

# Inteligência Artificial

Raoni F. S. Teixeira

Aula 9 - Redes Neurais

## 1 Introdução

Uma rede neural é um modelo formado por neurônios interconectados organizados em camadas de entrada, ocultas e saída. Essa estrutura permite ao modelo aprender padrões complexos e resolver diferentes tipos de problemas.

O termo “rede neural” surgiu como uma tentativa de representar matematicamente o sistema nervoso biológico. Hoje, no entanto, ele é usado mais por razões históricas do que por buscar replicar o funcionamento do cérebro humano.

## 2 Rede Neural

O neurônio é a unidade básica de processamento de uma rede neural. Ele combina linearmente as entradas e aplica uma função de ativação  $\sigma$  para gerar a saída.

A Figura 1 mostra um neurônio com três entradas. A saída é o resultado da função  $\sigma$  para a combinação gerada pelos parâmetros  $w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$  e  $b$ .

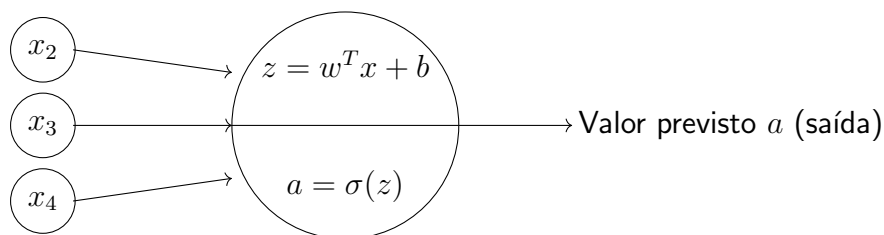


Figura 1: Neurônio artificial com quatro entradas.

Em geral, *sigma* é uma função não-linear, como sigmoide, tanh ou ReLU:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (\text{sigmoide}), \quad (1)$$

$$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (\text{tanh}) \text{ e} \quad (2)$$

$$\sigma(z) = \max(z, 0) \quad (\text{ReLU}). \quad (3)$$

Quando  $\sigma$  é a função sigmoide, o neurônio é equivalente ao modelo de regressão logística da aula anterior.

Os neurônios são conectados em camadas. Uma rede neural é, portanto, como um sistema de peças interligadas, onde camadas de neurônios simples se combinam para criar estruturas complexas. Por exemplo, a Figura 2 mostra uma rede neural com três camadas:

- **Camada de entrada**, que recebe os dados de entrada  $x_1$ ,  $x_2$  e  $x_3$ .
- **Camada intermediária (ou oculta)**, que processa os dados intermediários e cria representações abstratas e
- **Camada de saída**, que gera a previsão final do modelo.

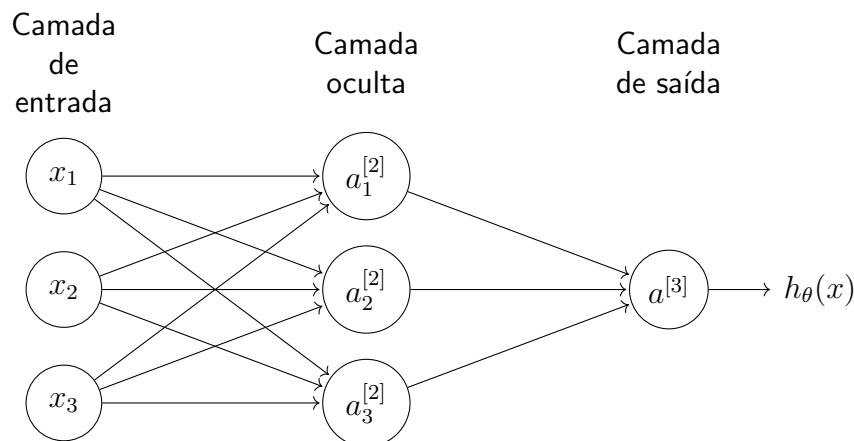


Figura 2: Rede neural com três camadas, três neurônios na camada oculta e um na saída.

A saída da rede é calculada progressivamente, camada por camada. As funções não lineares aplicadas em cada nível permitem identificar padrões complexos que modelos lineares não conseguem.

A saída da camada intermediária da Figura 2 é dada pelo vetor  $a^{[2]}$ :

$$a_1^{[2]} = \sigma(W_{1,1}^{[2]}x_1 + W_{1,2}^{[2]}x_2 + W_{1,3}^{[2]}x_3 + b_1^{[2]}), \quad (4)$$

$$a_2^{[2]} = \sigma(W_{2,1}^{[2]}x_1 + W_{2,2}^{[2]}x_2 + W_{2,3}^{[2]}x_3 + b_2^{[2]}), \quad (5)$$

$$a_3^{[2]} = \sigma(W_{3,1}^{[2]}x_1 + W_{3,2}^{[2]}x_2 + W_{3,3}^{[2]}x_3 + b_3^{[2]}), \quad (6)$$

O valor da camada de saída é um número:

$$a^{[3]} = \sigma(W_{1,1}^{[3]}a_1^{[2]} + W_{1,2}^{[3]}a_2^{[2]} + W_{1,3}^{[3]}a_3^{[2]} + b_1^{[3]}). \quad (7)$$

Os parâmetros das camadas oculta e de saída são representados pelas matrizes  $W^{[2]} \in \mathbb{R}^{3 \times 3}$ ,  $b^{[2]} \in \mathbb{R}^{1 \times 3}$ ,  $W^{[3]} \in \mathbb{R}^{1 \times 3}$  e  $b^{[3]} \in \mathbb{R}^{1 \times 1}$  e  $a_i^{[l]}$  indica  $i$ -ésimo neurônio da camada  $l$ .

