Introdução a Redes Neurais

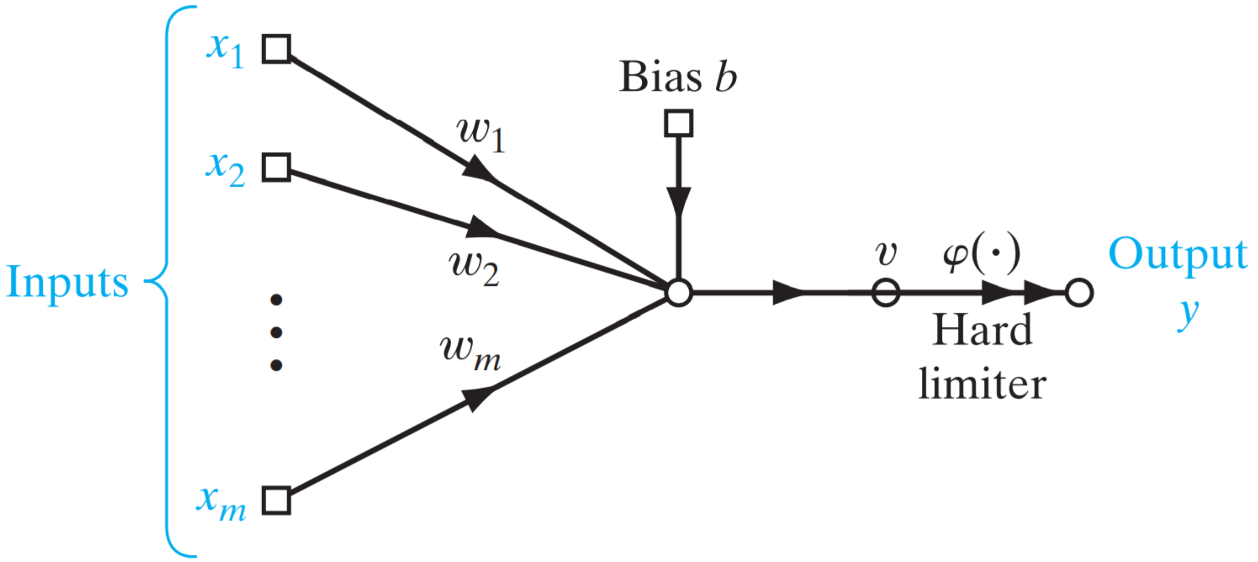
Marlon Sproesser Mathias

Aula 2 – Parte 1

- Perceptrons
- Classificação binária
- Gradientes descendentes
- Trabalho 1

Perceptrons

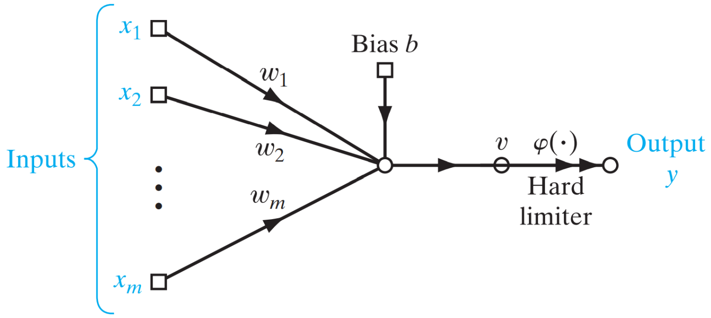
- Unidade de classificação
- Perceptron de Rosenblatt (1958)
- Cria uma hipersuperfície dentro do espaço das entradas

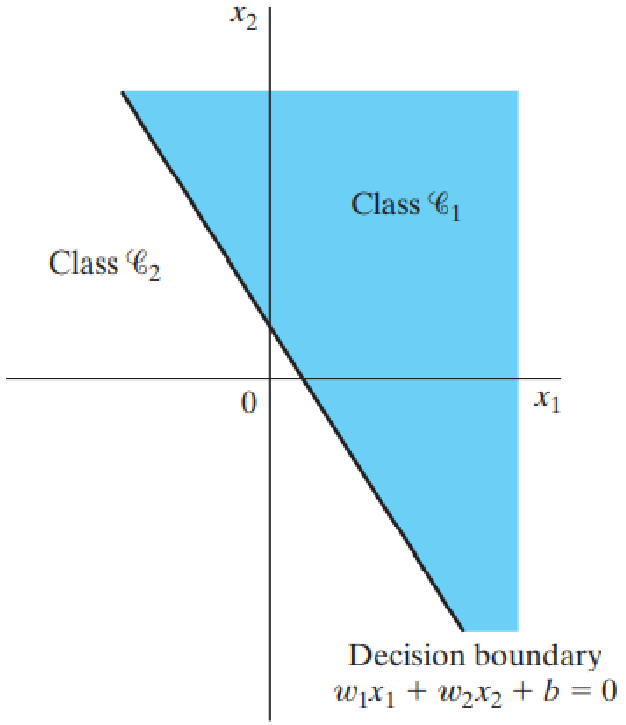


Haykin, S. Neural Networks and Learning Machines - 2009

Perceptrons

- Unidade de classificação
- Perceptron de Rosenblatt (1958)
- Cria uma hipersuperfície dentro do espaço das entradas
- Na prática, separa as saídas em duas classes

