

# Redes Neurais

Prof<sup>a</sup> Deborah Magalhães  
Sistemas de Informação



UNIVERSIDADE  
FEDERAL DO PIAUÍ

# Olá!



## **Curso: Bacharelado em Sistema de Informação**

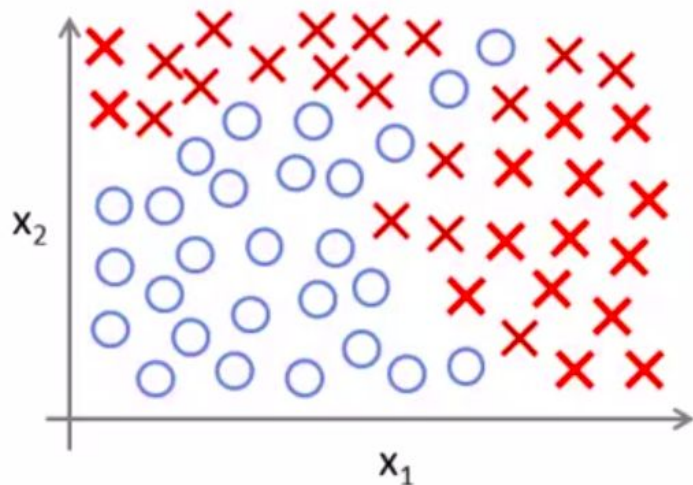
Disciplina: Sistemas Inteligentes

### ▶ **Visão Geral: Redes Neurais**

Você pode me encontrar em [deborah.vm@gmail.com](mailto:deborah.vm@gmail.com)  
(Dúvidas e sugestões serão bem-vindas =D)

Por que precisamos de outro algoritmo de aprendizado se já temos regressão linear e logística?

