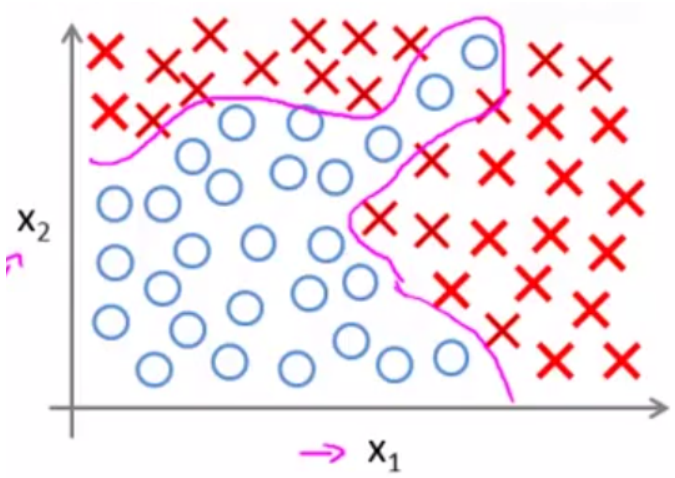
Redes Neurais:
Representação

Prof.: Eric A. Antonelo

Slides baseados no curso de *Machine Learning* de Andrew Ng

DAS-UFSC

Classificação não-linear



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

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

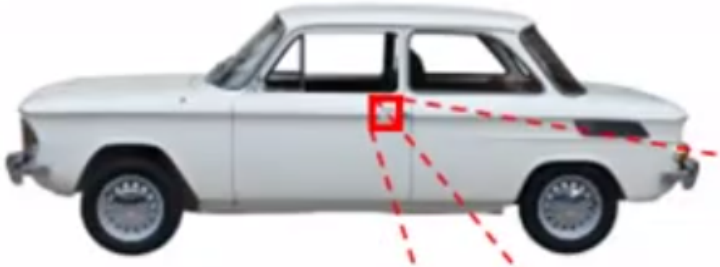
$n = 100$

Classificação não-linear

- O que é isso?
  - Você vê assim:

Classificação não-linear

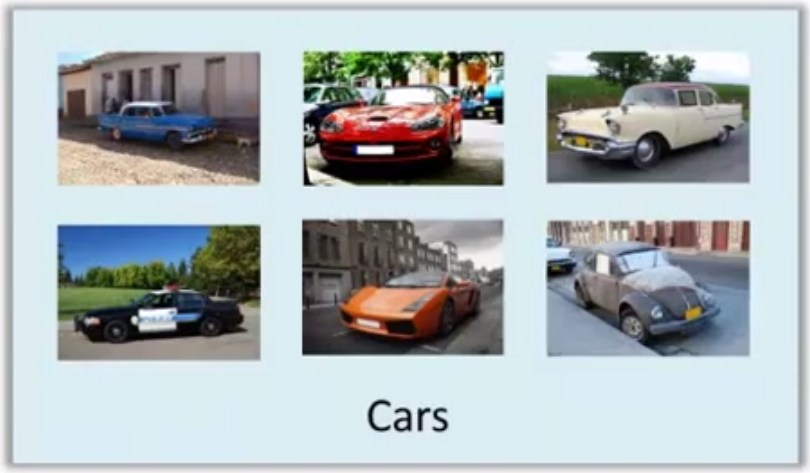
- O que é isso?
  - Você vê assim:



But the camera sees this:

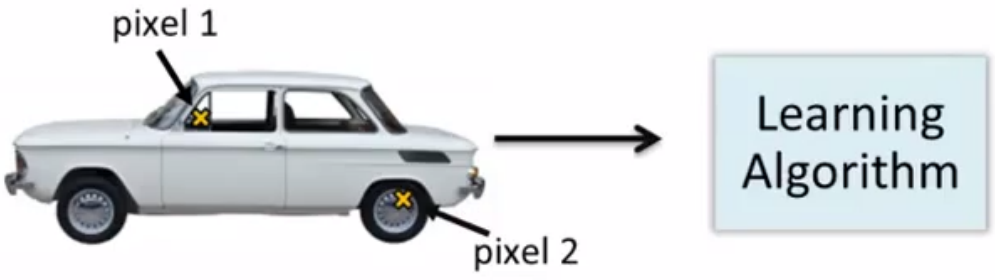
194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	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

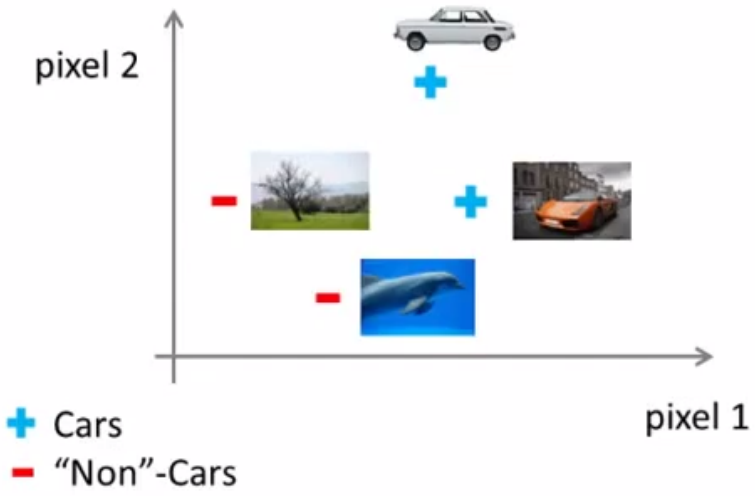
Visão computacional: Detecção de carros

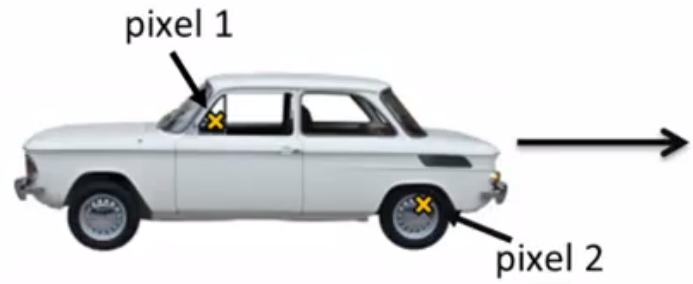


Teste:

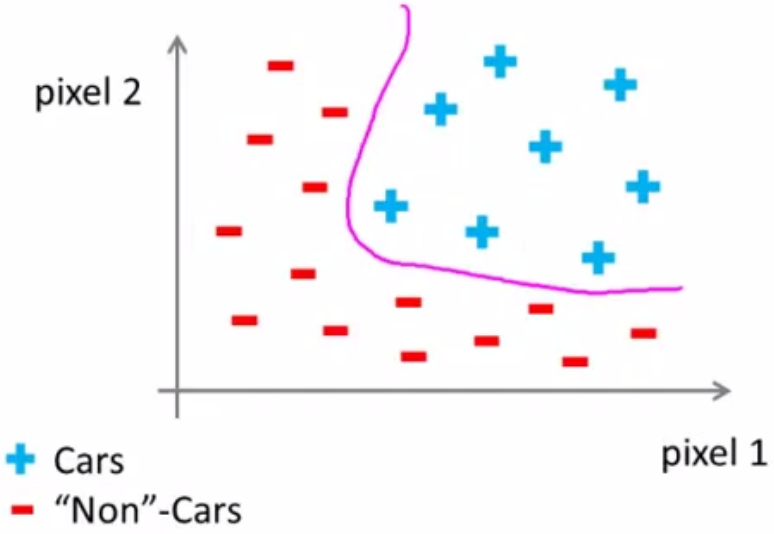
O que é isso?


pixel 1
pixel 2
Learning Algorithm


pixel 2
pixel 1
Cars
"Non"-Cars



Learning Algorithm



50 x 50 pixel images→ 2500 pixels

$n = 2500$ (7500 if RGB)

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Termos quadráticos ($x_i\, x_j$): 3 milhões de termos

Pergunta

Reconhecer carros a partir de imagens de 100x100 pixels (escala de cinza)

Usar regressão logística incluindo todos os termos quadráticos ($x_i\, x_j$)

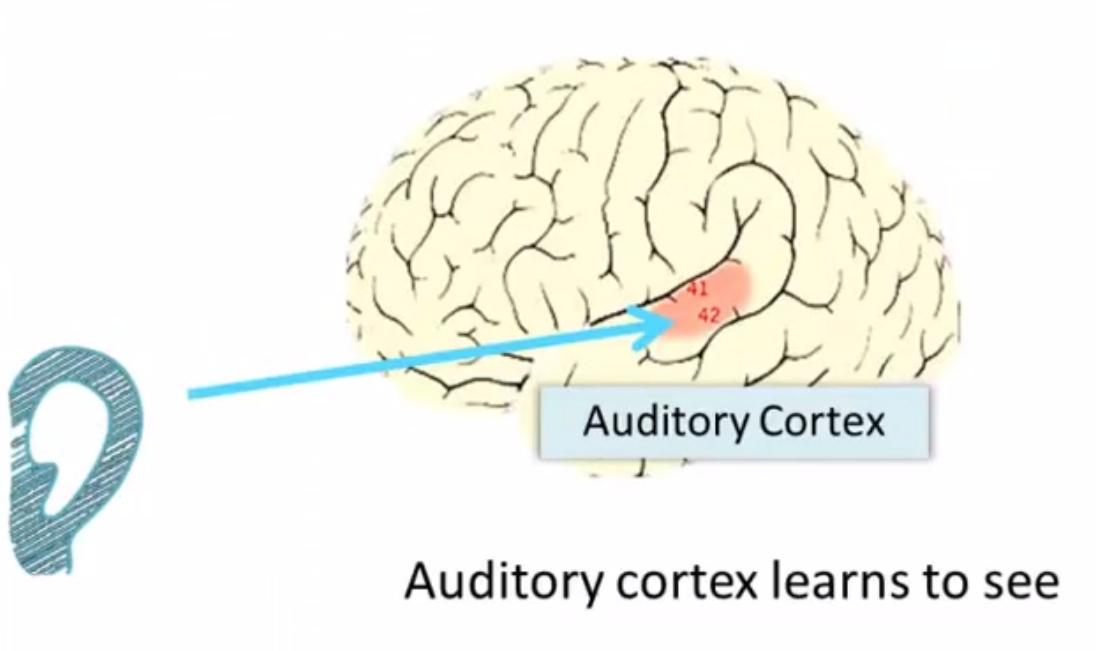
Há quantos termos destes aproximadamente?

Neurônios e o Cérebro

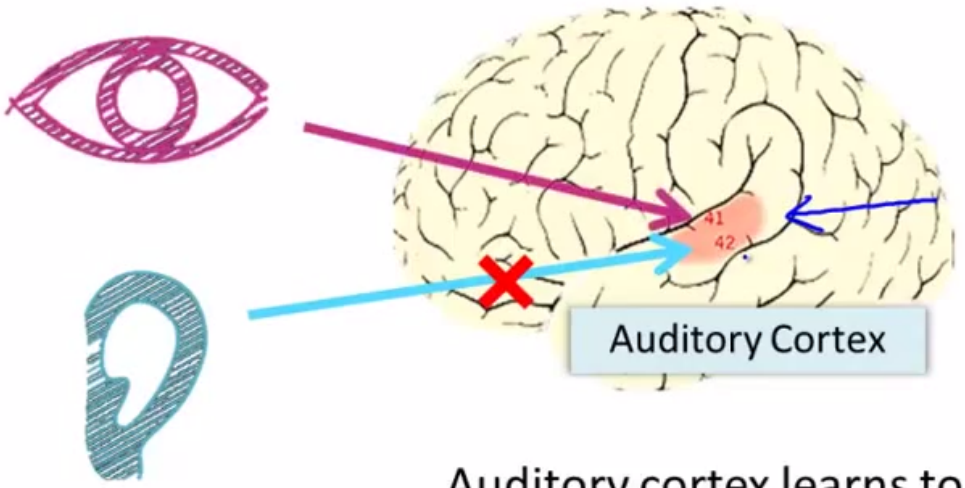
- Redes Neurais
  - Origens: Algoritmos que tentam imitar o cérebro
  - Usado amplamente nas décadas de 80 e 90. Popularidade diminui no final da década de 90.
  - Ressurgiu recentemente: técnica que alcança o estado da arte para muitas aplicações.

Neurociência computacional X Aprendizagem de Máquina

A hipótese “um algoritmo de aprendizagem”