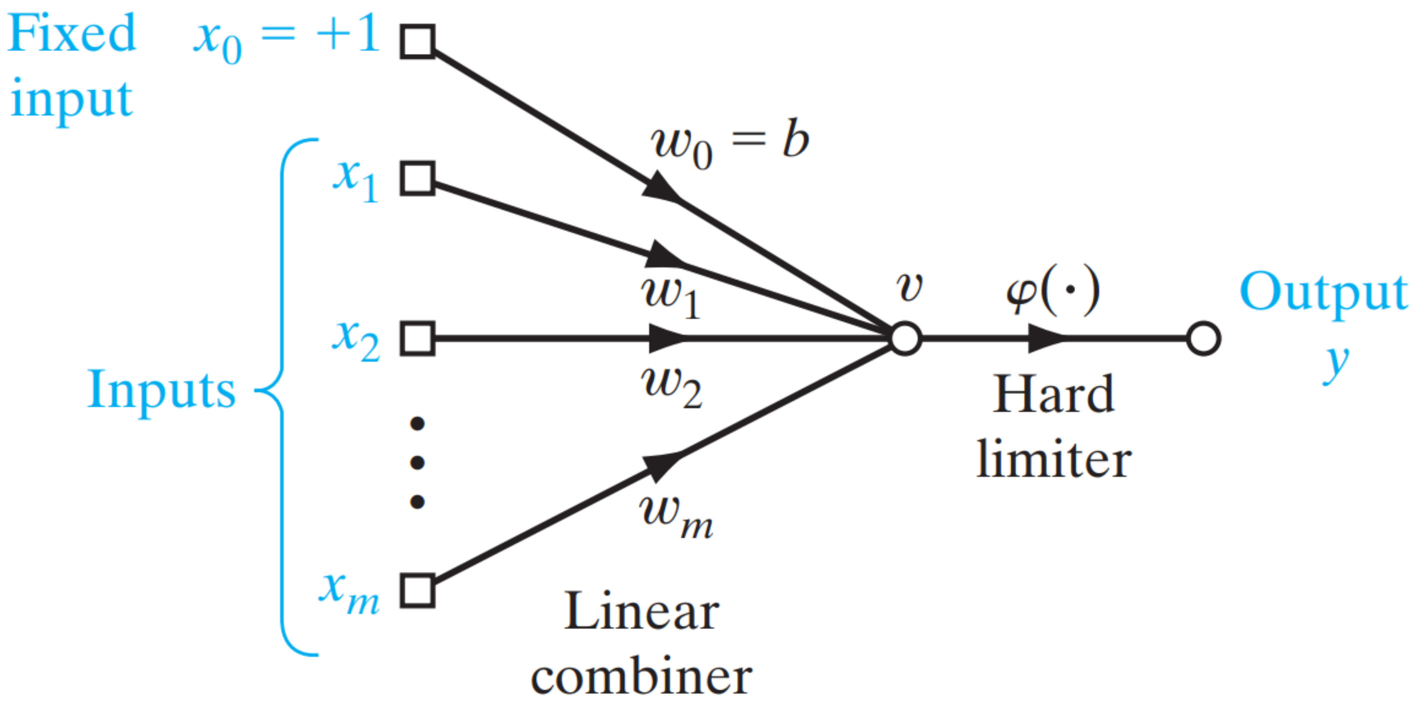




Haykin, S. Neural Networks and Learning Machines - 2009

Perceptrons

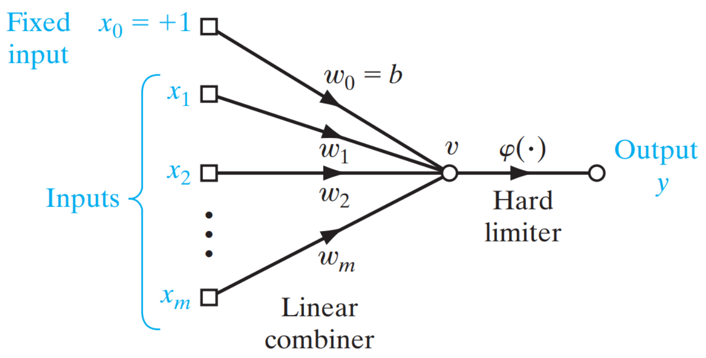
- Quais variáveis escolhemos?