# Regressão Logística



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

## Características:

$X_1$  = metragem

$X_2$  = número de quartos

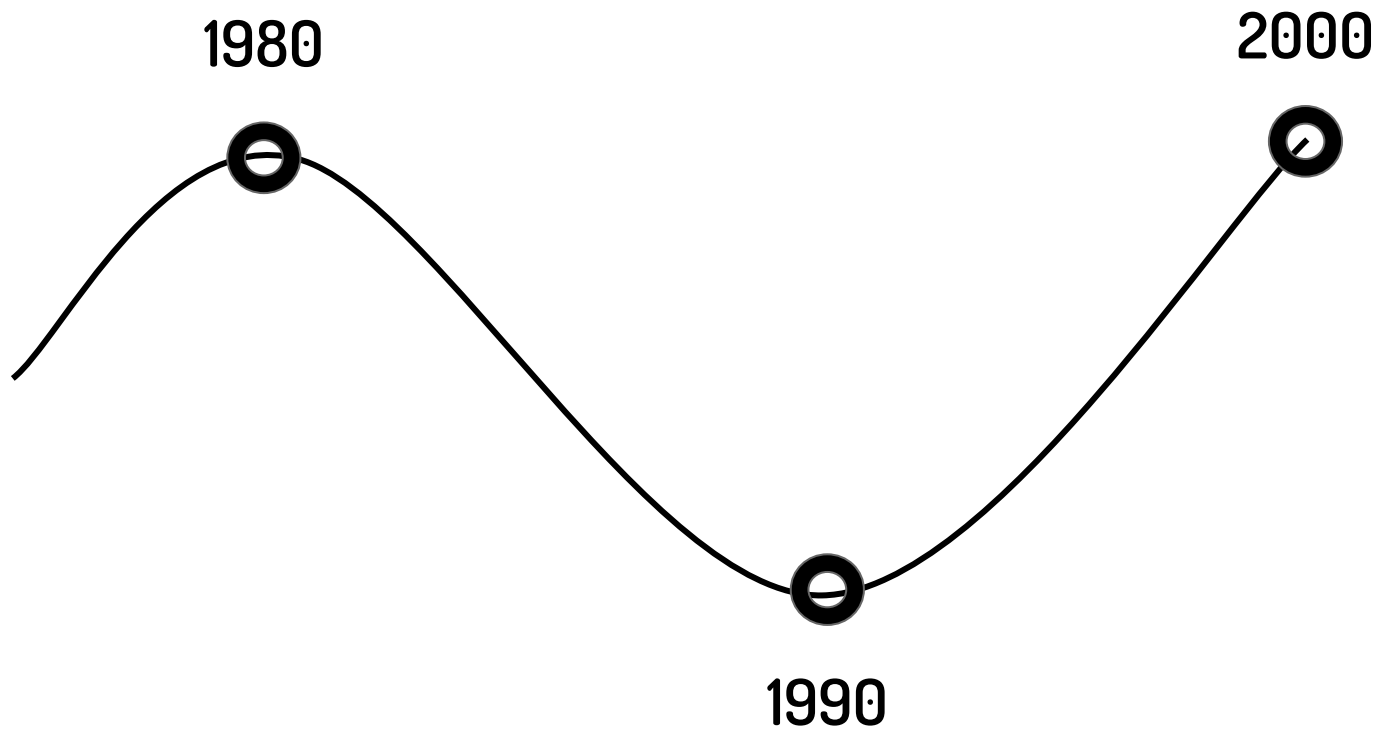
$X_3$  = número de andares

$X_4$  = ano de construção

.

$X_{100}$  = área externa

# Redes Neurais



# Redes Neurais

- ▶ Funcionamento inspirado no neurônio biológico.
- ▶ Tarefas de Aprendizado de Máquina
  - ▶ Classificação
  - ▶ Regressão

# Neurônio biológico X Neurônio Artificial

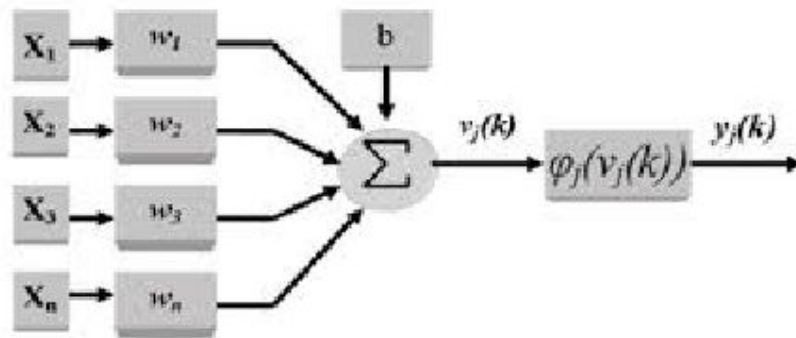
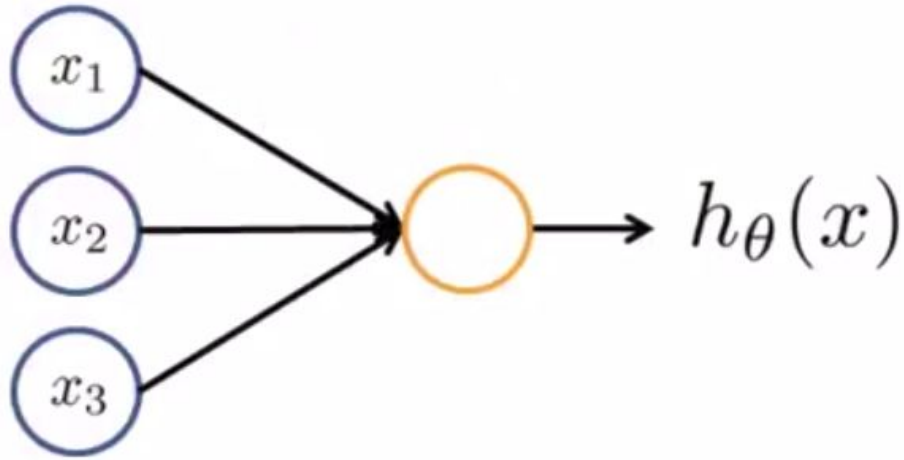


Figura 2: Representação do neurônio artificial.

