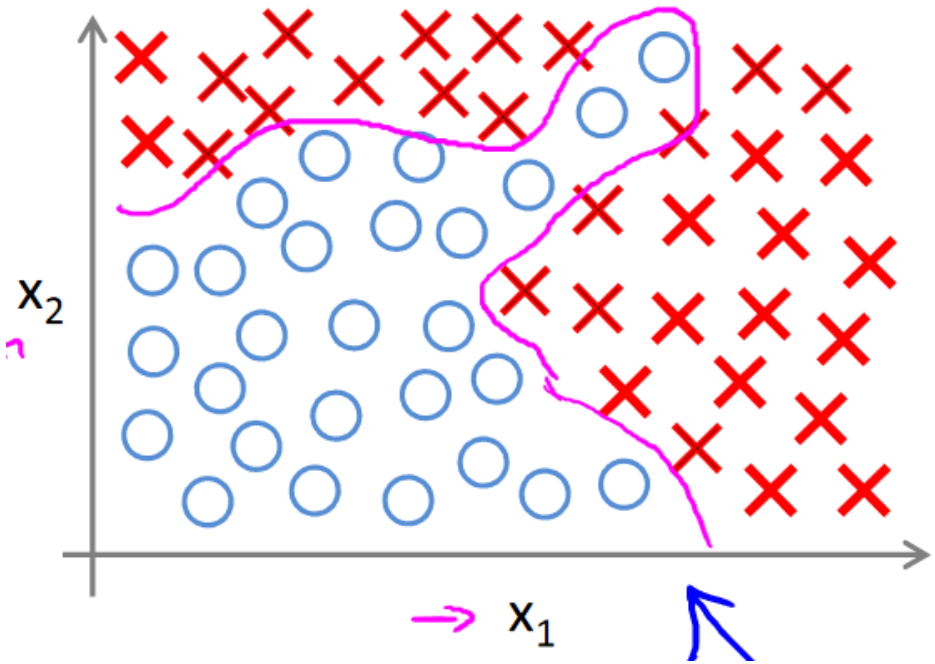
Aula 7 - Redes Neurais I (Conceito)

João Florindo

Instituto de Matemática, Estatística e Computação Científica
Universidade Estadual de Campinas - Brasil
florindo@unicamp.br

Outline

Imagine uma fronteira de decisão muito complexa:



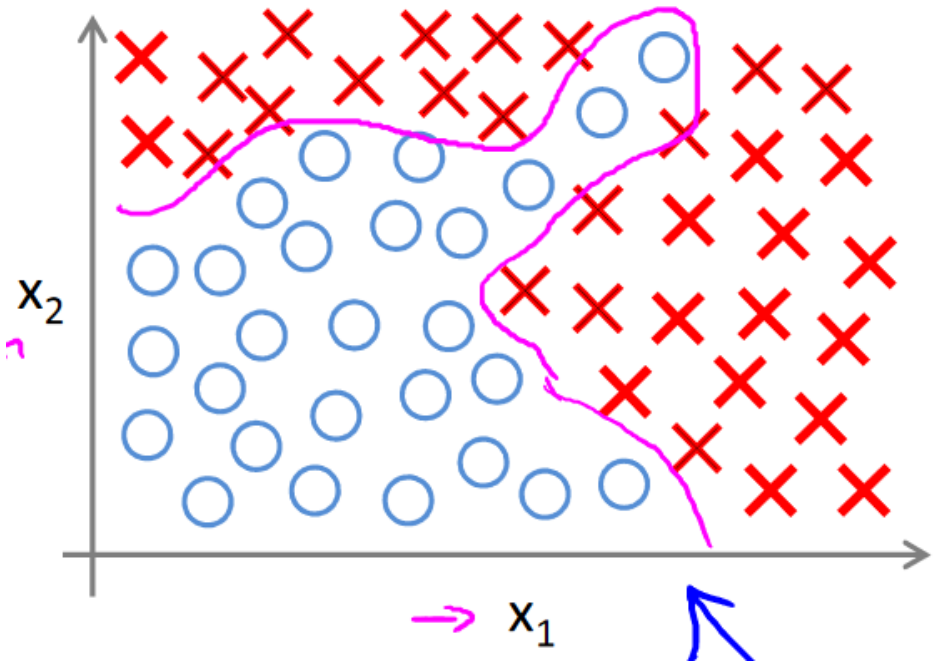
Vimos soluções como

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \cdots).$$

Mas:

- Propenso a *overfitting*!

Imagine uma fronteira de decisão muito complexa:



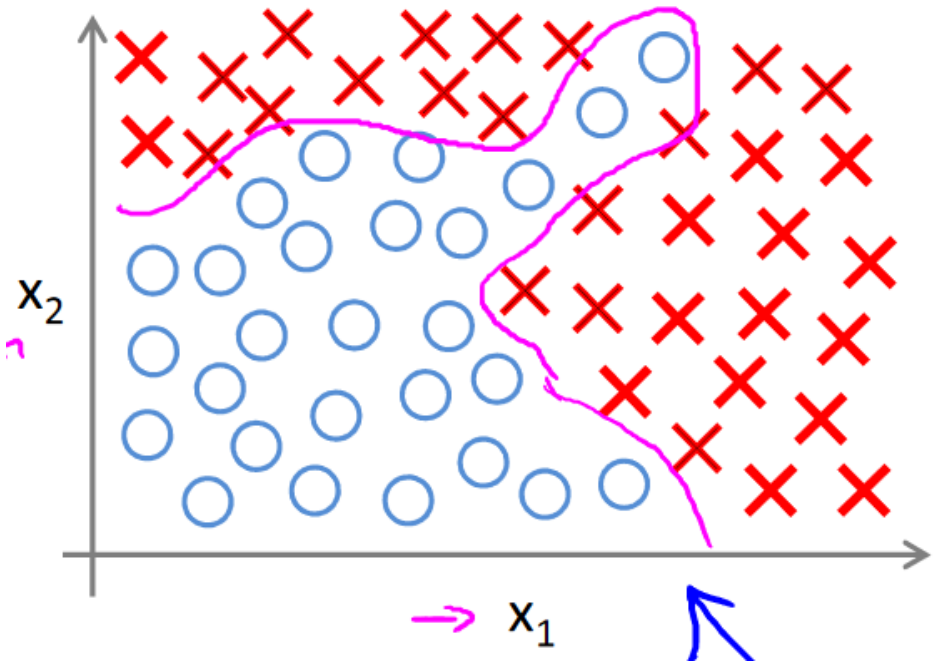
Vimos soluções como

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \cdots).$$

Mas:

- Propenso a *overfitting*!

Imagine uma fronteira de decisão muito complexa:



Vimos soluções como

$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \cdots).$$

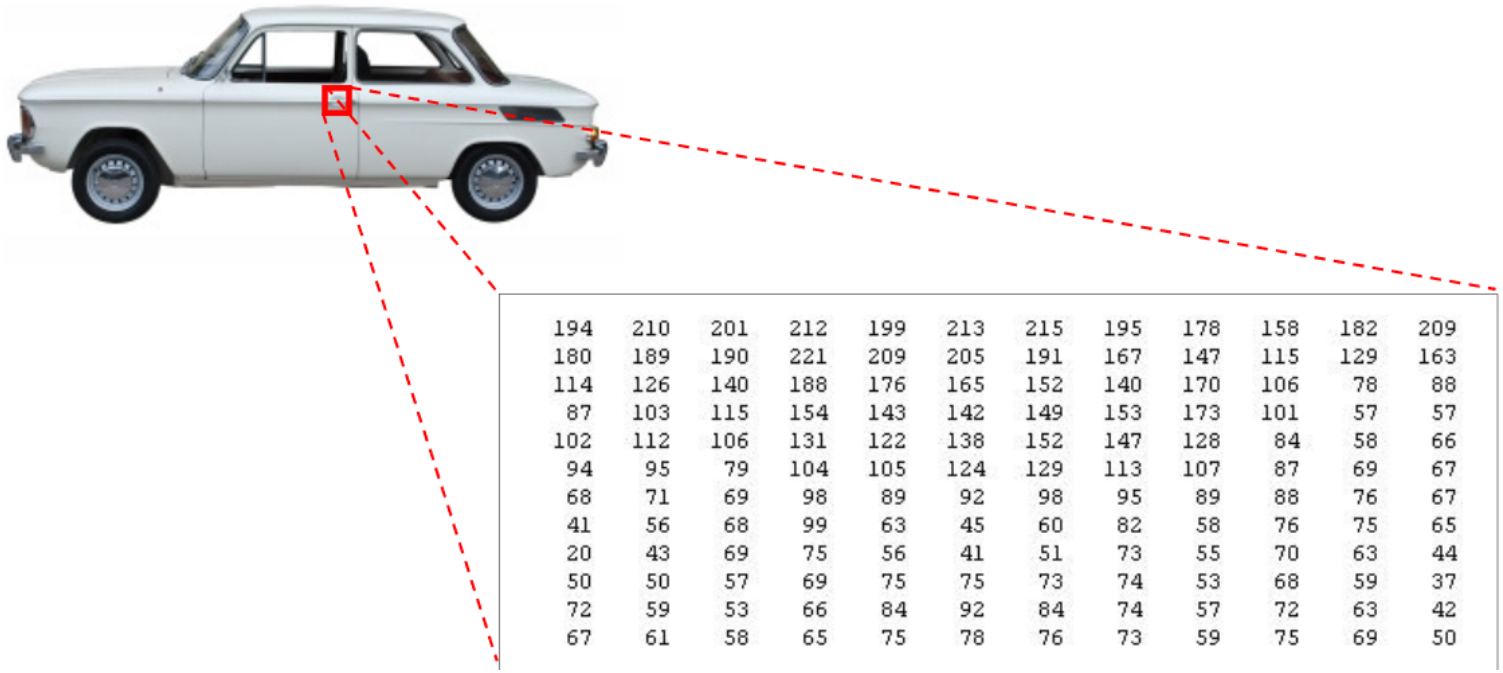
Mas:

- Propenso a *overfitting*!

- E se temos 100 atributos?
  - $\approx$ 5000 termos quadráticos!
  - $\approx$ 17000 termos cúbicos!
- Em visão computacional, cada pixel é um atributo
  - Uma imagem RGB 50×50 geraria 3 MILHÕES só de termos quadráticos na regressão logística não linear!

- *Overfitting* ALTÍSSIMO!

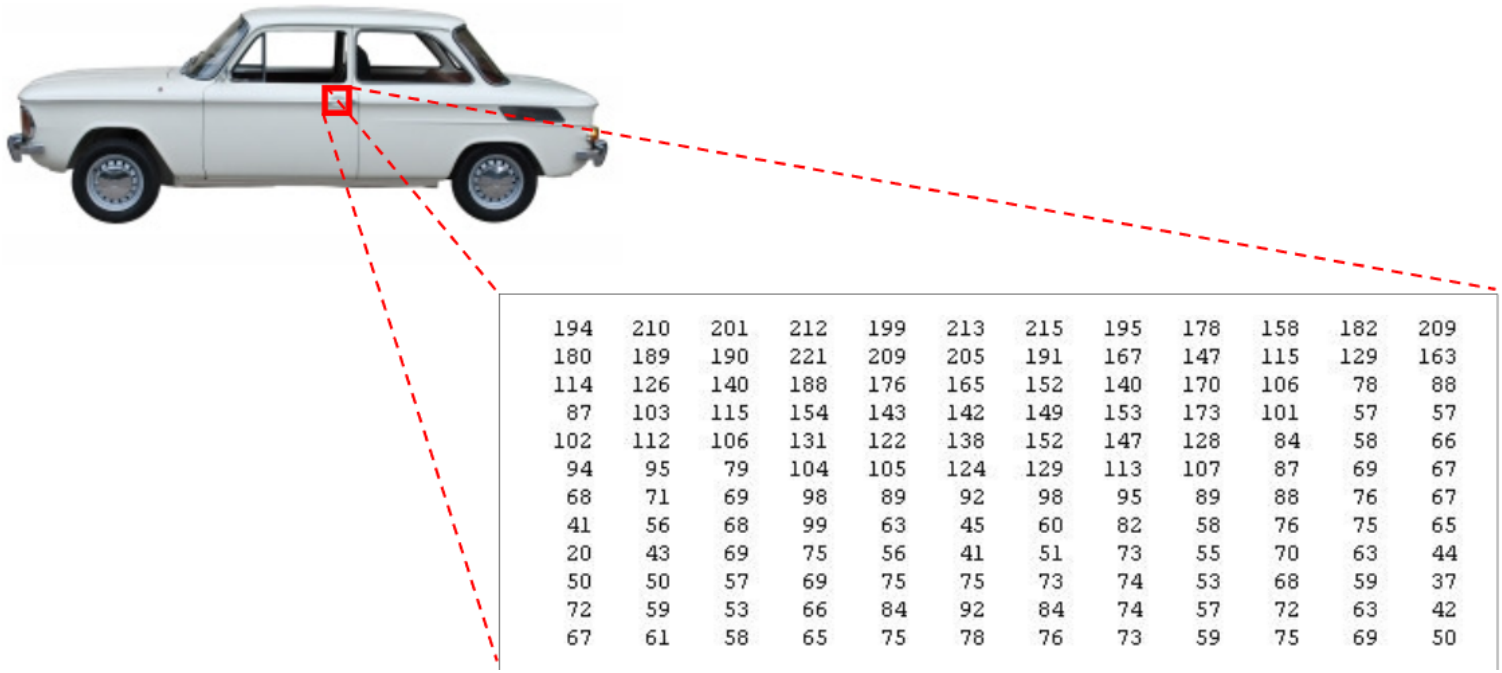
- E se temos 100 atributos?
  - $\approx$ 5000 termos quadráticos!
  - $\approx$ 17000 termos cúbicos!
- Em visão computacional, cada pixel é um atributo
  - Uma imagem RGB 50×50 geraria 3 MILHÕES só de termos quadráticos na regressão logística não linear!



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

- *Overfitting* ALTÍSSIMO!

- E se temos 100 atributos?
  - $\approx$ 5000 termos quadráticos!
  - $\approx$ 17000 termos cúbicos!
- Em visão computacional, cada pixel é um atributo
  - Uma imagem RGB 50×50 geraria 3 MILHÕES só de termos quadráticos na regressão logística não linear!



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

- *Overfitting* ALTÍSSIMO!

- Precisamos de algo mais eficiente e "inteligente"!

- Mas, afinal, como o cérebro aprende?
- Seu "algoritmo" é único?
  - Conjectura-se que SIM!
  - *Rewiring*: Qualquer região do cérebro pode aprender qualquer tarefa.
  - Diversas aplicações práticas deste conceito.

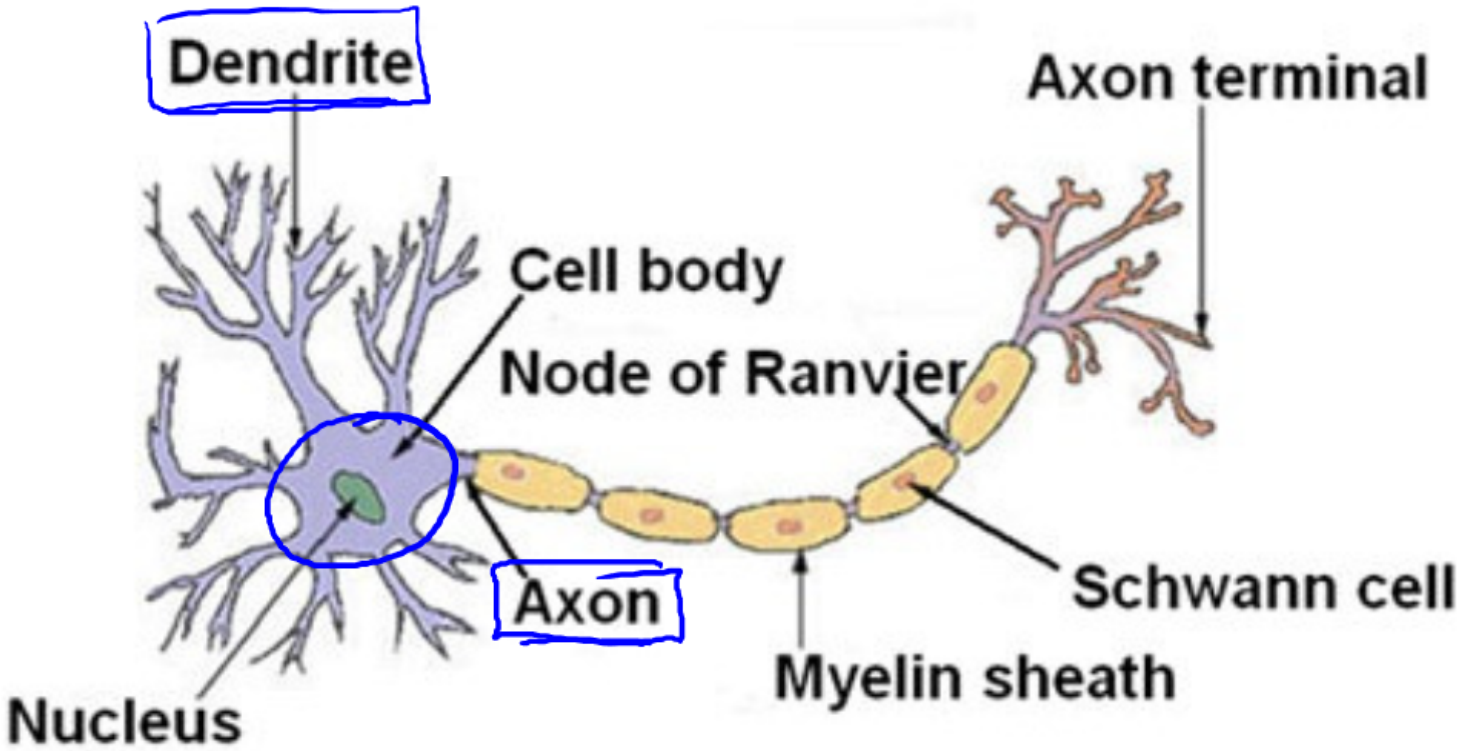
- Mas, afinal, como o cérebro aprende?
- Seu "algoritmo" é único?
  - Conjectura-se que SIM!
  - *Rewiring*: Qualquer região do cérebro pode aprender qualquer tarefa.
  - Diversas aplicações práticas deste conceito.

