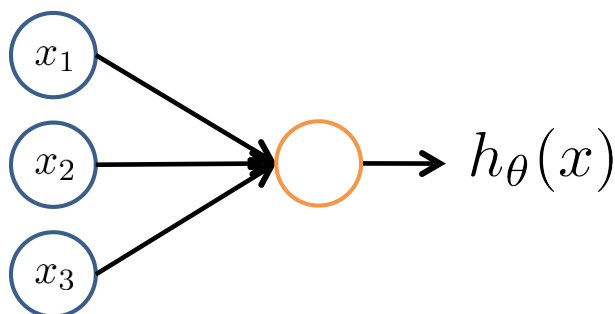
# Representación de redes neuronales

## Modelo de representación I

Machine Learning Aplicado

13

### Modelo neuronal : Unidad logistica



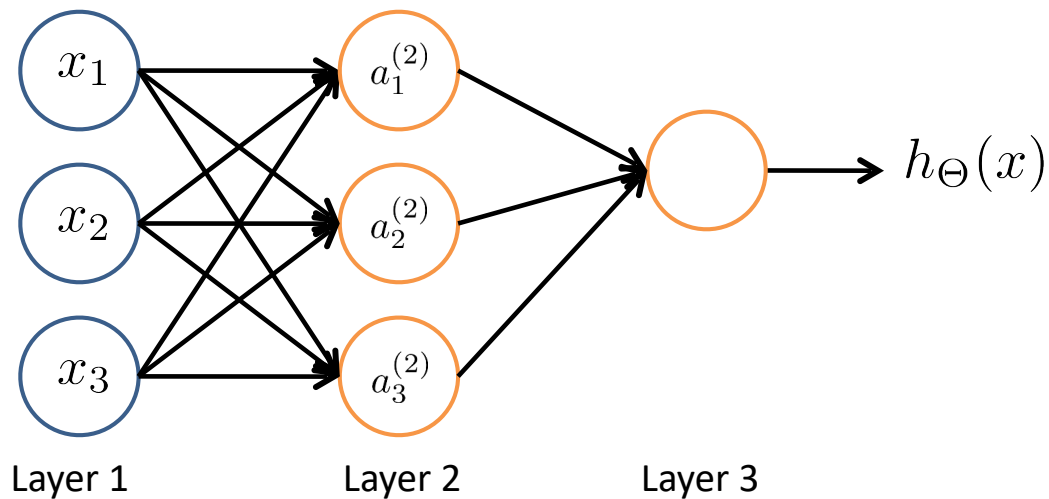
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Función de activación logistica.

Machine Learning Aplicado

14

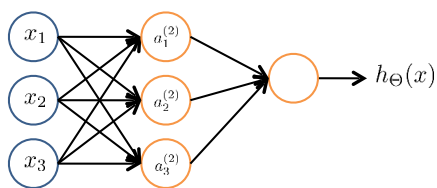
## Red neuronal



Machine Learning Aplicado

15

## Red neuronal



$a_i^{(j)}$  = Activación de unidad  $i$  en capa  $j$

$\Theta^{(j)}$  = Matriz de pesos que mapea capa  $j$  a capa  $j+1$

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

$$h_{\Theta}(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

Si red tiene  $s_{j+1}$  unidades en capa  $j+1$  y  $s_j$  unidades en capa  $j$ ; luego  $\Theta^{(j)}$  tendrá dimensionalidad:  $s_{j+1} \times (s_j + 1)$