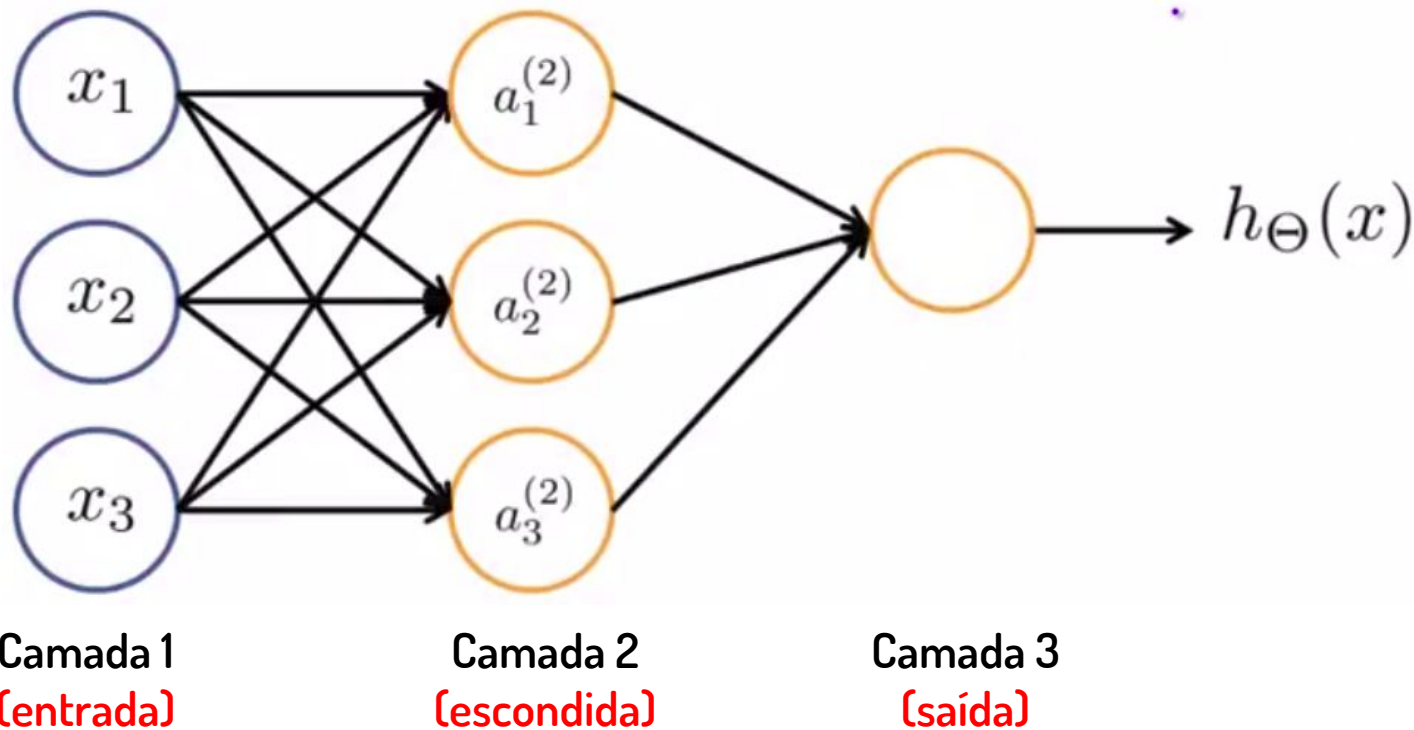
# Modelo de Neurônio: unidade logística



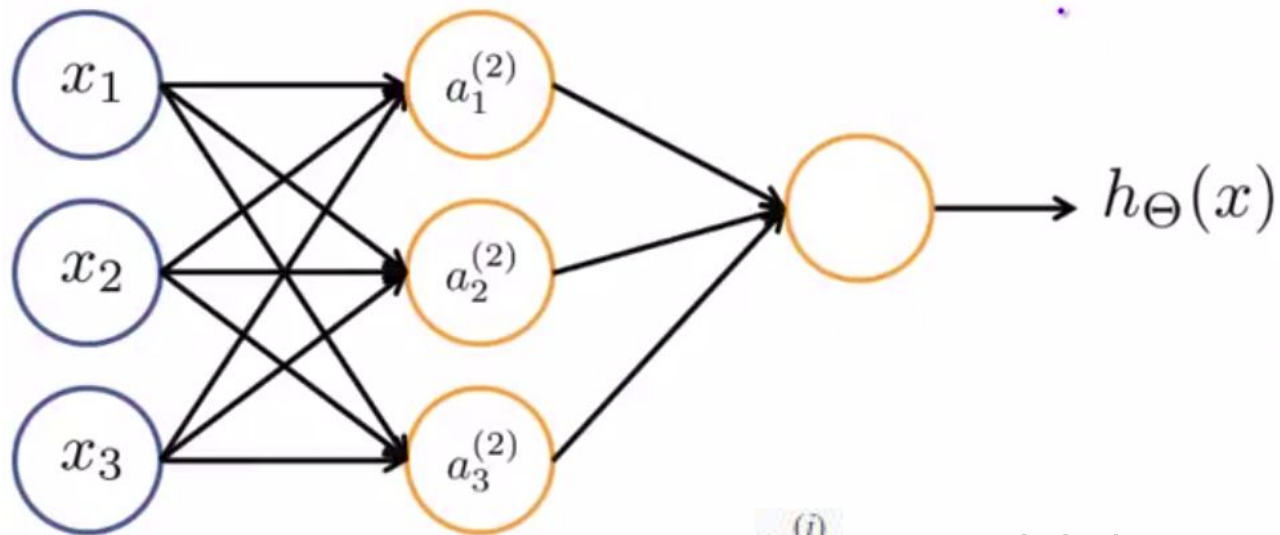
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$