- Mas, afinal, como o cérebro aprende?
- Seu "algoritmo" é único?
  - Conjectura-se que SIM!
  - *Rewiring*: Qualquer região do cérebro pode aprender qualquer tarefa.
  - Diversas aplicações práticas deste conceito.

- Mas, afinal, como o cérebro aprende?
- Seu "algoritmo" é único?
  - Conjectura-se que SIM!
  - *Rewiring*: Qualquer região do cérebro pode aprender qualquer tarefa.
  - Diversas aplicações práticas deste conceito.

- Mas, afinal, como o cérebro aprende?
- Seu "algoritmo" é único?
  - Conjectura-se que SIM!
  - *Rewiring*: Qualquer região do cérebro pode aprender qualquer tarefa.
  - Diversas aplicações práticas deste conceito.

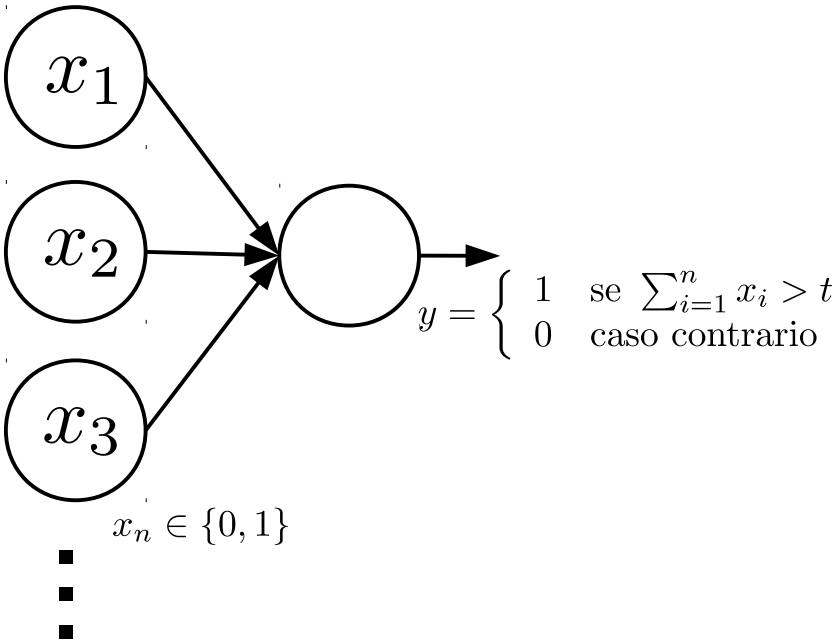
"Algoritmo" do Cérebro



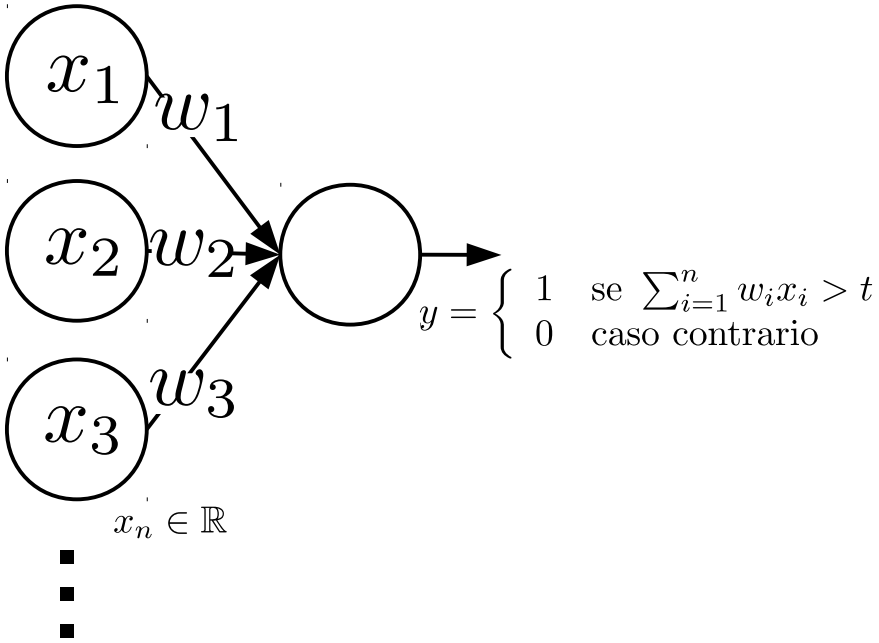
- Processamento em cascata e organização em camadas.