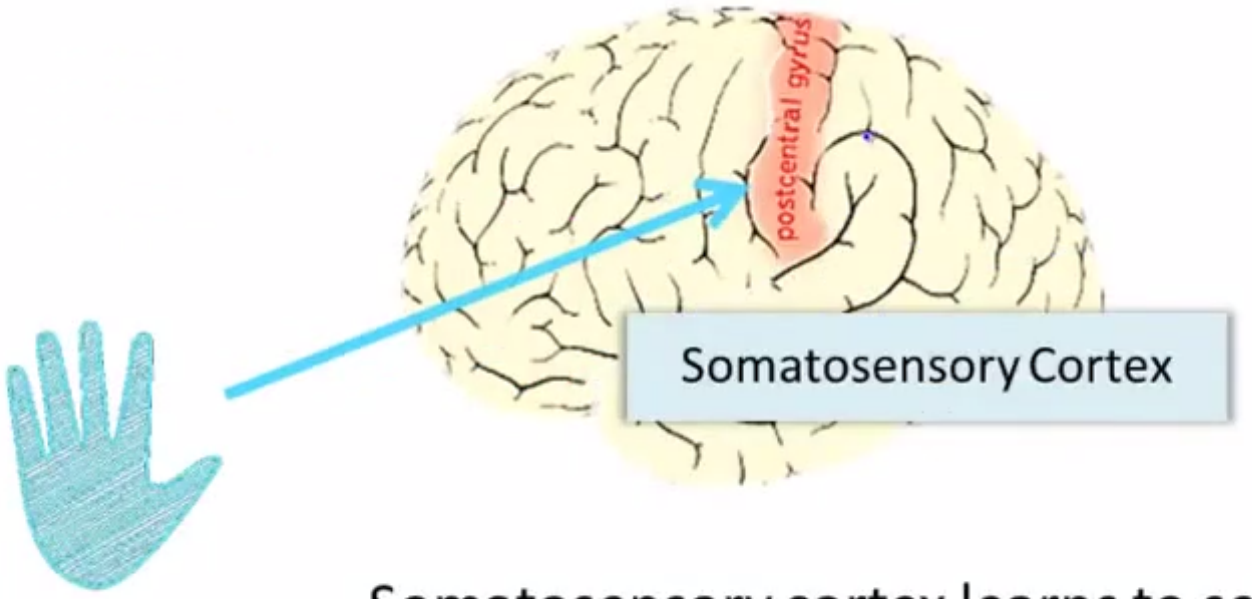


A hipótese “um algoritmo de aprendizagem”



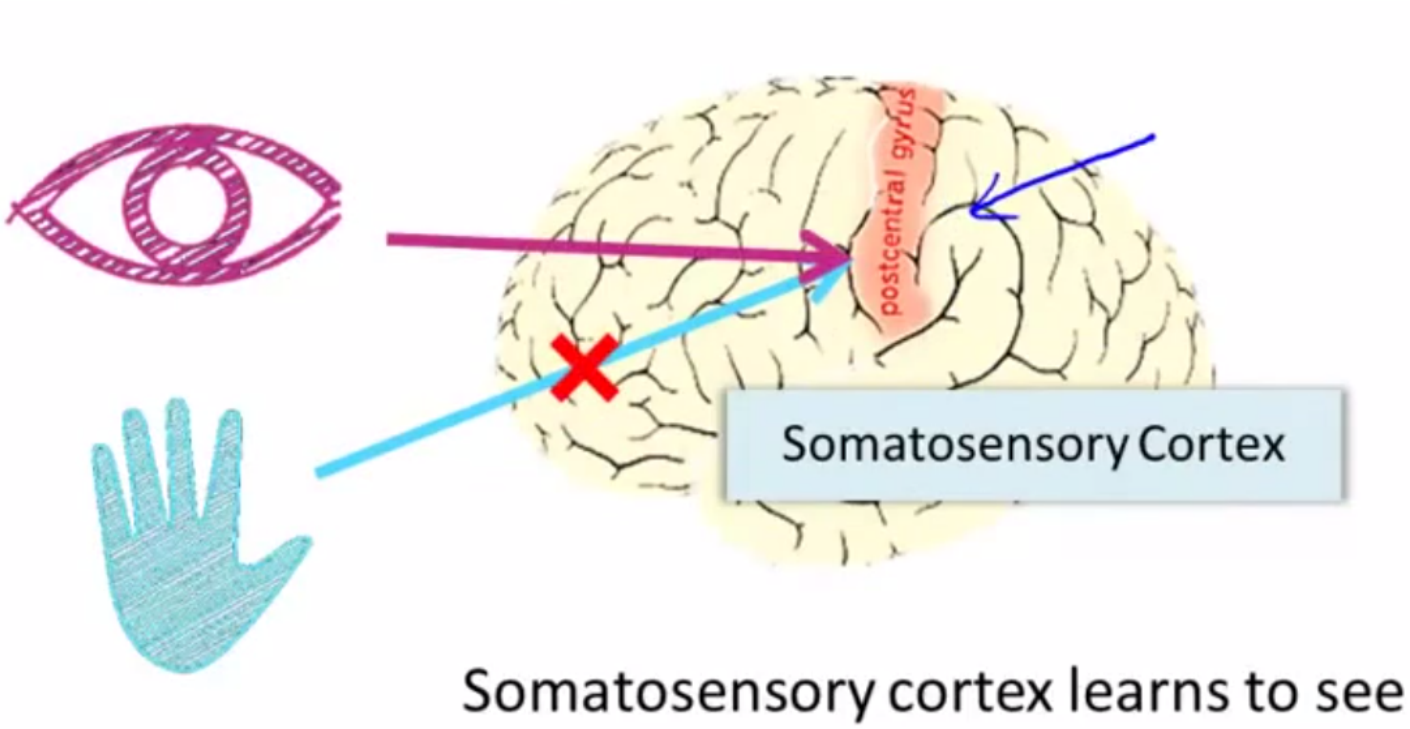
Auditory cortex learns to see

A hipótese “um algoritmo de aprendizagem”



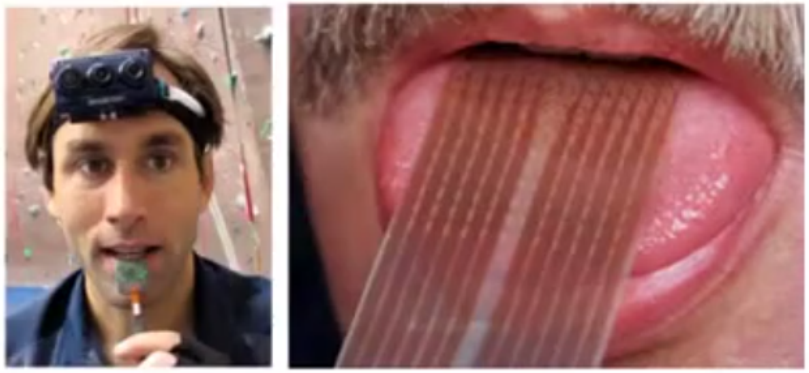
Somatosensory cortex learns to see

A hipótese “um algoritmo de aprendizagem”



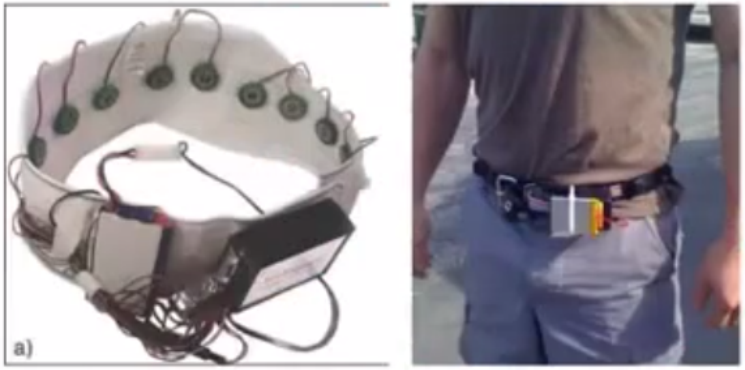
Somatosensory cortex learns to see

Representações sensoriais no cérebro



Seeing with your tongue

Human echolocation (sonar)



