

# **GEX1082 - Tópicos Especiais em Computação XXXIII**

## **Deep Learning**

**Redes Neurais Artificiais**

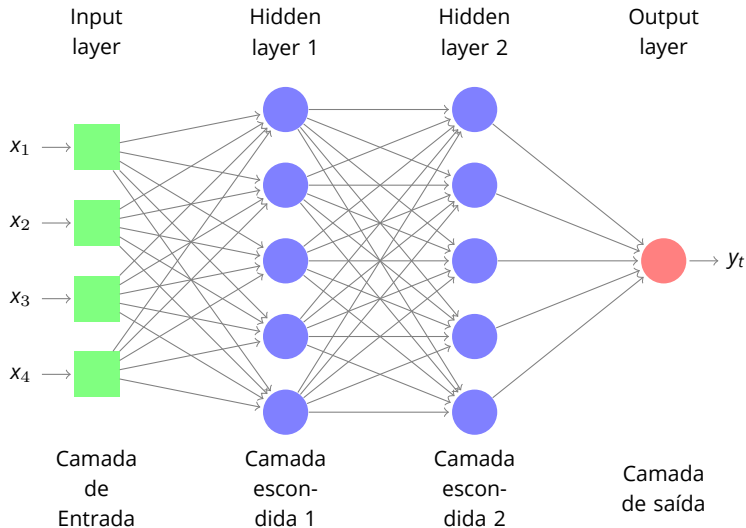


1100/1101 - CIÊNCIA DA COMPUTAÇÃO

Prof. Dr. Giancarlo D. Salton

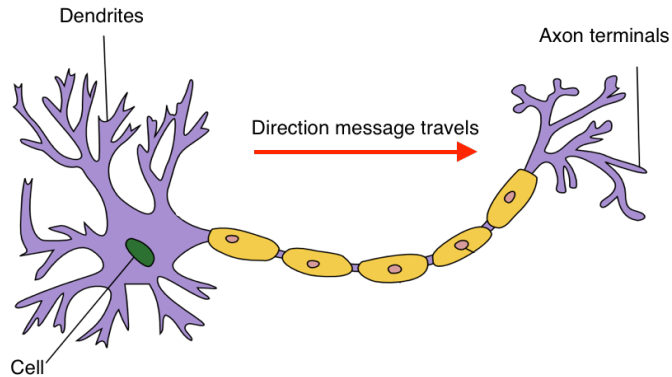
## Neurônios Artificiais

## Arquitetura das Redes Neurais Artificiais

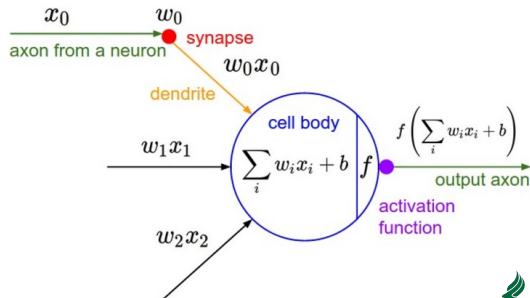
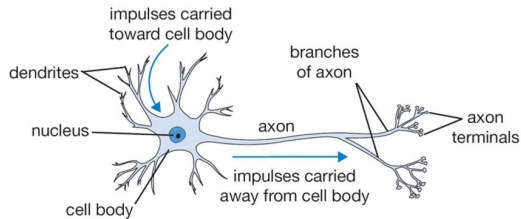




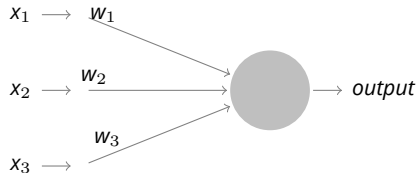
# Inspiração



# Inspiração



## Perceptron

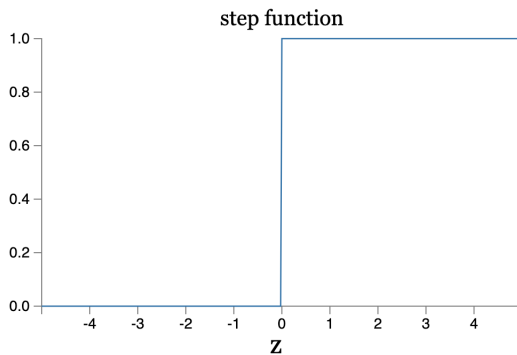


$$output = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

(1)

onde  $x_i$  são as *features* de entrada e  $w_i$  são os parâmetros associados à cada conexão por onde passam as *features*