# Rede Neural



# Rede Neural

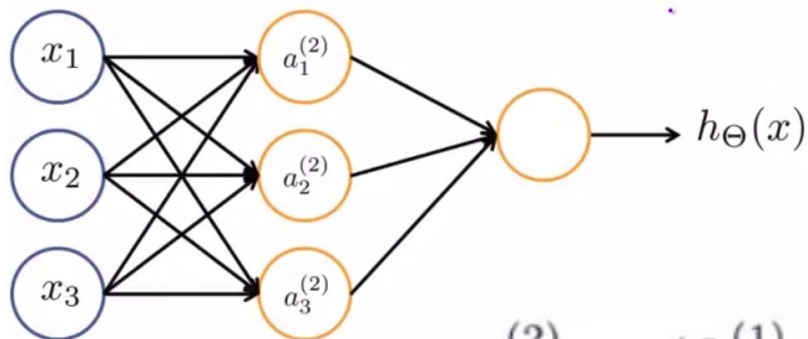


$a_i^{(j)}$   
 $\Theta^{(j)}$

-> unidade de ativação  $i$  na camada  $j$

-> matriz de pesos que mapeia a  
camada  $j$  na camada  $j+1$

# Rede Neural



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

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

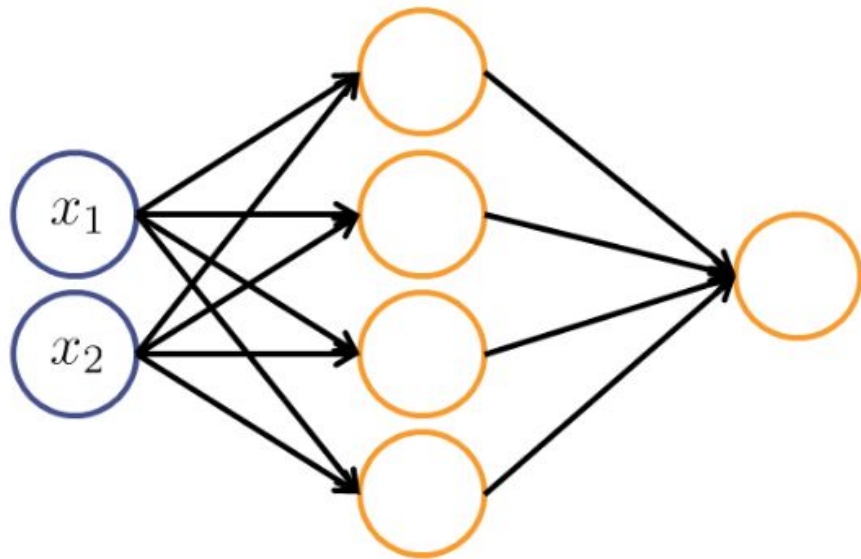
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

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

Conclusão: a rede neural faz a mesma coisa que a regressão logística, exceto que ao invés de se basear na características  $x_1, x_2, x_3..$ , a rede se baseia nas unidades de ativação que correspondem a combinação linear das características de entrada

## Questão:

Qual a dimensão de  $\Theta^{(1)}$  considerando a rede abaixo? (Dica: não esquecer da unidade de bias).



## Questão:

Considere uma rede com 4 camadas, sendo 3 entradas, uma camada escondida com 3 unidades de ativação, outra camada escondida com 2 unidades de ativação e uma camada de saída. Qual das opções abaixo representa  $a_1^{(4)}$  ?