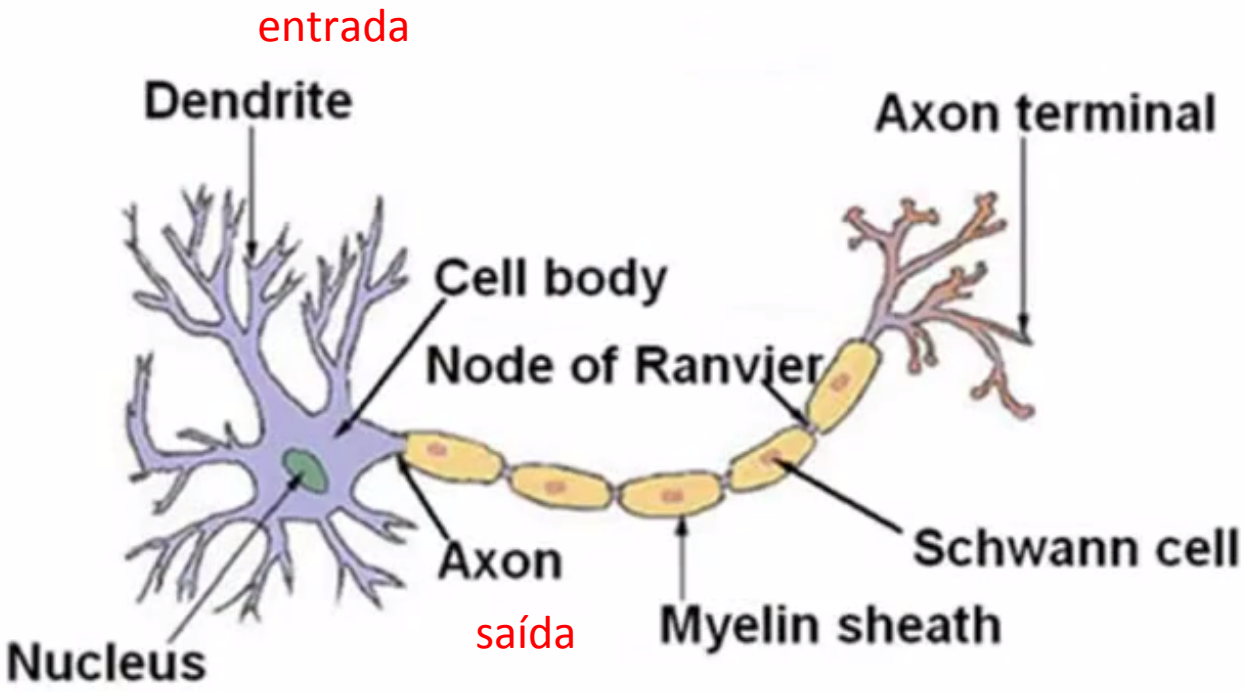
Haptic belt: Direction sense

Implanting a 3rd eye

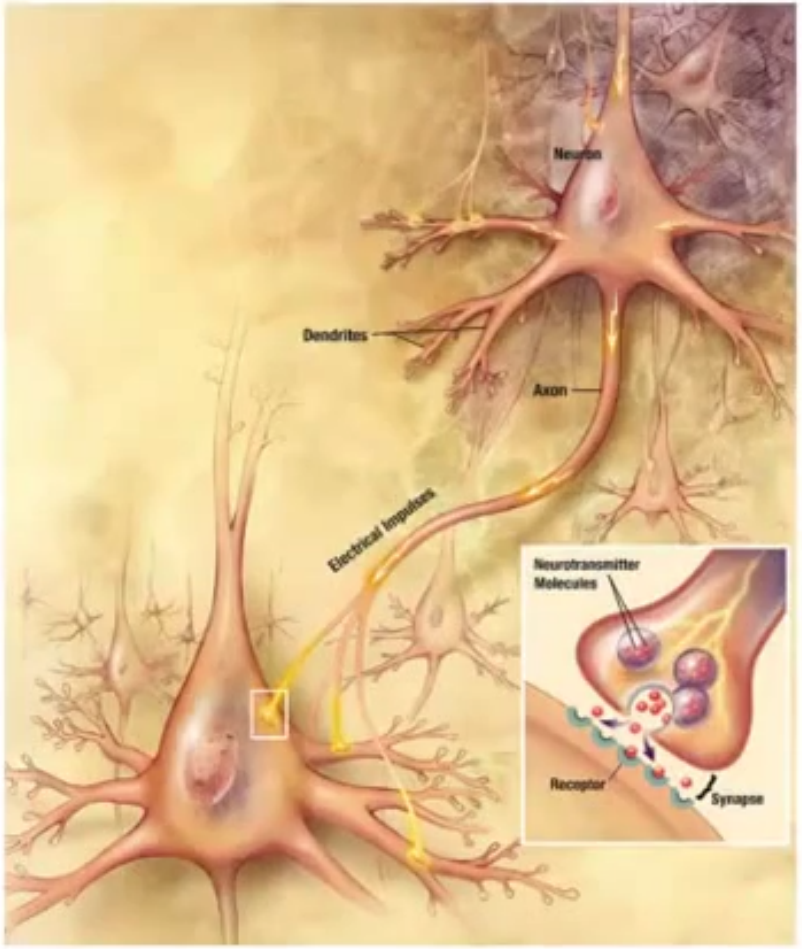
BrainPort; Welsh & Blasch, 1997; Nagel et al., 2005; Constantine-Paton & Law, 2009]

Redes Neurais:
Representação de Modelos

Neurônios no Cérebro



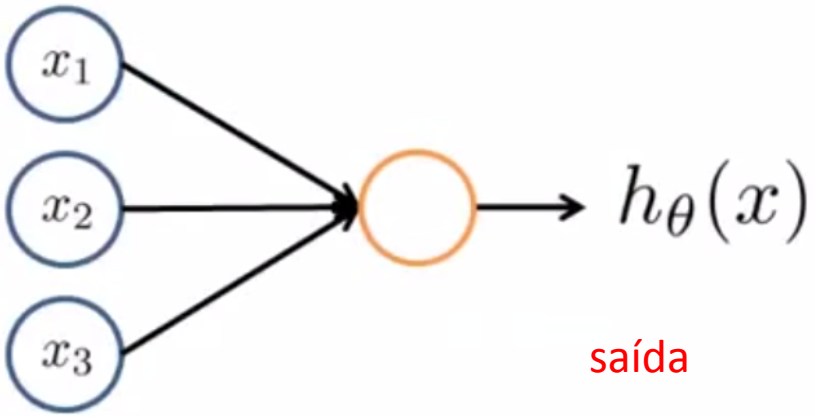
Neurônios no Cérebro



[Credit: US National Institutes of Health, National Institute on Aging]