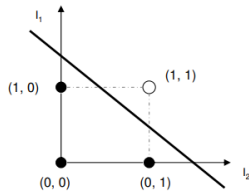
## Step Function



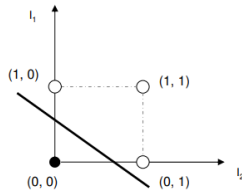


## Problema: função XOR

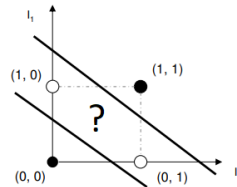
AND		
$I_1$	$I_2$	out
0	0	0
0	1	0
1	0	0
1	1	1



OR		
$I_1$	$I_2$	out
0	0	0
0	1	1
1	0	1
1	1	1

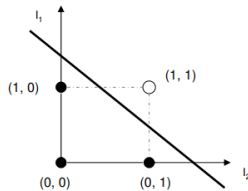


XOR		
$I_1$	$I_2$	out
0	0	0
0	1	1
1	0	1
1	1	0

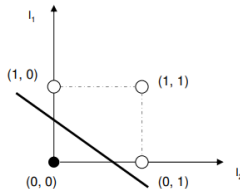


## Problema: função XOR

AND		
$I_1$	$I_2$	out
0	0	0
0	1	0
1	0	0
1	1	1

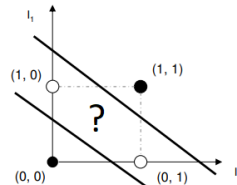


OR		
$I_1$	$I_2$	out
0	0	0
0	1	1
1	0	1
1	1	1

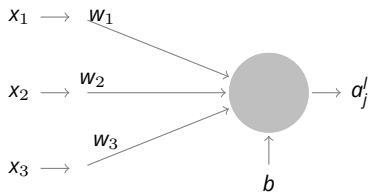


Esta descoberta fez com que as pesquisas em redes neurais parasse por vários anos.

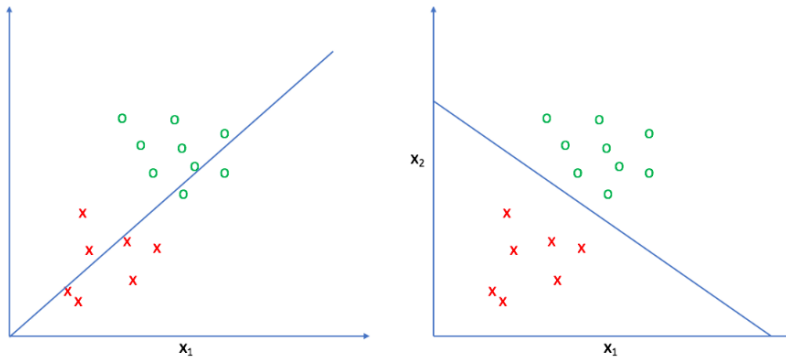
XOR		
$I_1$	$I_2$	out
0	0	0
0	1	1
1	0	1
1	1	0



## Neurônios



## Por quê o bias é importante?



Se não utilizarmos o bias  $b$ , a nossa linha é obrigada a passar pela origem  $(0, 0)$ . Por outro lado, ao adicionarmos  $b$ , também chamado de coeficiente angular ou intercept, temos mais flexibilidade para posicionar a linha.