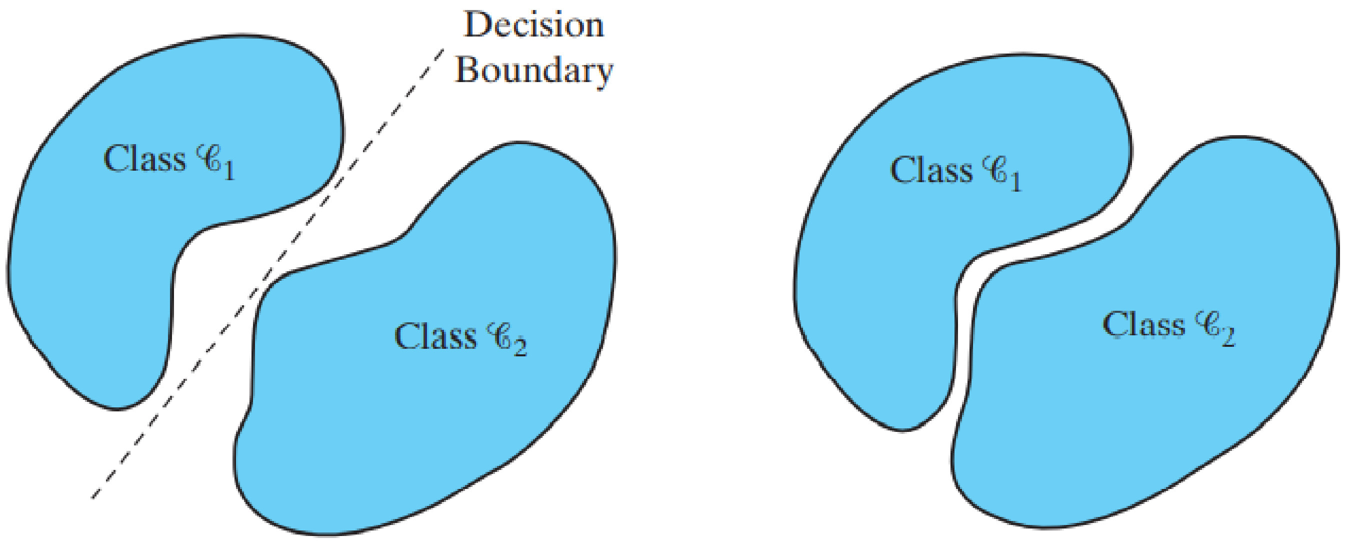


Haykin, S. Neural Networks and Learning Machines - 2009

Perceptrons

- Quais variáveis escolhemos?
- Limitado a separações mais simples → Apenas lineares



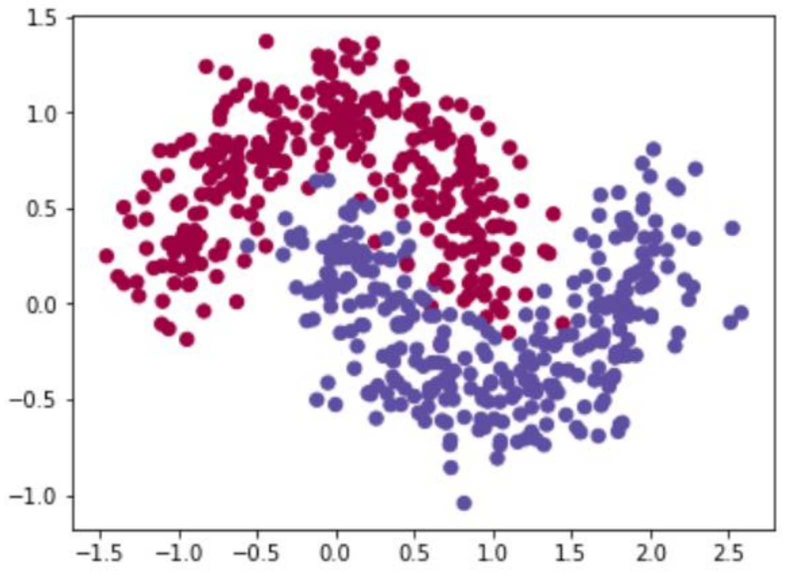


Haykin, S. Neural Networks and Learning Machines - 2009

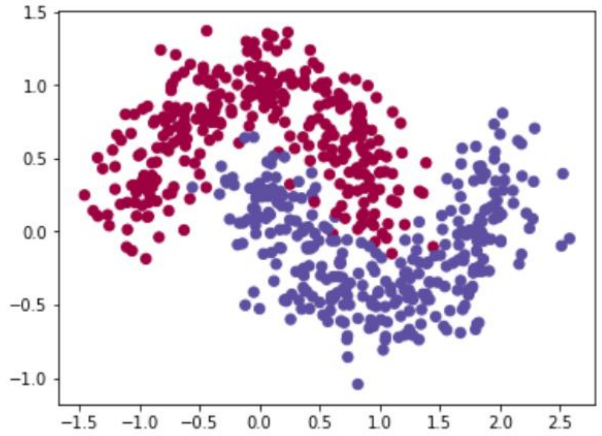
Classificação binária

- Separar dados em dois grupos
- Treinamento supervisionado → Já temos alguns exemplos