Modelo de neurônio: unidade logística

entradas



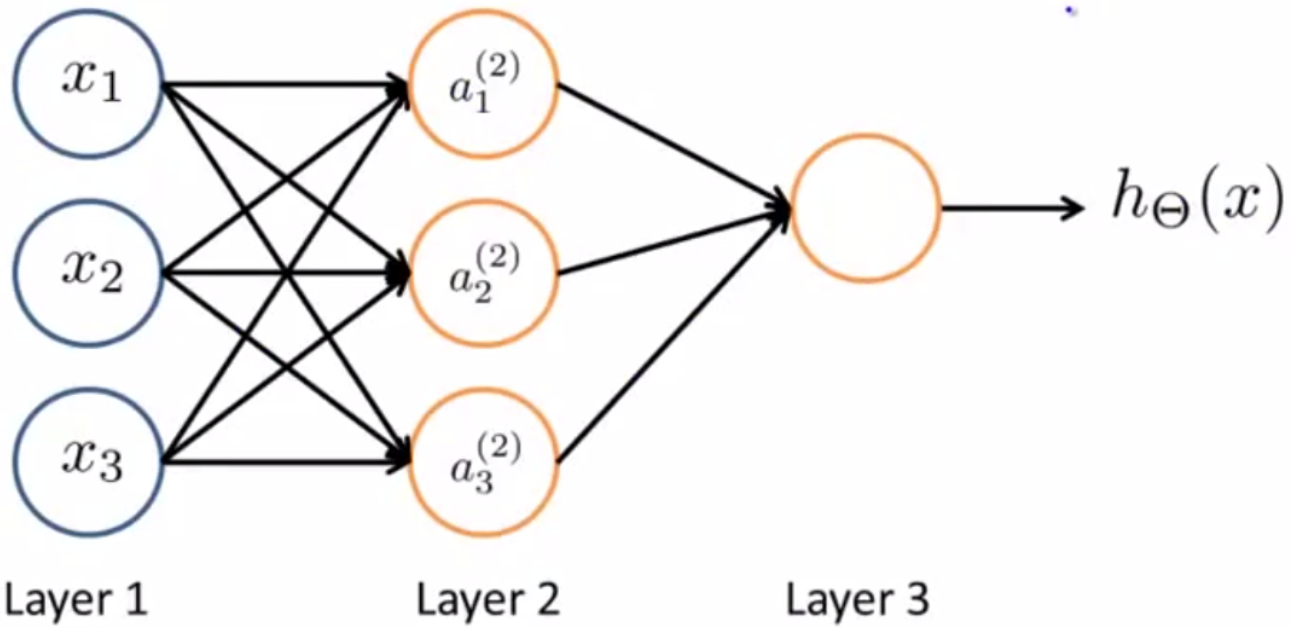
saída

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

"Pesos" ou
"parâmetros"

Função de ativação
logística (sigmóide)

Rede neural

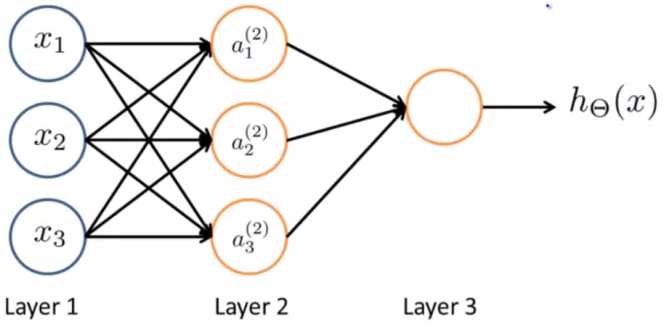


Camada de entrada

Camada oculta

Camada de saída

Rede neural

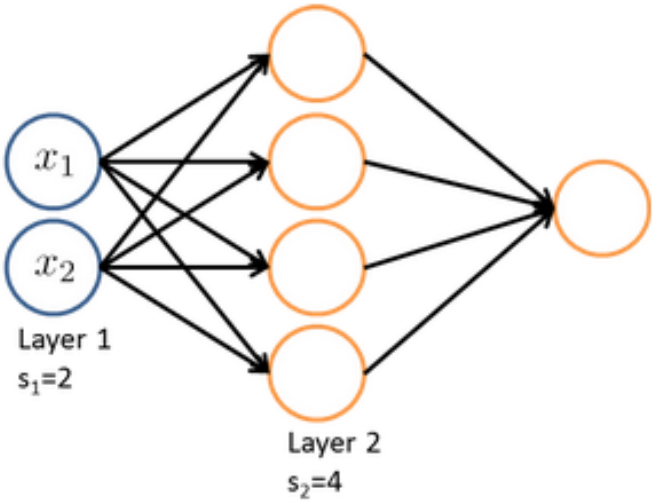


$a_i^{(j)} =$ Ativação da unidade *i* na camada *j*.

$\Theta^{(j)} =$ Matriz de pesos controlando o mapeamento da camada *j* para a camada *j+1*

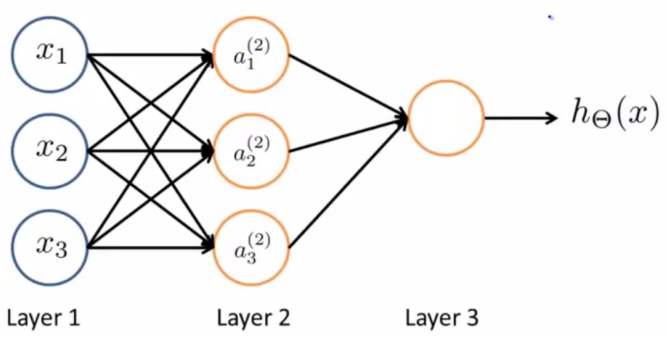
$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

Exercício:



Qual a dimensão de $\Theta^{(1)}$

Propagação para a frente: Implementação vetorial



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

$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

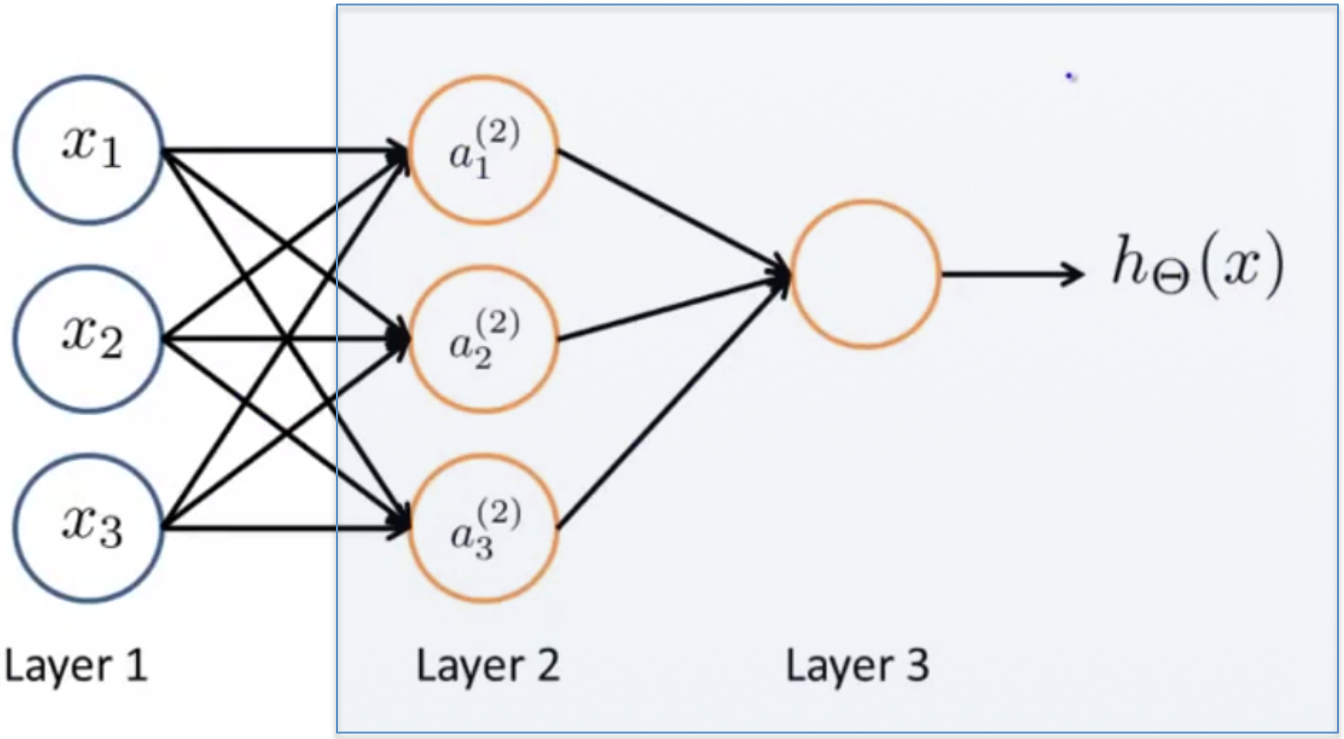
$$z^{(2)} = \Theta^{(1)} x$$
$$a^{(2)} = g(z^{(2)})$$

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

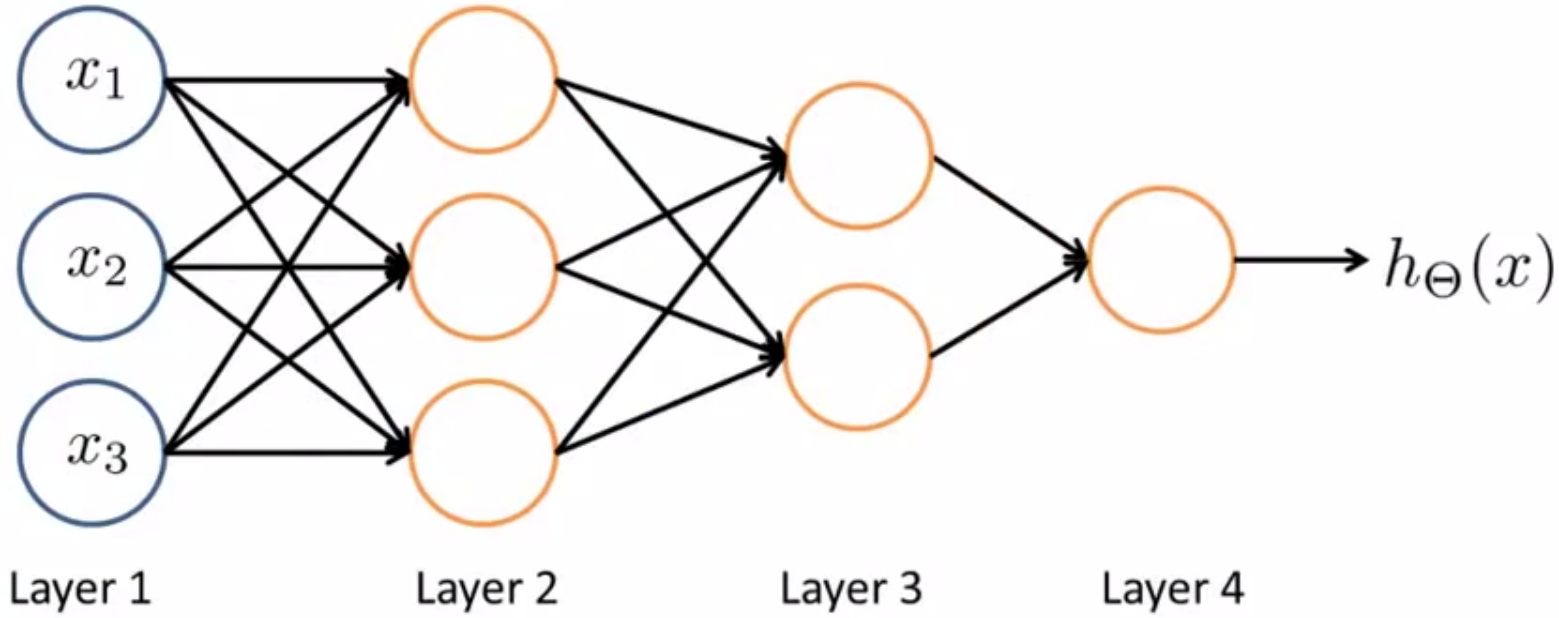
$$z^{(3)} = \Theta^{(2)} a^{(2)}$$
$$h_\Theta(x) = a^{(3)} = g(z^{(3)})$$