Neurônio Artificial

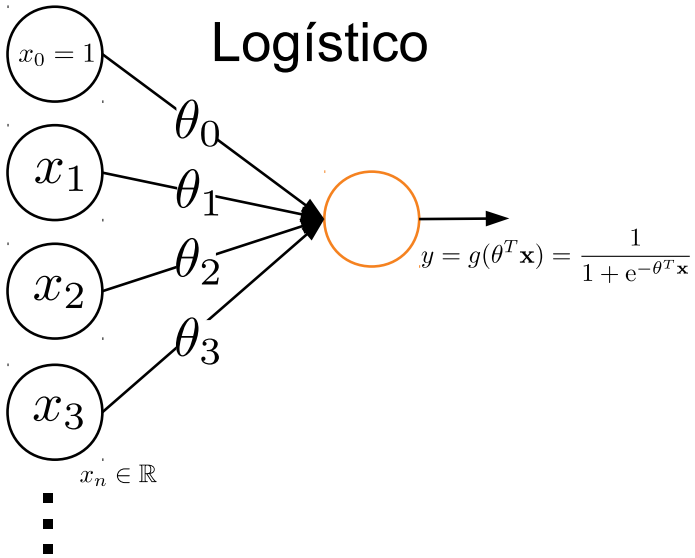
McCulloch-Pitts



Perceptron



Neurônio Artificial

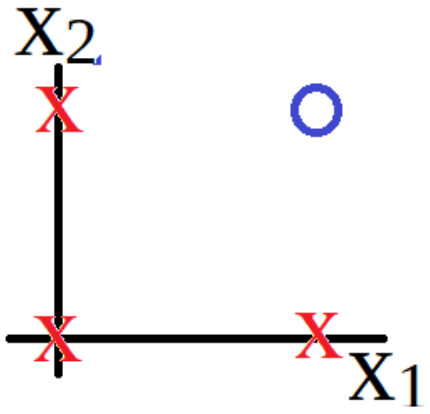


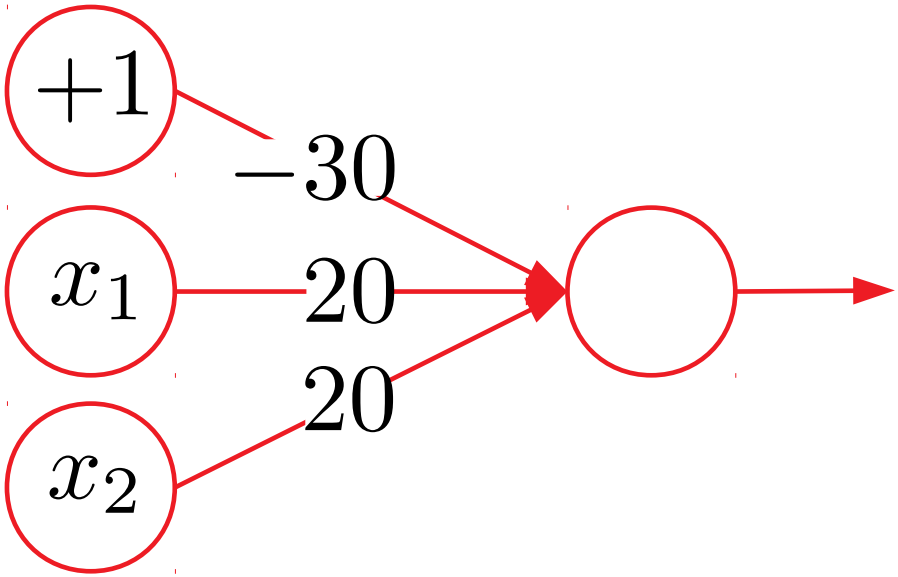
Nomenclatura

- Cada ““neurônio” faz regressão logística e é chamado de **unidade de ativação** (em laranja).
- Cada parâmetro θ_j é chamado agora de **peso**. θ_0 é o **bias**.
- A função sigmoide $g(z)$ é chamada de **função de ativação**.

Outline

Exemplo 1 - AND

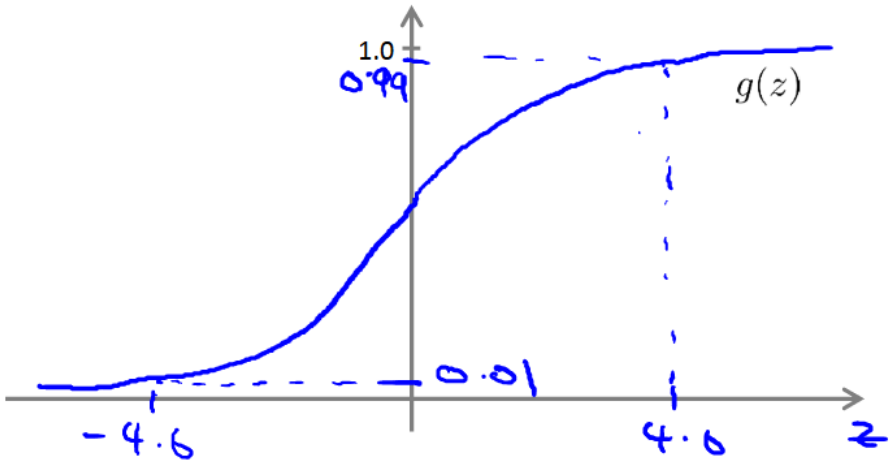




$$h_\Theta(x) = g(-30 + 20x_1 + 20x_2)$$

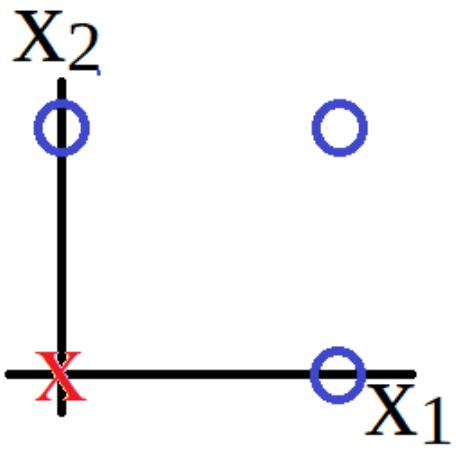
Exemplo 1 - AND

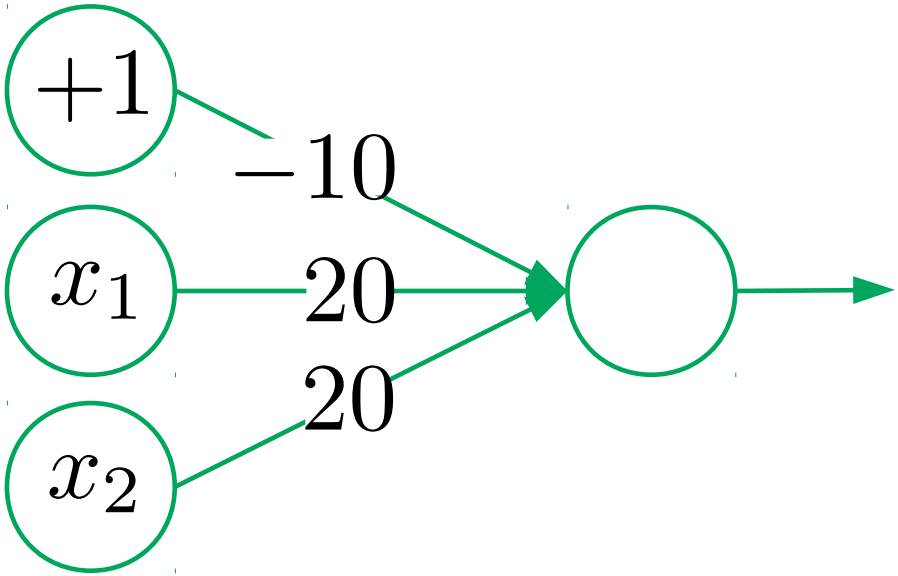
$g(z) \approx 0$ se $z < -4$ e $g(z) \approx 1$ se $z > 4$:



x_1	x_2	$h_\Theta(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

Exemplo 2 - OR

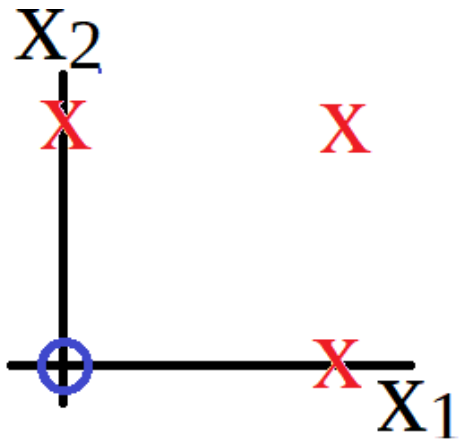


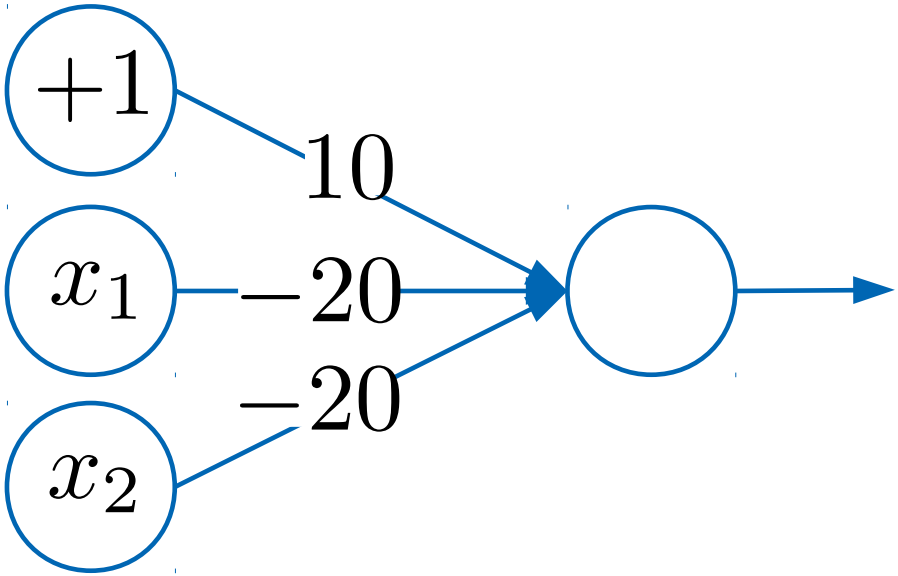


$$h_\Theta(x) = g(-10 + 20x_1 + 20x_2)$$

x_1	x_2	$h_\Theta(x)$
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	$g(10) \approx 1$
1	1	$g(10) \approx 1$

Exemplo 3 - (NOT x_1) AND (NOT x_2)





$$h_\Theta(x) = g(10 - 20x_1 - 20x_2)$$

x_1	x_2	$h_\Theta(x)$
0	0	$g(10) \approx 1$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(-30) \approx 0$

Exemplo 4 - XNOR

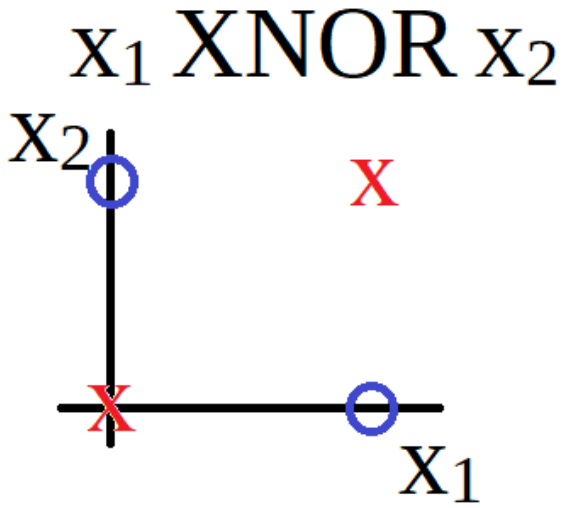
- O que fizemos até agora?
  - Regressão Logística (fronteira de decisão linear)!