## Regressão linear

$$Z = w_0 + w_1x_1 + w_2x_2 + w_3x_3$$

## Regressão linear

$$z = w_0 + w_1x_1 + w_2x_2 + w_3x_3$$

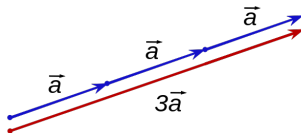
$$z = b + w_1x_1 + w_2x_2 + w_3x_3$$

## Neurônio artificial

$$z = b + w_1x_1 + w_2x_2 + w_3x_3$$

## Neurônio artificial

$$z = b + w_1x_1 + w_2x_2 + w_3x_3$$



Um neurônio artificial faz, inicialmente, uma transformação linear dos dados.



### Neurônio artificial em notação matemática

$$z = b + \sum_j w_j x_j$$

Perceba que  $\sum_j w_j x_j$  é equivalente a um produto escalar de vetores.

### Neurônio artificial em notação matemática

$$z = b + \sum_j w_j x_j$$

## Neurônio artificial em notação de vetores

$$z = \mathbf{b} + \mathbf{W}^T \mathbf{x}$$

Perceba que para realizar o cálculo do produto escalar, precisamos transpor o vetor (adiante falaremos mais sobre isso).

### Neurônio artificial em notação de vetores

$$z = \mathbf{b} + \mathbf{W}^T \mathbf{x}$$

## Funções de ativação

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$a = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$a = \text{ReLu}(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

(2)

Esta equação também é conhecida como "sigmoide".

## Funções de ativação

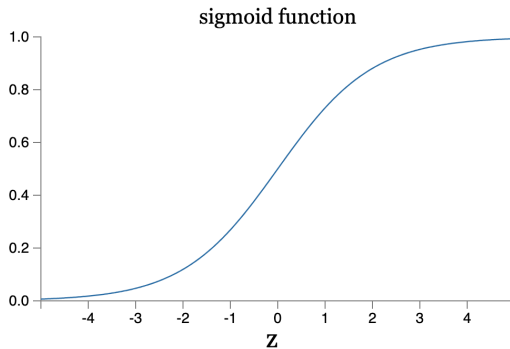
$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

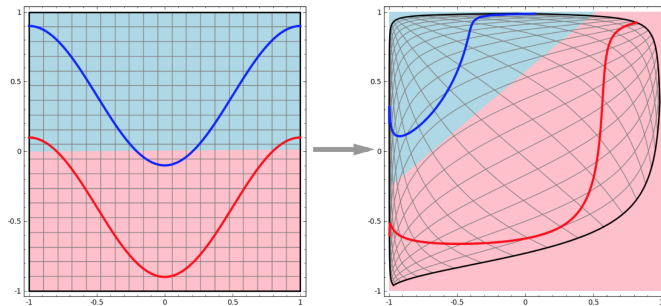
$$a = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$a = \text{ReLu}(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

(2)

## Sigmoid Function





Ao incluirmos um transformação não-linear, alteramos o formato do espaço onde os exemplos estão inseridos.  
(Retornaremos a este assunto mais à frente durante o desenvolvimento do CCR.)



## Funções de Ativação na Saída

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$a = \text{softmax}(x)_j = \frac{e^{x_j}}{\sum_k e^{x_k}}$$

