

# Algoritmo backpropagation

Prof. Daniel R. Cassar

ATP-303 - Redes Neurais e Algoritmos Genéticos

# O grafo computacional

$$y = a + b$$

# O grafo computacional

$$y = a + b$$

$$y = a \cdot x - b$$

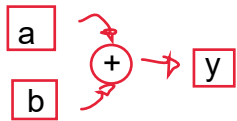
# O grafo computacional

$$y = a + b$$

$$y = a \cdot x - b$$

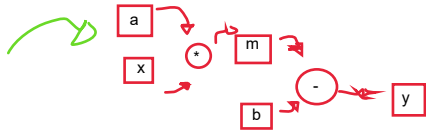
$$y = a^2$$

# O grafo computacional



$$y = a + b$$

$$y = a \cdot x - b$$



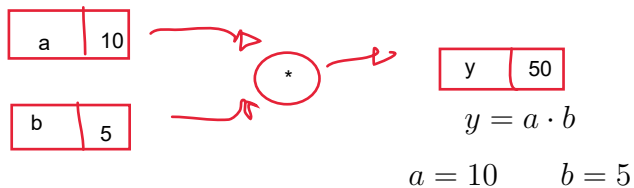
$$y = a^2$$



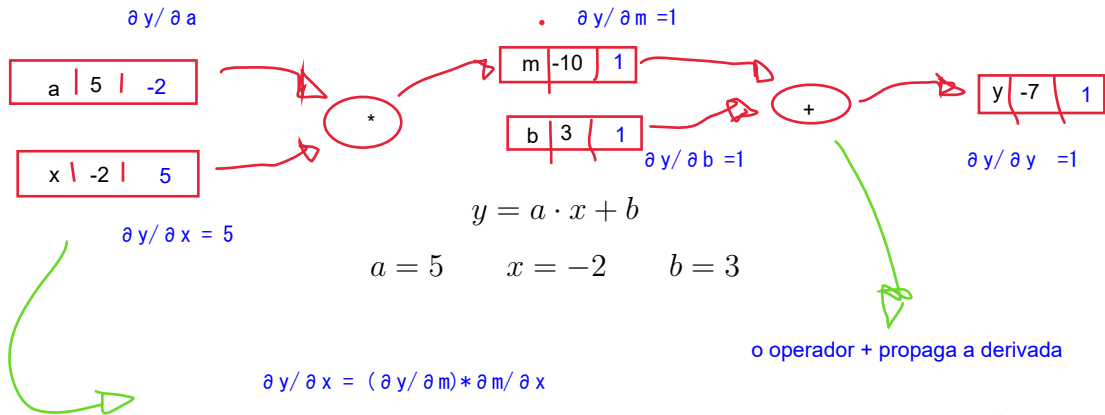
$$y = \exp(a)$$



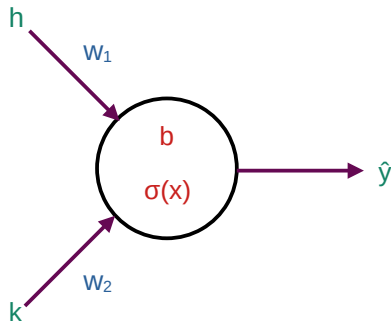
# Forward pass



# Forward pass e backpropagation



# Equação de um neurônio artificial (revisão)



$$\hat{y} = \sigma(hw_1 + kw_2 + b)$$



# A função de ativação identidade

$$\hat{y} = \sigma(h \cdot w_1 + k \cdot w_2 + b)$$

$$\sigma(x) = x$$

$$\hat{y} = h \cdot w_1 + k \cdot w_2 + b$$

# A função de perda

$$\hat{y} = h \cdot w_1 + k \cdot w_2 + b$$

$$L(y, \hat{y}) = (y - \hat{y})^2$$

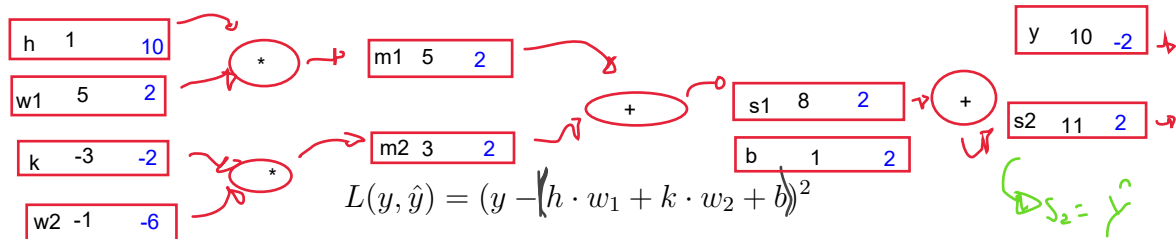
Real



$$L(y, \hat{y}) = (y - h \cdot w_1 + k \cdot w_2 + b)^2$$

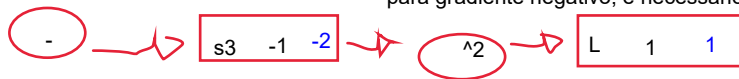
→ modelo

# Backpropagation da função de perda de um neurônio artificial



$$h = 1 \quad k = -3 \quad y = 10 \quad w_1 = 5 \quad w_2 = -1 \quad b = 3$$

para gradiente positivo, é necessário reduzir o valor peso para reduzir o loss  
para gradiente negativo, é necessário aumentar o valor peso para reduzir o loss



se eu mudar algo que possui uma derivada muito grande, a variação final também será muito grande.  
Caso a mudança seja em um peso que possui derivada muito pequena, a variação final tbm será pequena