MAS:

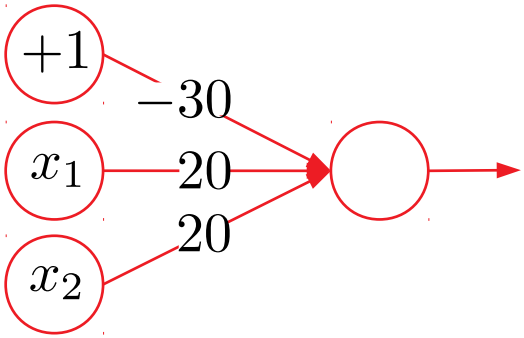
Exemplo 4 - XNOR

- O que fizemos até agora?
  - Regressão Logística (fronteira de decisão linear)!

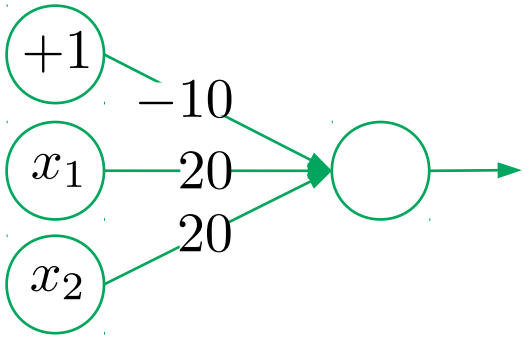
MAS:



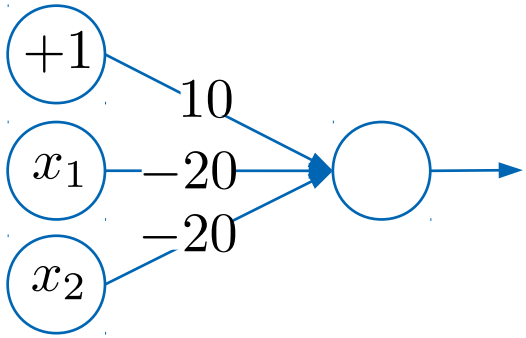
Exemplo 4 - XNOR



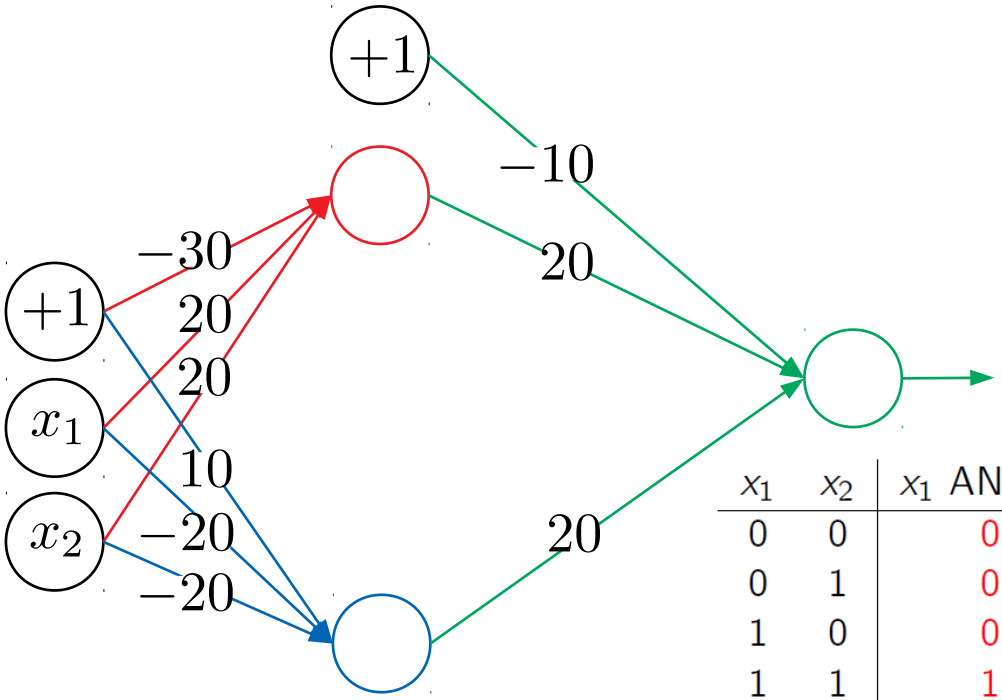
x_1 AND x_2



x_1 OR x_2



(NOT x_1) AND (NOT x_2)



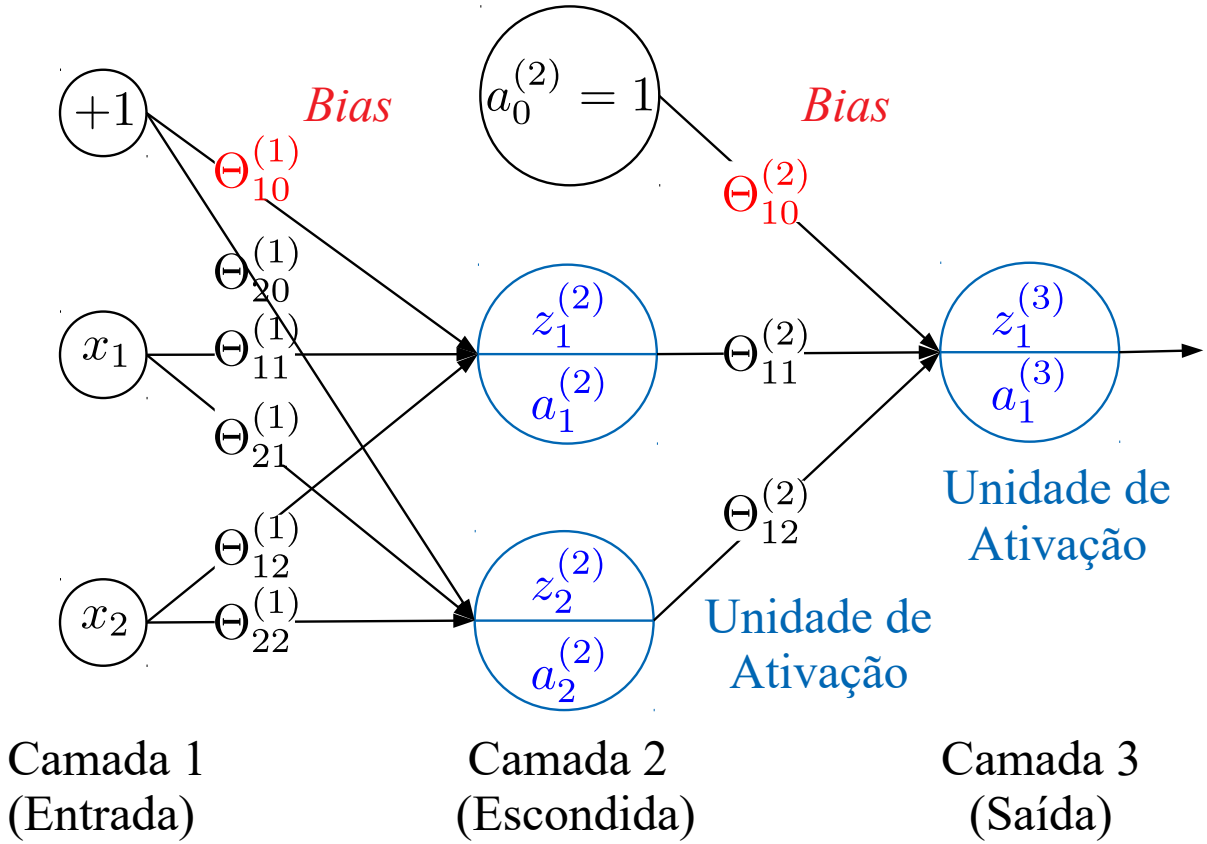
x_1	x_2	x_1 AND x_2	(NOT x_1) AND (NOT x_2)	x_1 XNOR x_2
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Outline