Machine Learning Aplicado

16

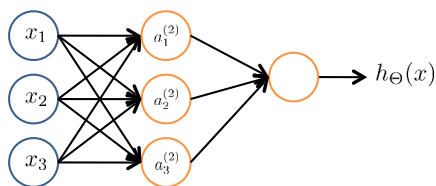


## Representación de redes neuronales

---

## Modelo de representación II

## Forward propagation: Vectorización



$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

$$h_{\Theta}(x) = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

$$z^{(2)} = \Theta^{(1)} x$$

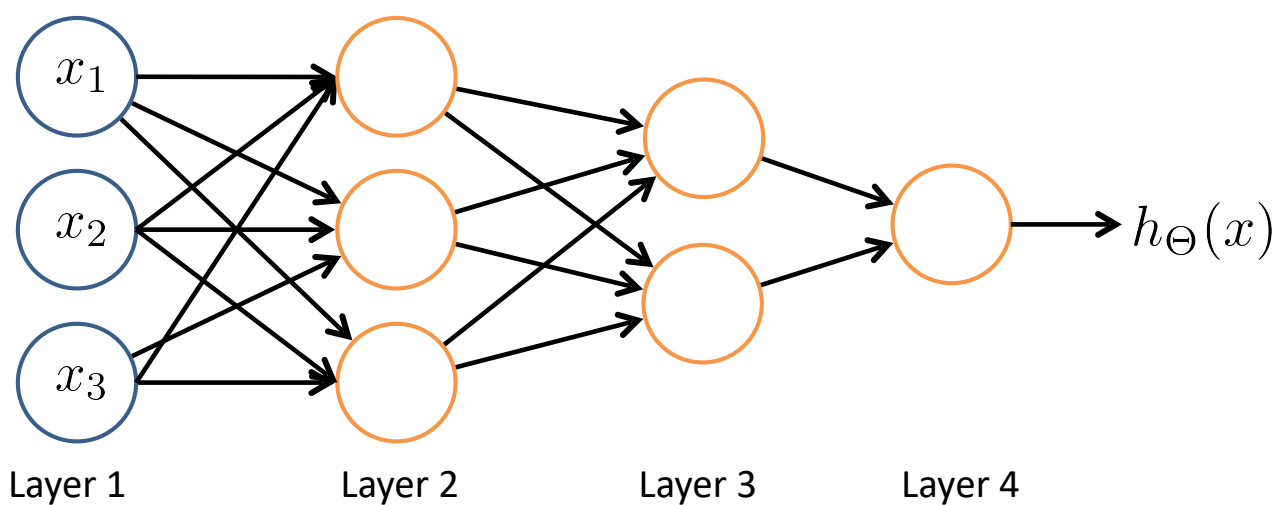
$$a^{(2)} = g(z^{(2)})$$

Add  $a_0^{(2)} = 1$ .

$$z^{(3)} = \Theta^{(2)} a^{(2)}$$

$$h_{\Theta}(x) = a^{(3)} = g(z^{(3)})$$

## Otras arquitecturas de redes.



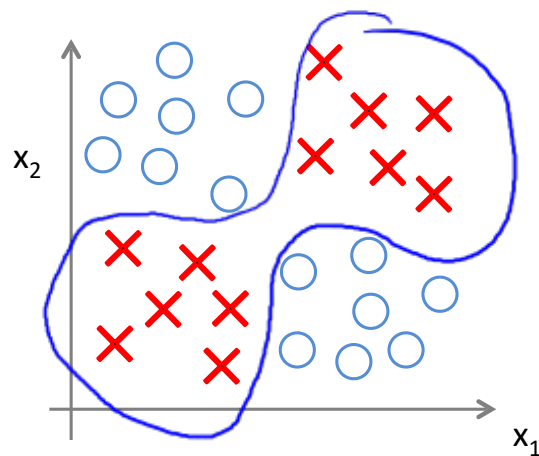
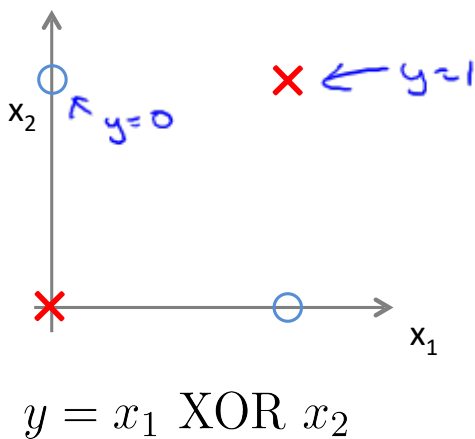
# Representación de redes neuronales