Rede neural:

aprendendo seus próprios termos não-lineares

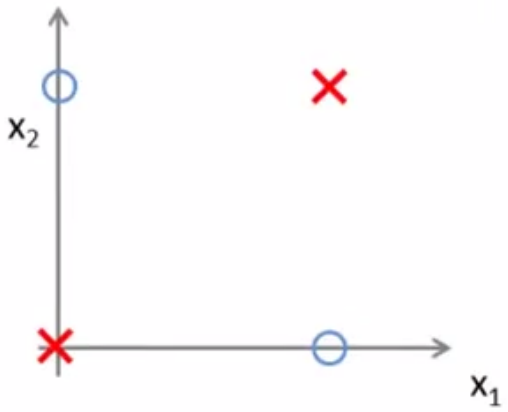


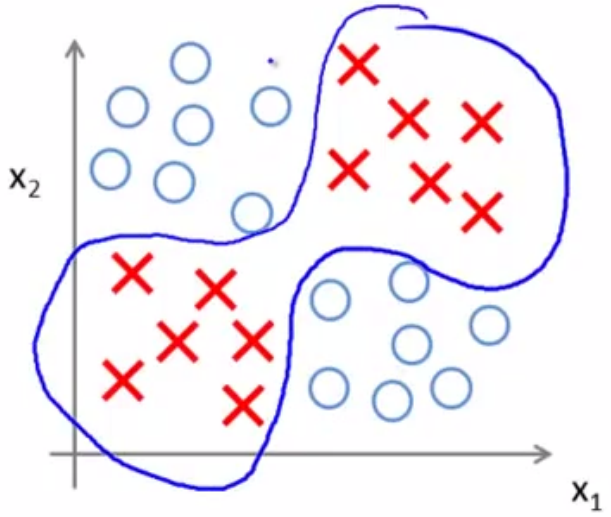
Outras arquiteturas:
mais camadas ocultas



Classificação não-linear: XOR/XNOR

x_1, x_2 are binary (0 or 1).





$y = x_1$ XOR x_2

x_1 XNOR x_2

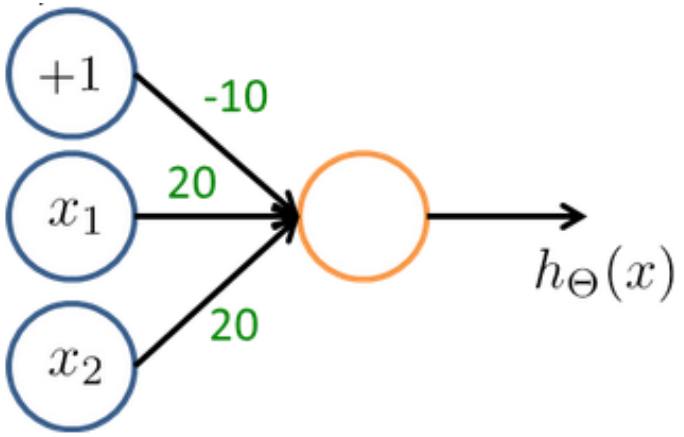
NOT (x_1 XOR x_2)

Exemplo Simples: E (*And*)

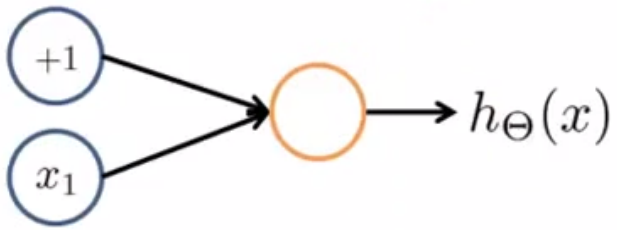
$$x_1, x_2 \in \{0, 1\}$$
$$y = x_1 \text{ AND } x_2$$

Exercício:

Qual função a rede neural abaixo modela?



Negação

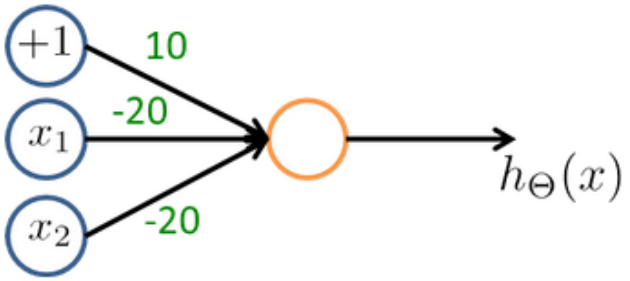


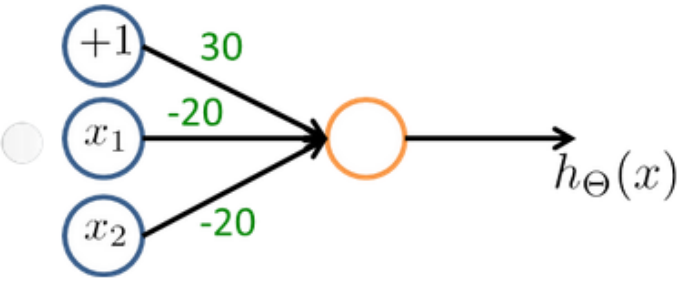
x_1	$h_\Theta(x)$
0	
1	

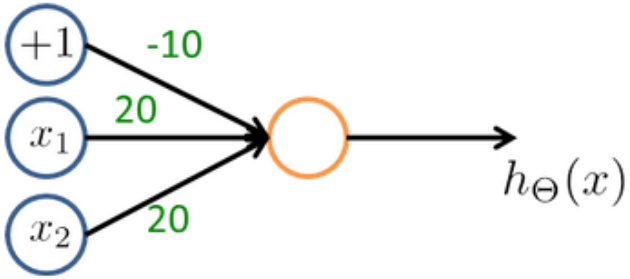
Exercício

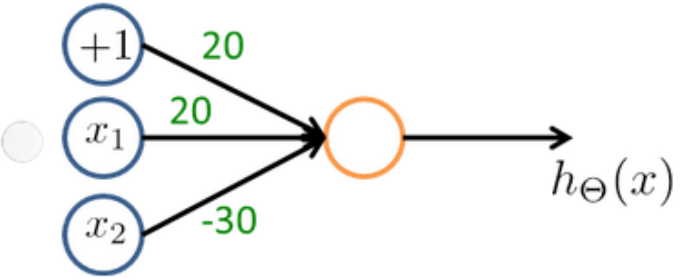
- Criar uma rede simples que modela a função:

$$(\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$$

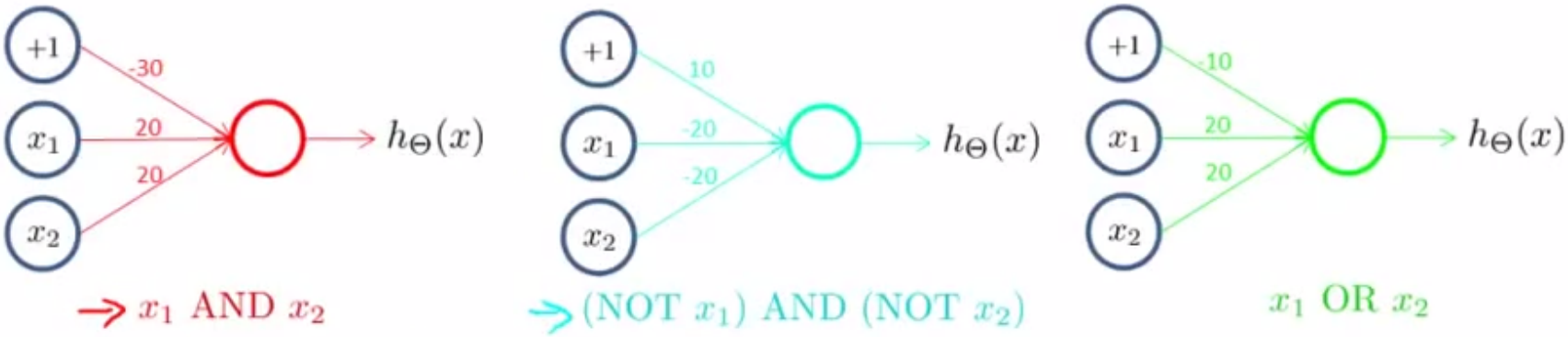




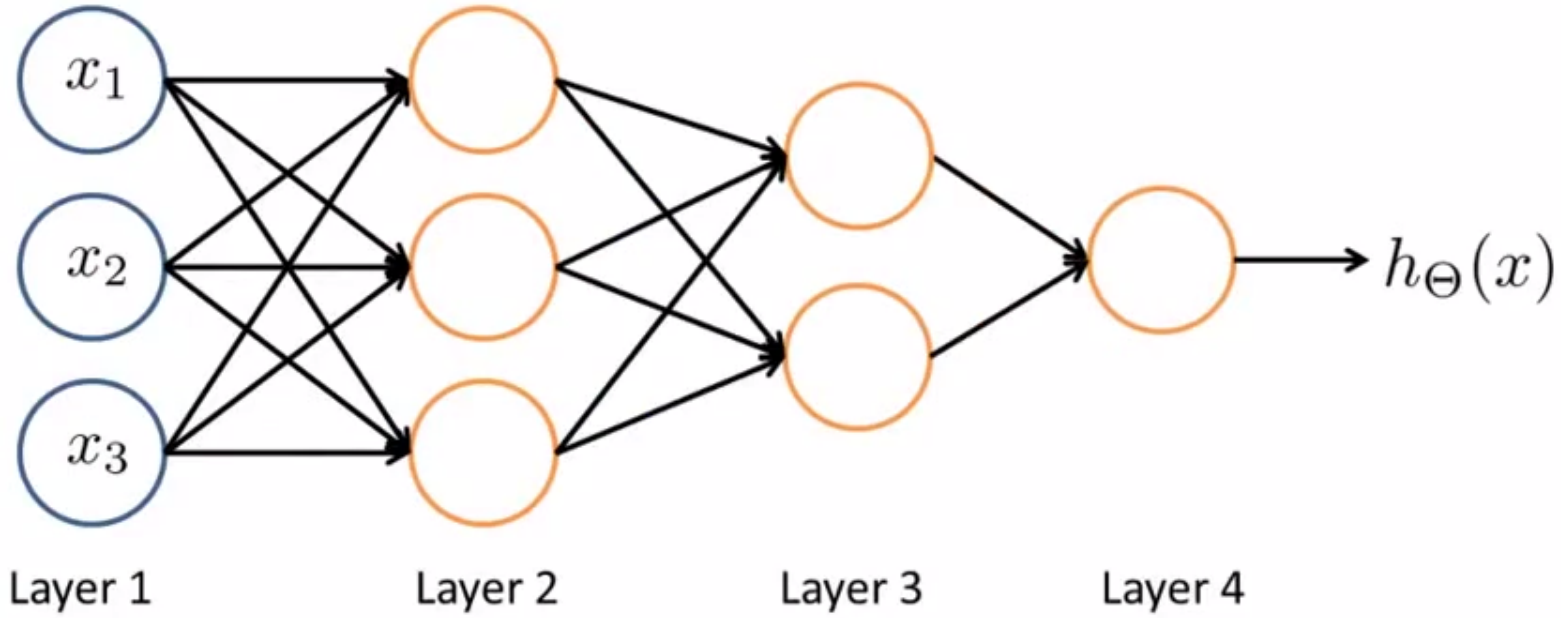




Juntando: x_1 XNOR x_2



Múltiplas camadas:
maior complexidade



Video:

Reconhecimento de dígito: Jann LeCun

Classificação com múltiplas classes

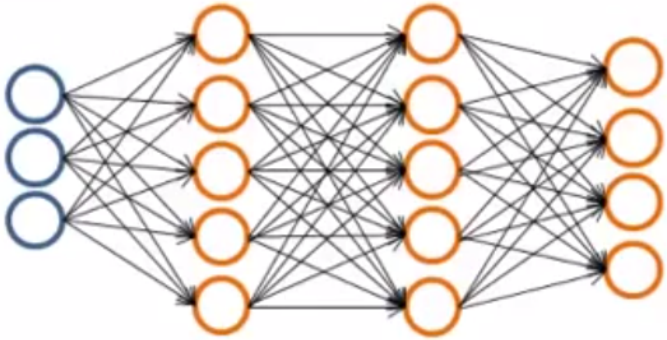
Múltiplas Unidades de Saída: Um-vs-Todos

Pedestrian

Car

Motorcycle

Truck

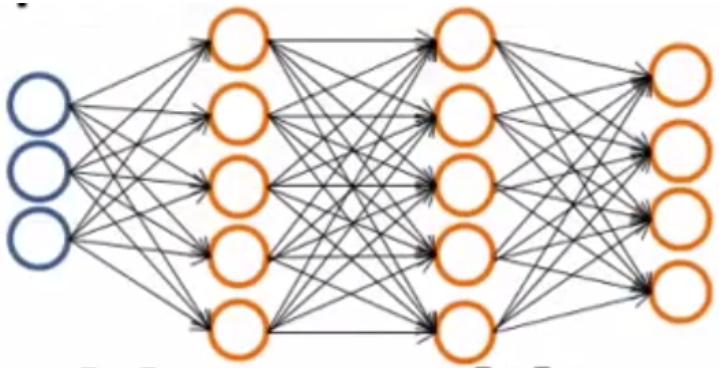


$h_\Theta(x) \in \mathbb{R}^4$

Want $h_\Theta(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, etc.

when pedestrian when car when motorcycle

Múltiplas Unidades de Saída: Um-vs-Todos



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

Want $h_\Theta(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, etc.

when pedestrian when car when motorcycle

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

$y^{(i)}$ one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

pedestrian car motorcycle truck

Exercício

- Suponha que você tenha um problema de classificação com 10 classes. A rede neural tem 3 camadas, sendo que a camada oculta (camada 2) tem 5 unidades. Usando o método Um-vs-Todos, quantos elementos tem $\Theta^{(2)}$?