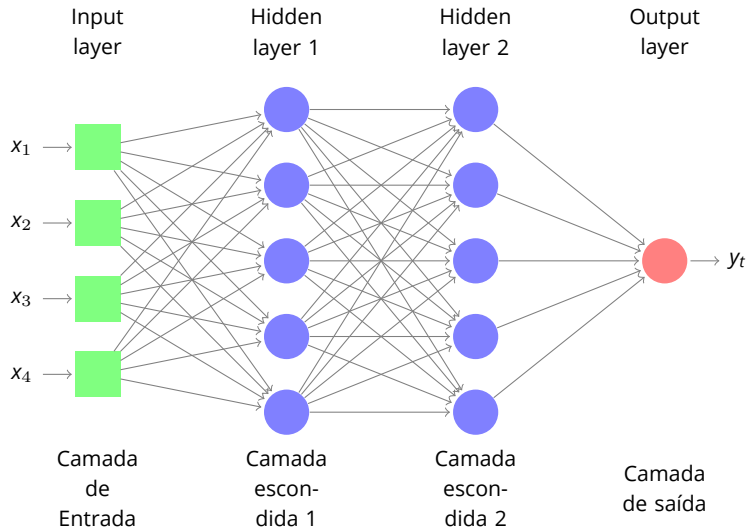
## Funções de Ativação na Saída

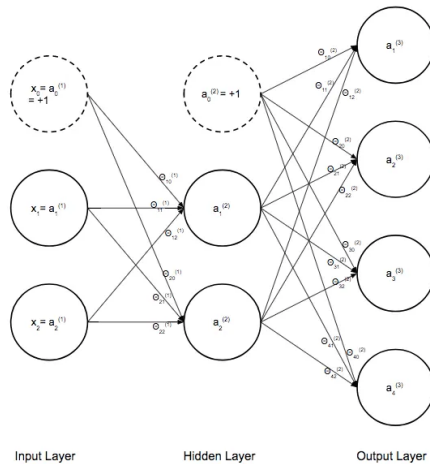
$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$

$$a = \text{softmax}(x)_j = \frac{e^{x_j}}{\sum_k e^{x_k}}$$

A função *softmax* define uma distribuição de probabilidades entre os possíveis *targets*. Note que a sigmoide também define uma distribuição de probabilidade, mas entre duas opções apenas.

# Arquitetura das Redes Neurais Artificiais

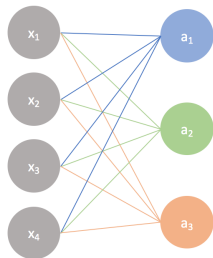




## Uma rede neural simples

Input layer

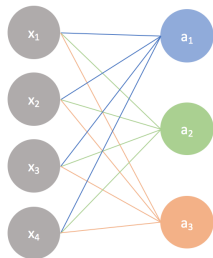
Output layer



$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \end{bmatrix} \xrightarrow{\text{activation}} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

## Uma rede neural simples

Input layer      Output layer



Perceba que se usada a função  $\sigma$  como ativação, esta rede se torna uma regressão logística do tipo 1-vs-All.

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \end{bmatrix} \xrightarrow{\text{activation}} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

### Cálculo da ativação dos neurônios

$$a_i^{(l)} = g \left( b_i^{(l)} + \sum_k w_{ik}^{(l)} a_k^{(l-1)} \right)$$

- A ativação  $a_i^{(l)}$ , ou seja, ativação  $a$  do neurônio  $i$  da camada  $l$  é relacionada com as ativações da camada  $l - 1$  através das conexões  $k$ .
- Desta forma, cada ativação de um neurônio na camada  $l - 1$ , se torna entrada para os neurônios da camada seguinte,  $l$ .
- Em formato de matrizes



**Em formato de matriz**

$$a^{(l)} = g \left( b^{(l)} + W^{T(l)} a^{(l-1)} \right)$$

- Perceba que quando escrevermos em formato de matriz, devemos transpor a matriz de parâmetros para que cada vetor fique em uma linha e assim possamos realizar o cálculo do produto escalar.

### Exemplo com uma rede de 3 camadas utilizando notação de matrizes

$$h^{(1)} = g \left( b^{(1)} + W^{T(1)}x \right)$$
$$o = g \left( b^{(2)} + W^{T(2)}h^{(1)} \right)$$