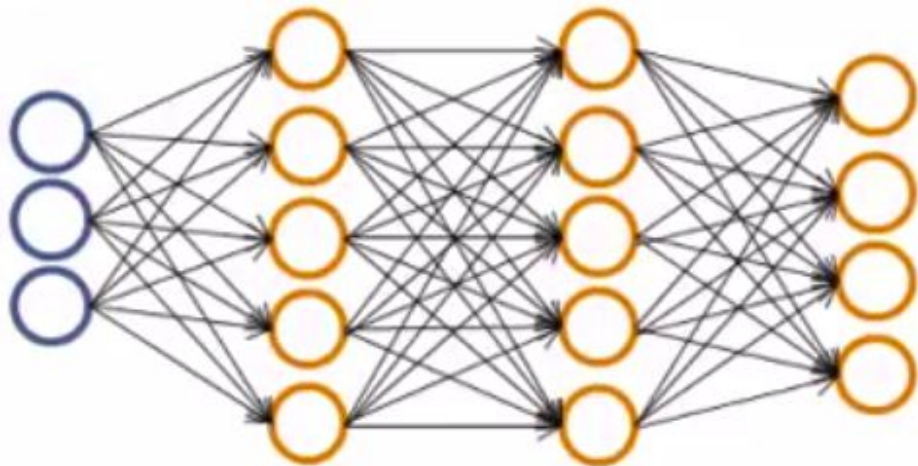
☐  $\ominus^{(1)} a^{(2)}$

☐  $\ominus^{(3)} a^{(4)}$

☐  $\ominus^{(3)} a^{(3)}$

☐  $\ominus^{(1)} x$

# Rede Neural Multiclasse



$L$  = número total de camadas da rede  
 $S_l$  = número de unidades de ativação por camada



# Função Custo

## Regressão Logística

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

## Rede Neural

$$J(\Theta) = -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[ y_k^{(i)} \log((h_{\Theta}(x^{(i)}))_k) + (1 - y_k^{(i)}) \log(1 - (h_{\Theta}(x^{(i)}))_k) \right] + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (\Theta_{j,i}^{(l)})^2$$



# Algoritmo de Retropropagação

Objetivo:  $\min_{\Theta} J(\Theta)$

$$- J(\Theta)$$

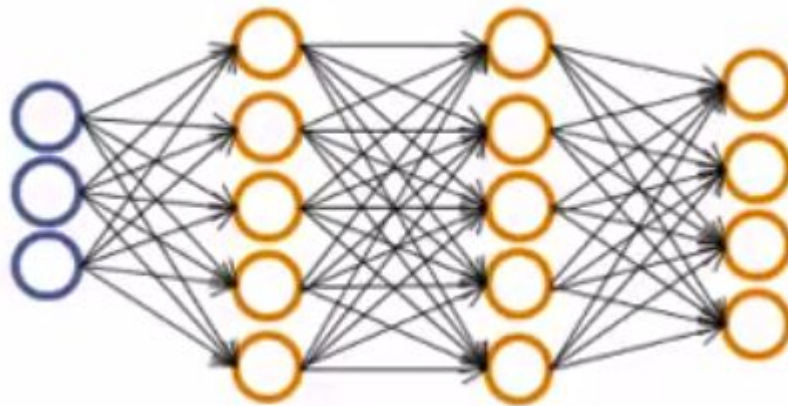
$$- \frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$$