Rede Neural

Definição

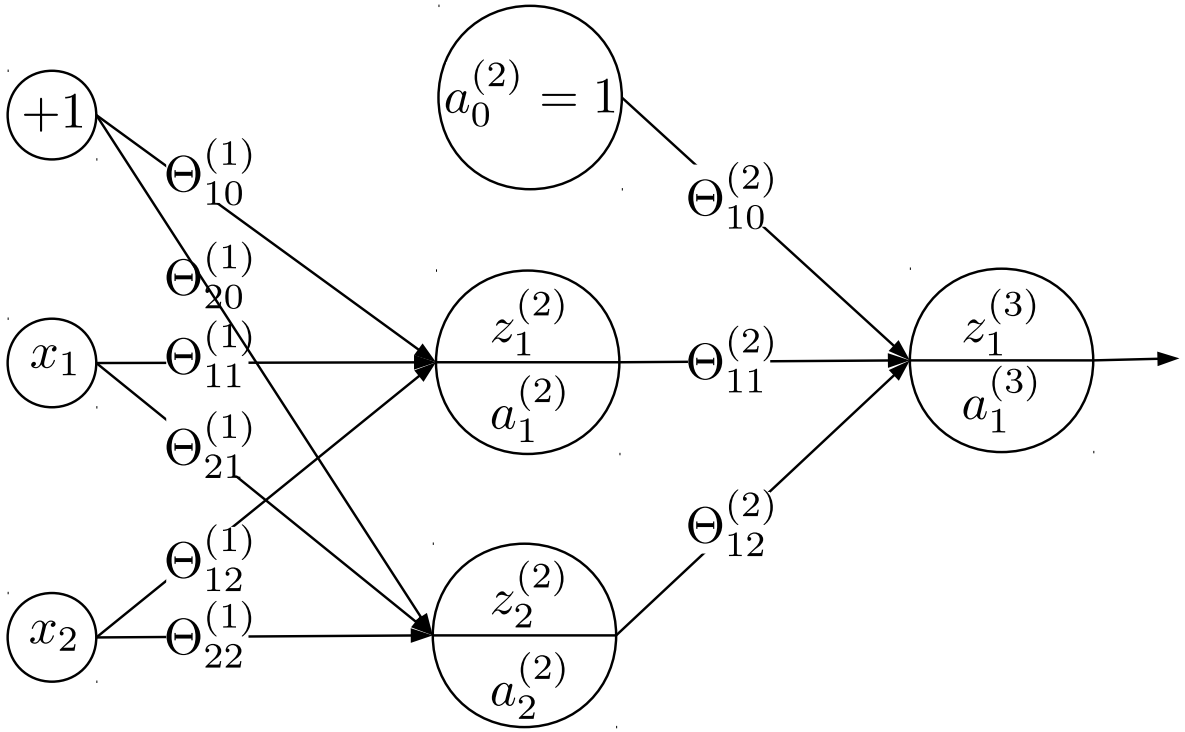
Uma **rede neural** é um conjunto destes "neurônios" conectados entre si e originizados em camadas.



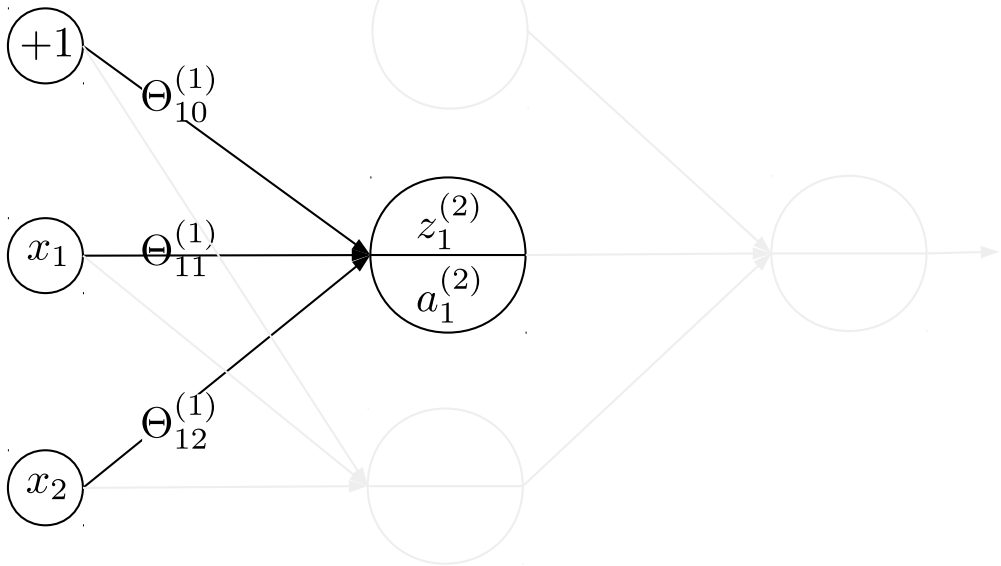
Notação

- $z_i^{(j)}$: soma ponderada das entradas na unidade i na camada j.
- $a_i^{(j)} = g(z_i^{(j)})$: ativação da unidade i na camada j (unidade 0 é o *bias*).
- $\Theta^{(j)}$: matriz de pesos sobre a saída da camada j, saída esta que é uma das entradas da camada $j+1$.
- $\Theta_{ab}^{(j)}$: parâmetro que multiplica a saída da unidade b da camada j para compor a entrada da unidade a na camada $j+1$.

Outline



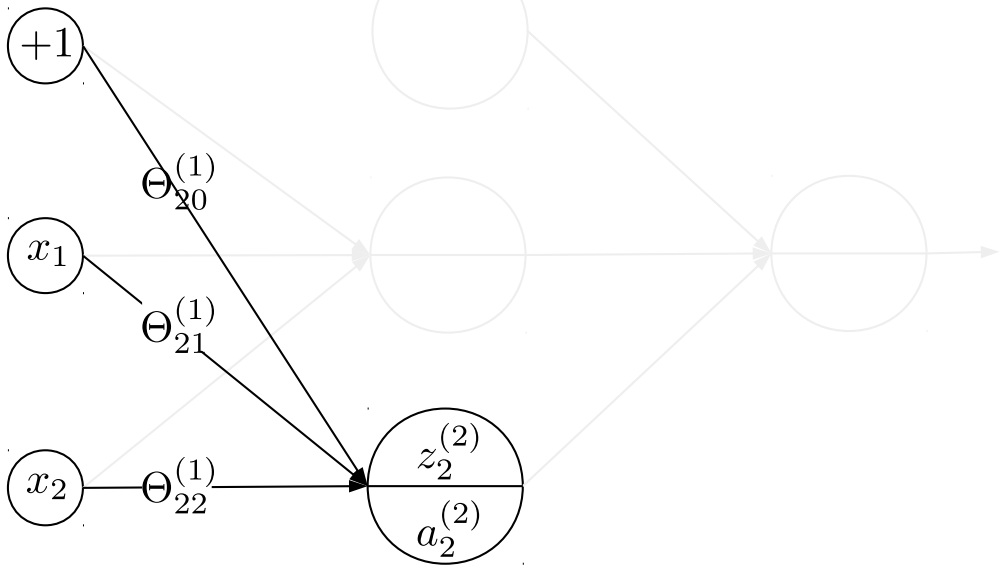
$x_1 = 0.5$	$x_2 = 1.0$				
$\Theta_{10}^{(1)} = -0.1$	$\Theta_{20}^{(1)} = 0.2$	$\Theta_{11}^{(1)} = 1.3$	$\Theta_{21}^{(1)} = -0.5$	$\Theta_{12}^{(1)} = 0.6$	$\Theta_{22}^{(1)} = 0$
$\Theta_{10}^{(2)} = 2.1$	$\Theta_{11}^{(2)} = -1.5$	$\Theta_{12}^{(2)} = -0.3$			