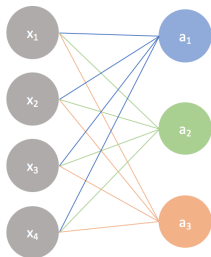
onde

- ✓  $x = a^{(0)}$
- ✓  $h^{(1)} = a^{(1)}$
- ✓  $o = a^{(2)}$

## Utilizando vários exemplos ao mesmo tempo

Input layer

Output layer



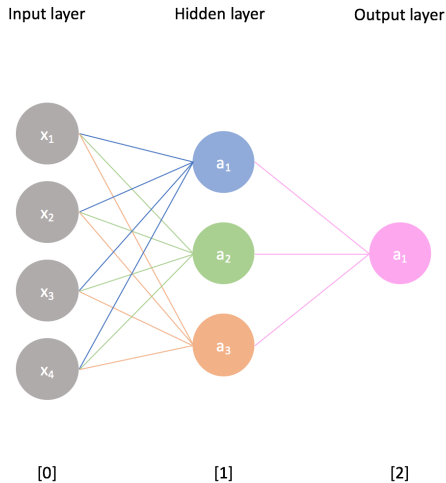
$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} \text{Observation 1} & \text{Observation 2} & \text{Observation 3} & \text{Observation 4} \\ x_1 & x_1 & x_1 & x_1 \\ x_2 & x_2 & x_2 & x_2 \\ x_3 & x_3 & x_3 & x_3 \\ x_4 & x_4 & x_4 & x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix} \xrightarrow{\text{activation}} \begin{bmatrix} \text{Observation 1} & \text{Observation 2} & \text{Observation 3} & \text{Observation 4} \\ a_1 & a_1 & a_1 & a_1 \\ a_2 & a_2 & a_2 & a_2 \\ a_3 & a_3 & a_3 & a_3 \end{bmatrix}$$

## Como ficam os parâmetros

$$w_1 = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}, w_2 = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}, w_3 = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \longrightarrow \begin{bmatrix} \leftarrow w_1^T \rightarrow \\ \leftarrow w_2^T \rightarrow \\ \leftarrow w_3^T \rightarrow \end{bmatrix}$$

## Incluindo hidden layers



## Regras das Camadas

### Camada de Entrada

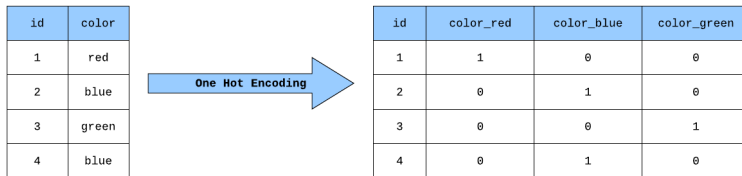
- possui 1 (um) neurônio para cada *feature* de entrada.
- os neurônios desta camada possuem como ativação a *função identidade*:  $a_i = x_i$

## Regras das Camadas

### Camada de Saída

- caso alvo seja de *regressão* (e em algumas classificações binárias) possui apenas 1 (um) neurônio
- senão, terá um neurônio para cada *target* identificado no problema
  - ✓ e.g., prever polaridade do sentimento de um *review* de produto, requer 3 (três) neurônios: positivo, negativo e neutro
- A ativação da camada de saída depende do problema:
  - ✓ Se regressão, pode ser a função identidade
  - ✓ Se classificação binária:  $\sigma$ ; ou *softmax*
  - ✓ Se classificação com mais de duas classes: *softmax*

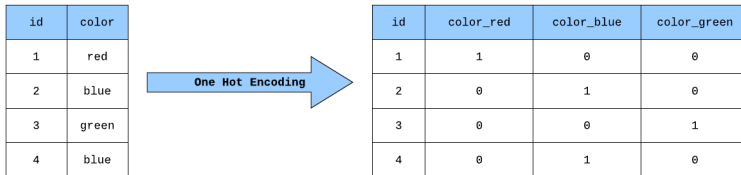
## One-hot-encoding





## One-hot-encoding

Veja que, como há um neurônio para cada *target* em tarefas de classificação, esperamos que a rede neural tenha “certeza” sobre a resposta. Desta forma, a rede deveria emitir uma saída com 100% para a resposta correta e 0% para as demais.



## Regras das Camadas

### Demais camadas (camadas escondidas)

- Número variável de neurônios em cada camada ou podem ser todas iguais
- Em geral, todas as camadas possuem a mesma função de ativação

## Neurônios Artificiais

## Arquitetura das Redes Neurais Artificiais