Sabendo que  $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$ ,  $W^{[1]} = \begin{bmatrix} W_{1,1}^{[2]} & W_{1,2}^{[2]} & W_{1,3}^{[2]} \\ W_{2,1}^{[2]} & W_{2,2}^{[2]} & W_{2,3}^{[2]} \\ W_{3,1}^{[2]} & W_{3,2}^{[2]} & W_{3,3}^{[2]} \end{bmatrix}$  e  $W^{[3]} = \begin{bmatrix} W_{1,1}^{[3]} & W_{1,2}^{[3]} & W_{1,3}^{[3]} \end{bmatrix}$ ,

podemos vetorizar esses os cálculos da seguinte forma:

1.  $z^{[2]} = W^{[2]}x + b^{[2]}$ ,
2.  $a^{[2]} = \sigma(z^{[2]})$
3.  $z^{[3]} = W^{[3]}a^{[2]} + b^{[3]}$
4.  $a^{[3]} = \sigma(z^{[3]}).$

Essas operações são mais eficientes porque são processadas em blocos e não individualmente. Dessa forma, é possível aproveitar o poder dos processadores modernos, especialmente GPUs.

## 2.1 Arquitetura da Rede

A arquitetura de uma rede neural depende do problema específico e deve ser ajustada por meio de experimentos. Em geral, é preferível aumentar a quantidade de camadas, em vez de aumentar o número de neurônios em cada camada. No entanto, não existem números mágicos para definir a configuração ideal.

Outro ponto importante é que, em problemas de regressão, a saída da rede não deve utilizar uma função de ativação não-linear estritamente positiva (como ReLU ou sigmoide), pois o valor  $y$  pode ser negativo.

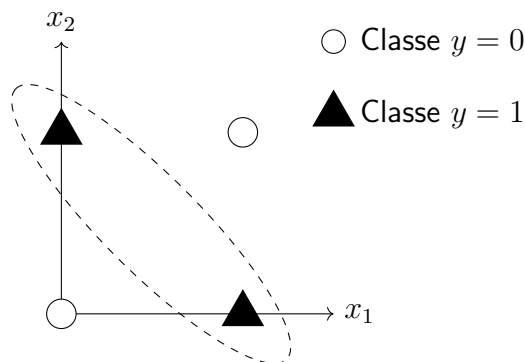


Figura 3: Exemplo de classificação não-linear com operação XOR.

## 2.2 Exemplo do Operador XOR

As redes neurais resolvem problemas que modelos lineares não conseguem. Um exemplo clássico é o operador XOR, onde os pontos de uma classe ( $y = 1$ ) não podem ser separados dos da outra classe ( $y = 0$ ) com uma linha reta, como mostrado na Figura 3.

Uma rede neural com três camadas e ativação sigmoide, ilustrada na Figura 4, resolve esse problema ao criar padrões intermediários. Os neurônios das camadas ocultas aprendem operações simples, como  $x_1$  AND  $x_2$  e NOT  $x_1$  AND NOT  $x_2$ . A camada final combina essas representações para produzir a saída XOR. A Tabela 1 mostra a tabela verdade desses operadores.

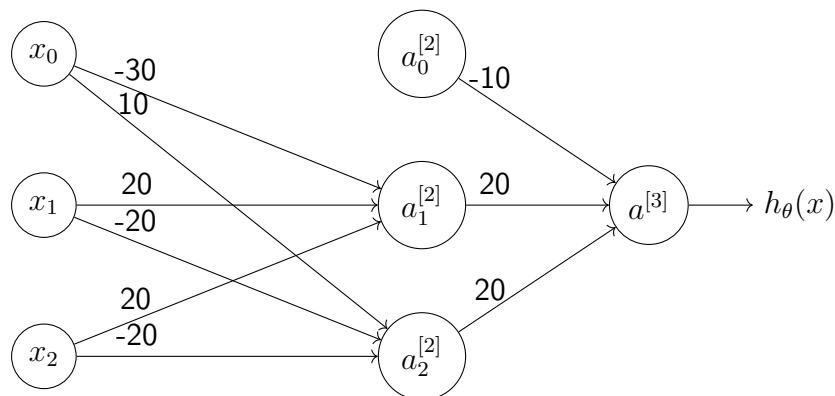


Figura 4: Rede Neural XOR com ativação sigmoide.

$x_1$	$x_2$	$a_1^{[2]}$	$a_2^{[2]}$	$a^{[3]}$	XOR
0	0	0	1	1	1
0	1	0	0	0	0
1	0	0	0	0	0
1	1	1	0	1	1

Tabela 1: Tabela verdade da rede neural XOR.

## Exercícios

1. Qual é o papel das camadas intermediárias na aprendizagem de superfícies de decisão não lineares? Explique a importância da função de ativação não linear.
2. Crie uma rede neural para prever os preços de imóveis em uma cidade. Qual é a função de ativação da saída da rede?
3. Por quê não se deve utilizar a função identidade,  $\sigma(z) = z$ , nas camadas intermediárias da rede?
4. Qual a função de custo você usaria para treinar uma rede neural com um neurônio na camada de saída e ativação sigmoide?