$x_1 = 0.5$	$x_2 = 1.0$	
$\Theta_{10}^{(1)} = -0.1$	$\Theta_{11}^{(1)} = 1.3$	$\Theta_{12}^{(1)} = 0.6$

$$
\begin{aligned}
z_1^{(2)} &= \Theta_{10}^{(1)} \cdot 1 + \Theta_{11}^{(1)} \cdot x_1 + \Theta_{12}^{(1)} \cdot x_2 \\
&= (-0.1) \cdot 1 + 1.3 \cdot 0.5 + 0.6 \cdot 1.0 = 1.15.
\end{aligned}
$$

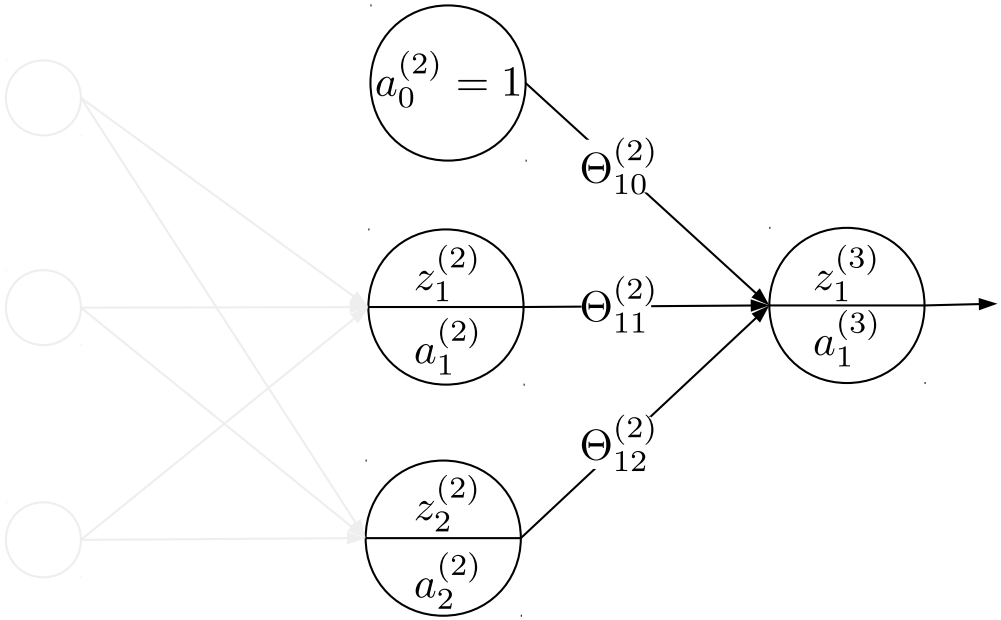
$$
a_1^{(2)} = g(z_1^{(2)}) = \frac{1}{1 + \mathrm{e}^{-1.15}} = \mathbf{0.76}.
$$



$x_1 = 0.5$	$x_2 = 1.0$	
$\Theta_{20}^{(1)} = 0.2$	$\Theta_{21}^{(1)} = -0.5$	$\Theta_{22}^{(1)} = 0$

$$
\begin{aligned}
z_2^{(2)} &= \Theta_{20}^{(1)} \cdot 1 + \Theta_{21}^{(1)} \cdot x_1 + \Theta_{22}^{(1)} \cdot x_2 \\
&= 0.2 \cdot 1 + (-0.5) \cdot 0.5 + 0 \cdot 1.0 = -0.05.
\end{aligned}
$$

$$
a_2^{(2)} = g(z_2^{(2)}) = \frac{1}{1 + \mathrm{e}^{0.05}} = \mathbf{0.49}.
$$



$a_1^{(2)} = 0.76$	$a_2^{(2)} = 0.49$	
$\Theta_{10}^{(2)} = 2.1$	$\Theta_{11}^{(2)} = -1.5$	$\Theta_{12}^{(2)} = -0.3$

$$
\begin{aligned}
z_1^{(3)} &= \Theta_{10}^{(2)} \cdot 1 + \Theta_{11}^{(2)} \cdot a_1^{(2)} + \Theta_{12}^{(2)} \cdot a_2^{(2)} \\
&= 2.1 \cdot 1 + (-1.5) \cdot 0.76 + (-0.3) \cdot 0.49 = 0.81.
\end{aligned}
$$

$$
a_1^{(3)} = g(z_1^{(3)}) = \frac{1}{1 + \mathrm{e}^{-0.81}} = \mathbf{0.69}.
$$

Outline

Vetorização (**forward propagation**)

$$a^{(1)} = x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1.0 \end{bmatrix} \qquad \Theta^{(1)} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} -0.1 & 1.3 & 0.6 \\ 0.2 & -0.5 & 0 \end{bmatrix}$$

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1.15 \\ -0.05 \end{bmatrix}$$

NOTE QUE:

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

ATIVAÇÃO:

$$a^{(2)} = g(z^{(2)}) = \begin{bmatrix} g(z_1^{(2)}) \\ g(z_2^{(2)}) \end{bmatrix} = \begin{bmatrix} 0.76 \\ 0.49 \end{bmatrix}$$

Vetorização (**forward propagation**)

$$a^{(1)} = x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1.0 \end{bmatrix} \qquad \Theta^{(1)} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} -0.1 & 1.3 & 0.6 \\ 0.2 & -0.5 & 0 \end{bmatrix}$$

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1.15 \\ -0.05 \end{bmatrix}$$

NOTE QUE:

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

ATIVAÇÃO:

$$a^{(2)} = g(z^{(2)}) = \begin{bmatrix} g(z_1^{(2)}) \\ g(z_2^{(2)}) \end{bmatrix} = \begin{bmatrix} 0.76 \\ 0.49 \end{bmatrix}$$

Vetorização (**forward propagation**)

$$a^{(1)} = x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0.5 \\ 1.0 \end{bmatrix} \qquad \Theta^{(1)} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} \end{bmatrix} = \begin{bmatrix} -0.1 & 1.3 & 0.6 \\ 0.2 & -0.5 & 0 \end{bmatrix}$$

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1.15 \\ -0.05 \end{bmatrix}$$

NOTE QUE:

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

ATIVAÇÃO:

$$a^{(2)} = g(z^{(2)}) = \begin{bmatrix} g(z_1^{(2)}) \\ g(z_2^{(2)}) \end{bmatrix} = \begin{bmatrix} 0.76 \\ 0.49 \end{bmatrix}$$

Vetorização (**forward propagation**)

ADICIONANDO $a_0^{(2)} = 1$:

$$a^{(2)} = \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0.76 \\ 0.49 \end{bmatrix} \qquad \Theta^{(2)} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \end{bmatrix} = \begin{bmatrix} 2.1 & -1.5 & -0.3 \end{bmatrix}$$

$$z^{(3)} = 0.81.$$

NOTE QUE:

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

ATIVAÇÃO:

$$a^{(3)} = g(z^{(3)}) = 0.69.$$

Vetorização (**forward propagation**)

ADICIONANDO $a_0^{(2)} = 1$:

$$a^{(2)} = \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0.76 \\ 0.49 \end{bmatrix} \qquad \Theta^{(2)} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \end{bmatrix} = \begin{bmatrix} 2.1 & -1.5 & -0.3 \end{bmatrix}$$

$$z^{(3)} = 0.81.$$

NOTE QUE:

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

ATIVAÇÃO:

$$a^{(3)} = g(z^{(3)}) = 0.69.$$

Vetorização (**forward propagation**)

ADICIONANDO $a_0^{(2)} = 1$:

$$a^{(2)} = \begin{bmatrix} a_0^{(2)} \\ a_1^{(2)} \\ a_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0.76 \\ 0.49 \end{bmatrix} \qquad \Theta^{(2)} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \end{bmatrix} = \begin{bmatrix} 2.1 & -1.5 & -0.3 \end{bmatrix}$$

$$z^{(3)} = 0.81.$$

NOTE QUE:

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

ATIVAÇÃO:

$$a^{(3)} = g(z^{(3)}) = 0.69.$$

Assim temos o algoritmo:

$$z^{(2)} = \Theta^{(1)} a^{(1)}$$
$$a^{(2)} = g(z^{(2)})$$
Adicionar $a_0^{(2)} = 1$
$$z^{(3)} = \Theta^{(2)} a^{(2)}$$
$$h_\Theta(x) = a^{(3)} = g(z^{(3)})$$

Em geral:

$$z^{(j)} = \Theta^{(j-1)} a^{(j-1)}$$
$$a^{(j)} = g(z^{(j)})$$

Assim temos o algoritmo:

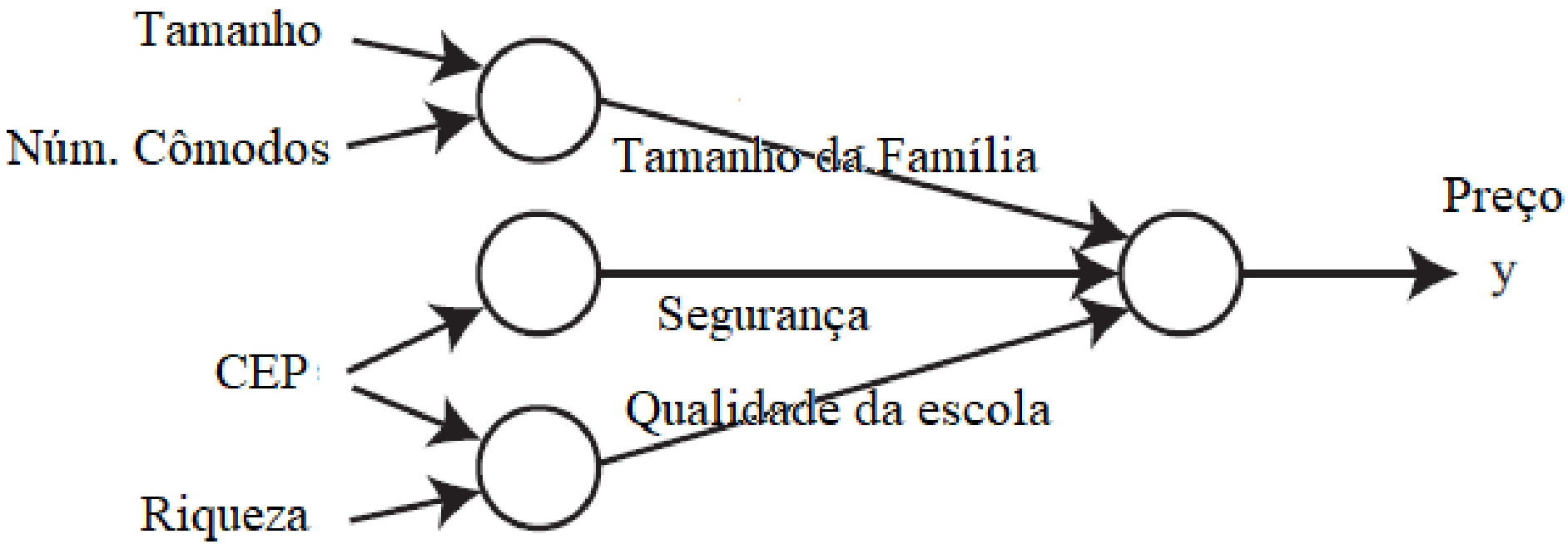
$$z^{(2)} = \Theta^{(1)} a^{(1)}$$
$$a^{(2)} = g(z^{(2)})$$
Adicionar $a_0^{(2)} = 1$
$$z^{(3)} = \Theta^{(2)} a^{(2)}$$
$$h_\Theta(x) = a^{(3)} = g(z^{(3)})$$

Em geral:

$$z^{(j)} = \Theta^{(j-1)} a^{(j-1)}$$
$$a^{(j)} = g(z^{(j)})$$

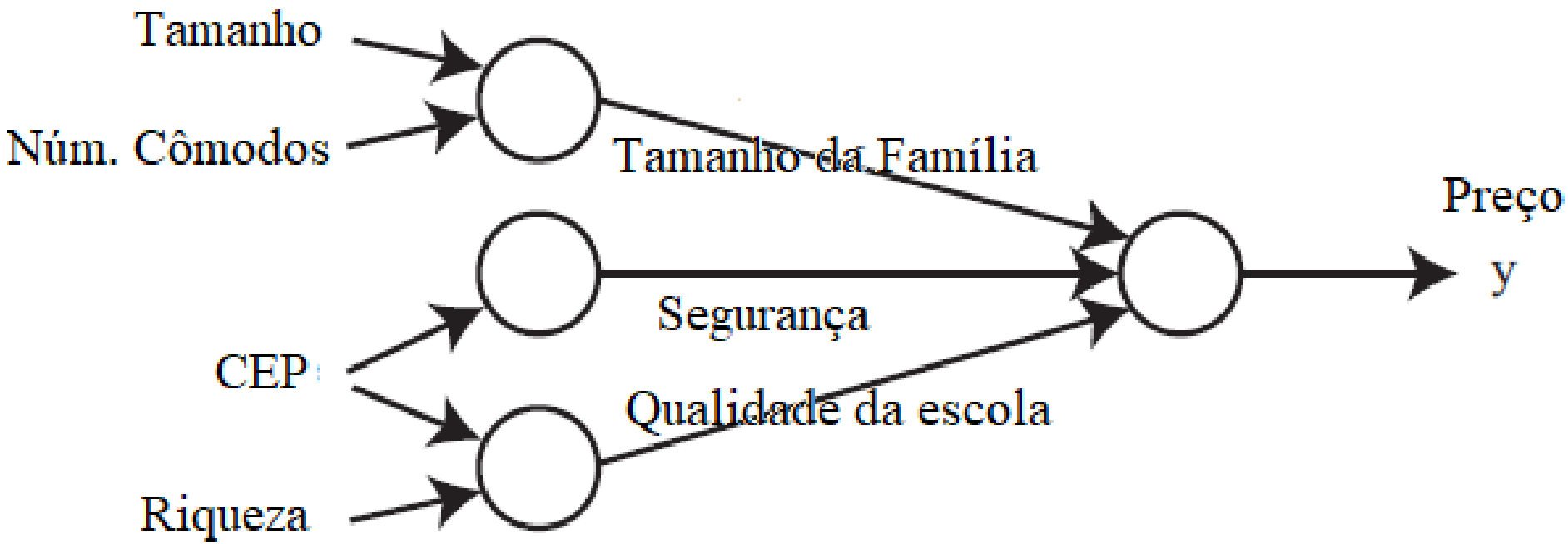
Outline

Intuição



- MELHOR PARTE: A rede neural "descobre" esses novos atributos automaticamente!

Intuição



- MELHOR PARTE: A rede neural "descobre" esses novos atributos automaticamente!

Outline

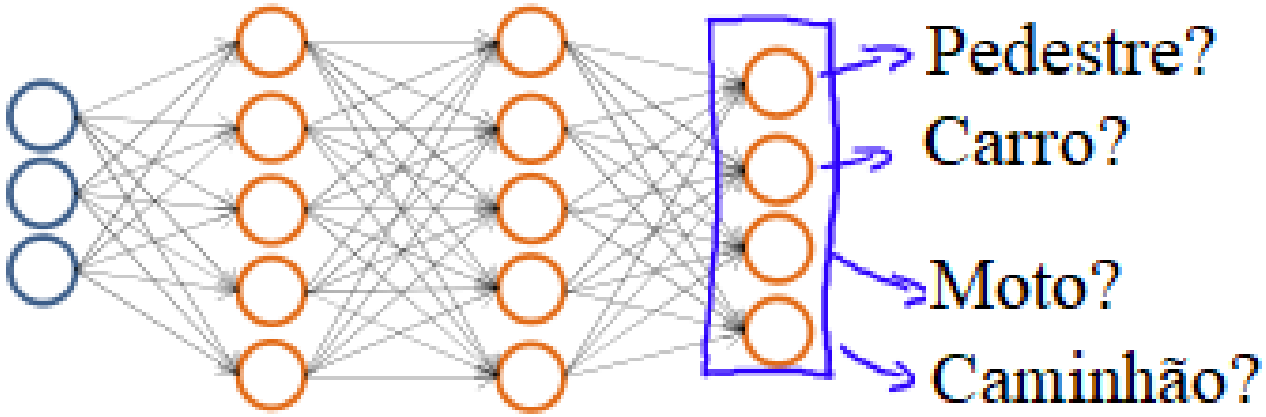
- Abordagem "um-*vs*-todos" que já vimos.

Pedestre

Carro

Moto

Caminhão



Pedestre

$$h_\Theta(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Carro

$$h_\Theta(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Moto

$$h_\Theta(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Caminhão

$$h_\Theta(x) \approx \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

- NOTA: Conjunto de treinamento:

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots (x^{(m)}, y^{(m)}),$$

porém agora $y^{(i)}$ é um vetor:

$$\begin{array}{cccc} \text{Pedestre} & \text{Carro} & \text{Moto} & \text{Caminhão} \\ y^{(i)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} & y^{(i)} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} & y^{(i)} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} & y^{(i)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \end{array}$$