# Algoritmo de Retropropagação

Passo 1: inicializar as unidades de ativação de entrada

Para todo exemplo de treinamento  $(x^{(t)}, y^{(t)})$ , onde  $t = 1$  até  $m$

1.  $a^{(1)} := x^{(t)}$



# Algoritmo de Retropropagação

Passo 2: propagação para frente p/ achar  $a^{(2)} \dots a^{(l)}$

$$a^{(1)} =$$

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

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

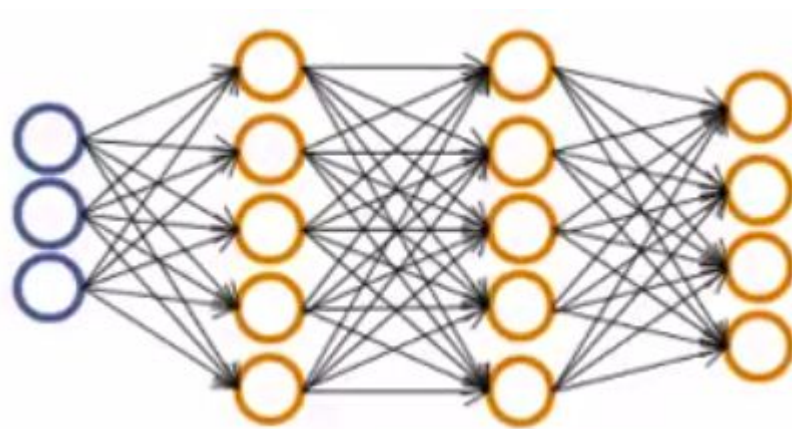
$$z^{(3)} =$$

$$a^{(3)} =$$

$$z^{(4)} =$$

$$a^{(4)} =$$

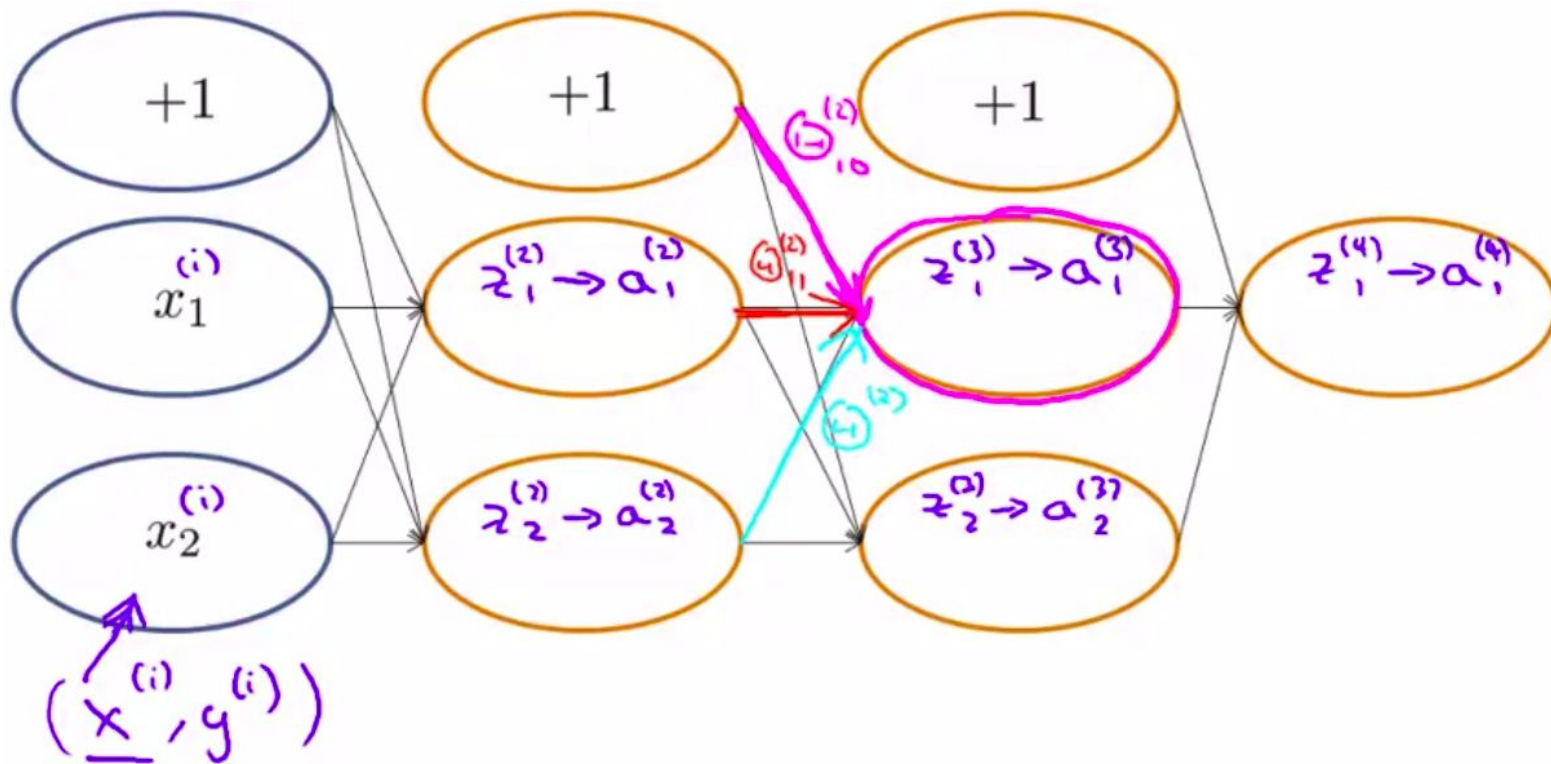
?