## Ejemplos e intuiciones I

Machine Learning Aplicado

### Ejemplo de clasificación no lineal: XNOR

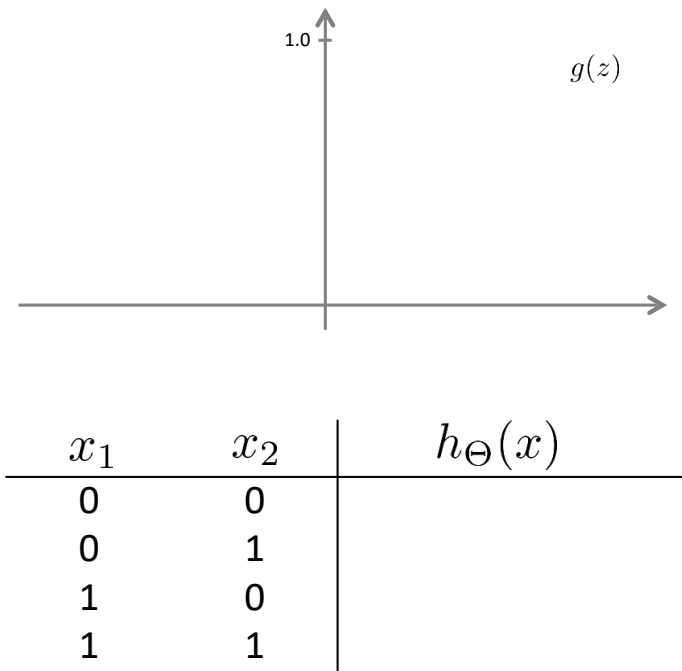
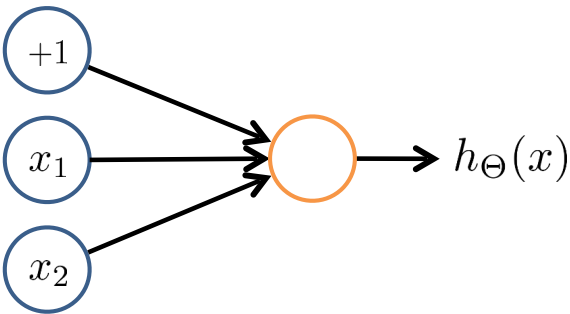
$x_1, x_2$  son binarios (0 o 1).



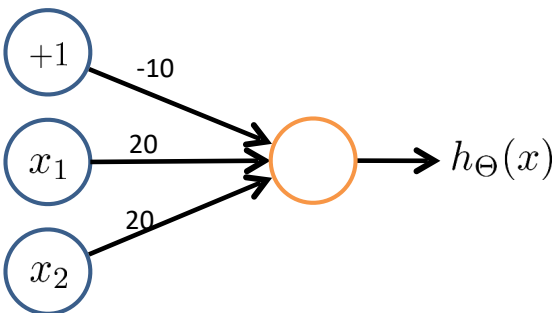
Machine Learning Aplicado

# Ejemplo: AND

$x_1, x_2 \in \{0, 1\}$   
 $y = x_1 \text{ AND } x_2$

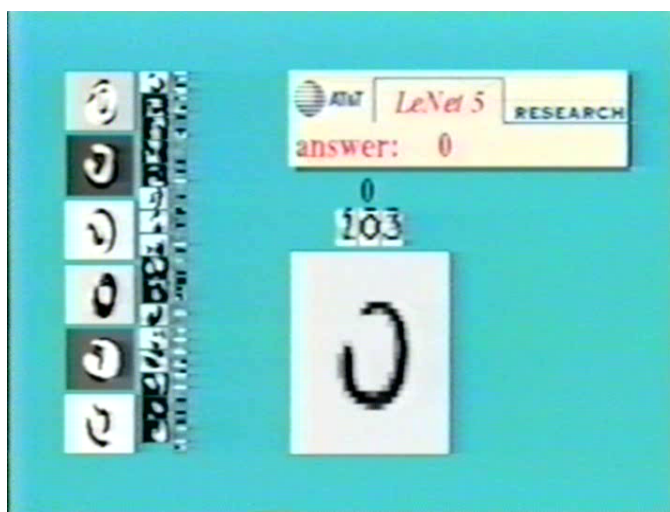


# Ejemplo: OR



$x_1$	$x_2$	$h_{\Theta}(x)$
0	0	
0	1	
1	0	
1	1	

## Clasificación de números manuscritos



<http://yann.lecun.com/exdb/lenet/index.html>

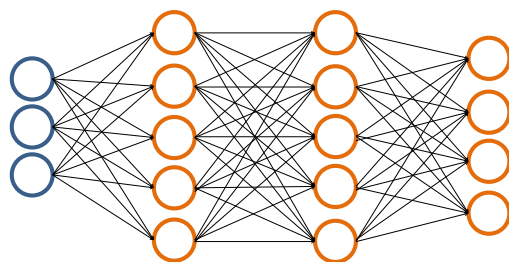
# Representación de redes neuronales

## Clasificación multiclase

Machine Learning Aplicado

27

### Múltiples salidas: One-vs-all.



$$h_{\Theta}(x) \in \mathbb{R}^4$$

$$h_{\Theta}(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ persona}, \quad h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ auto}, \quad h_{\Theta}(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ motocicleta}, \quad \text{etc.}$$

Training set:  $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

$$y^{(i)} \text{ es } \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ Persona}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \text{ auto}, \quad \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \text{ motocicleta}, \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ bus}$$

Machine Learning Aplicado

28