# Algoritmo de Retropropagação

20

Passo 2: propagação para frente p/ achar  $a^{(2)} \dots a^{(l)}$

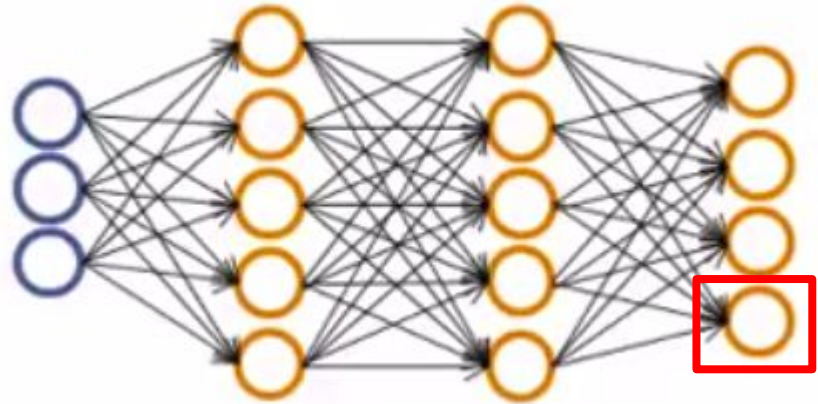


# Algoritmo de Retropropagação

Passo 3: calcular erro associado a última camada de ativação

$\delta^{(l)}_j$  = erro associado ao nó/neurônio  $j$  da camada  $L$

$$\delta^{(4)}_j = a^{(4)}_j - y_j$$



# Algoritmo de Retropropagação

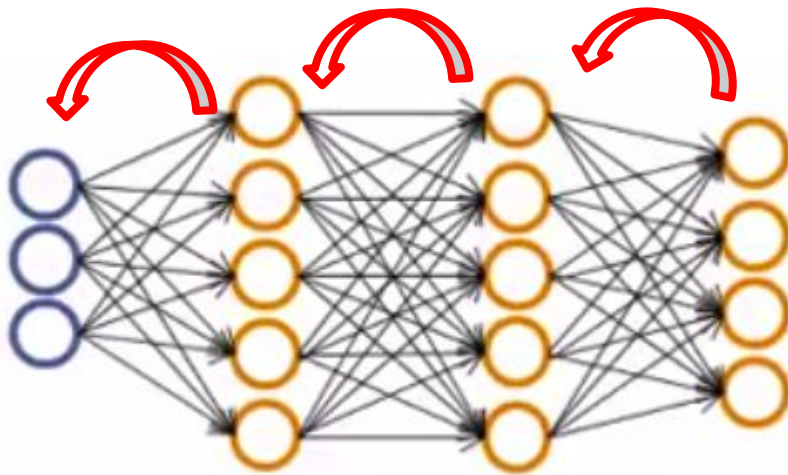
Passo 4: calcular os erros associados às outras camadas

Calcular  $\delta^{(L-1)}, \delta^{(L-2)}, \dots, \delta^{(2)}$

$$\delta^{(4)} = a^{(4)} - y \rightarrow a^4 - y$$

$$\delta^{(3)} = ?$$

$$\delta^{(2)} = ?$$



# Implementação da RN

## ► Regressão Logística

```
function [jVal, gradient] = costFunction(theta)
...
optTheta = fminunc(@costFunction, initialTheta, options)
```

## ► Rede Neural

- $\Theta(1), \Theta(2), \Theta(3), \dots$
- $D(1), D(2), D(3), \dots$

# Implementação da RN

## Redimensionando os parâmetros

### ► No Octave:

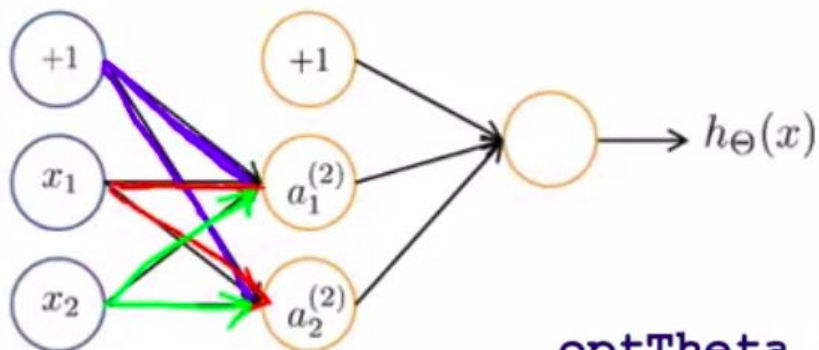
```
thetaVec = [ Theta1(:); Theta2(:); Theta3(:)];  
DVec = [D1(:); D2(:); D3(:)];
```

```
Theta1 = reshape(thetaVec(1:110),10,11);  
Theta2 = reshape(thetaVec(111:220),10,11);  
Theta3 = reshape(thetaVec(221:231),1,11);
```



# Implementação da RN

## Inicialização randômica



```
optTheta = fminunc(@costFunction,  
    initialTheta, options)
```

```
initialTheta = zeros(n,1)
```

# Implementação da RN

## Inicialização randômica

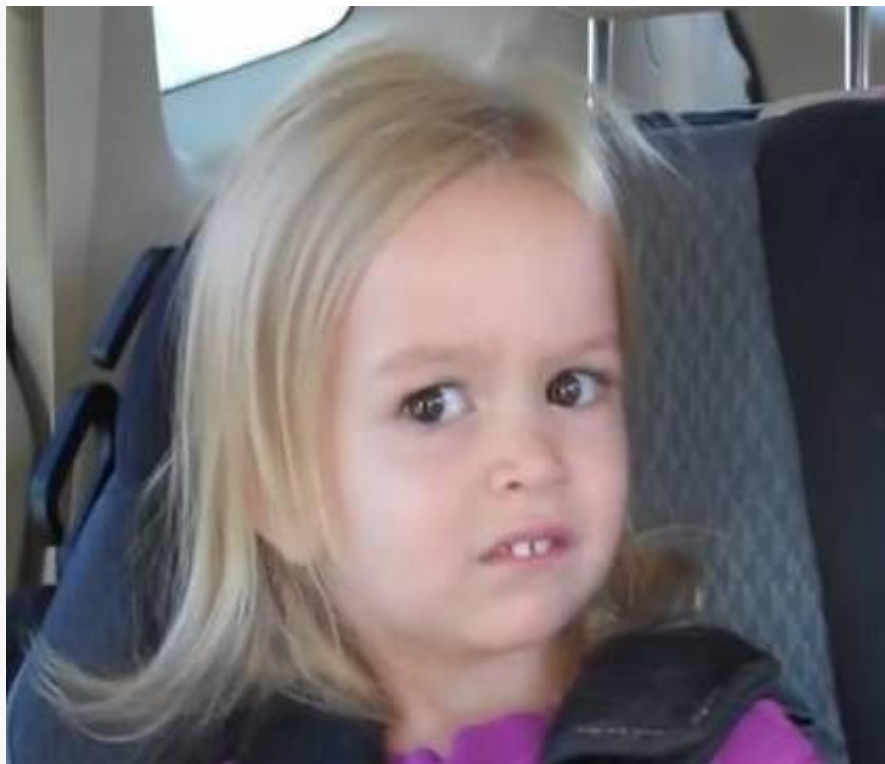
► No Octave:

```
Theta1 = rand(10,11)*(2*INIT_EPSILON)  
- INIT_EPSILON;
```

```
Theta2 = rand(1,11)*(2*INIT_EPSILON)  
- INIT_EPSILON;
```

# Resumindo

- ▶ Na hora de treinar sua rede os seguintes passos devem ser utilizados:
  - i. Definir uma arquitetura
  - ii. Inicializar os pesos randomicamente
  - iii. Implementar o algoritmo de propagação para frente a fim de encontrar  $h_{\Theta}(x^{(i)})$
  - iv. Implementar o código para calcular a função custo  $J(\Theta)$
  - v. Implementar o algoritmo de retropropagação para calcular as função gradiente  $\frac{\partial}{\partial \Theta_{jk}^{(l)}} J(\Theta)$
  - vi. Utilizar o gradiente descendente para minimizar a função custo



**Dúvidas? Sugestões?  
Inquietações?  
Aconselhamentos?**

- ▶ Desabafe em:  
[deborah.vm@gmail.com](mailto:deborah.vm@gmail.com)