- Dados pertencem a dois grupos distintos
- Cada ponto tem coordenadas (x_1, x_2)
- Como separamos os grupos?
- Como treinamos uma RNA para separar os grupos?



Classificação binária



Coordenadas
(x_1, x_2)

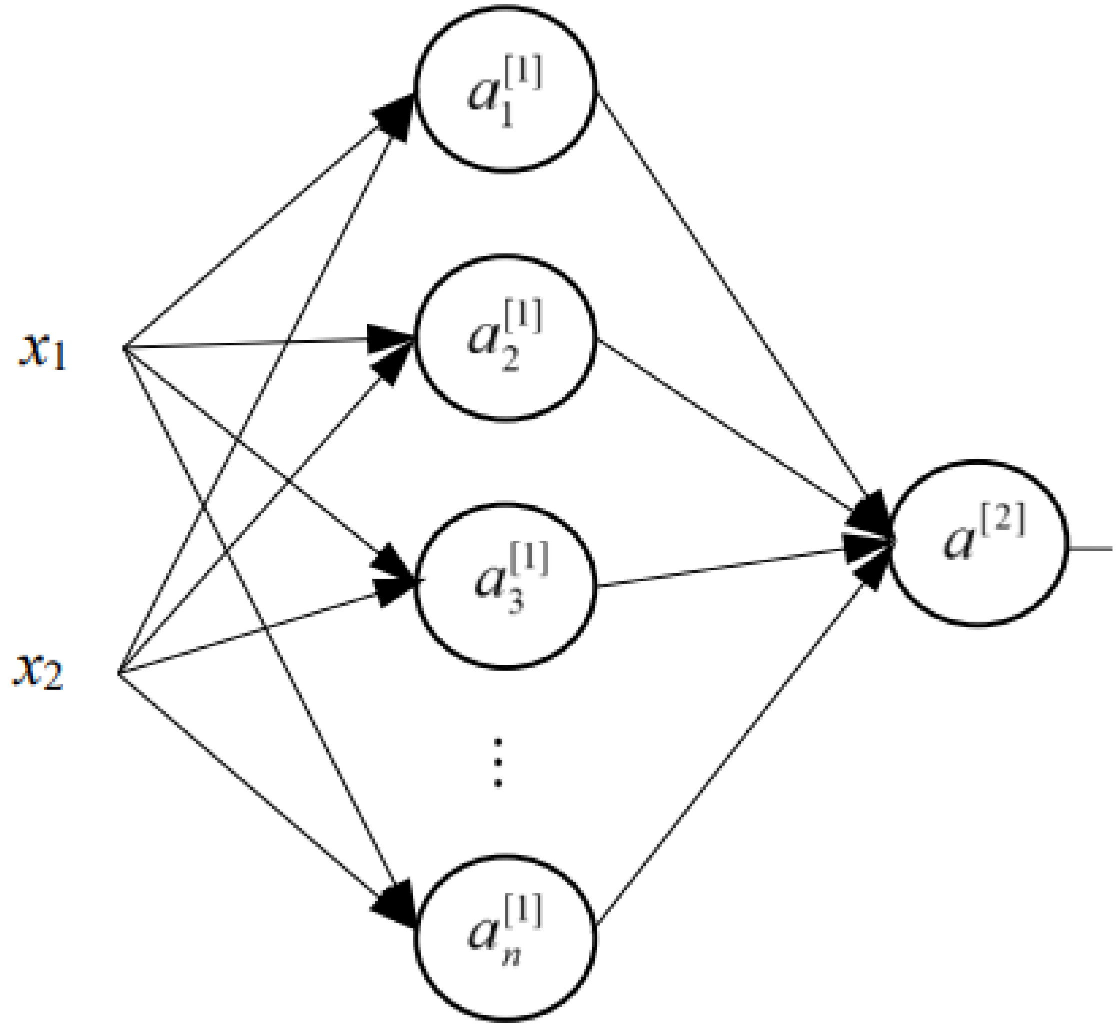
Classe
(y)

0 → Azul

1 → Vermelha

Estrutura da RNA

- Entrada → Vetor $\boldsymbol{x}^{(\boldsymbol{i})} \in \mathbb{R}^2$
- Saída → $\boldsymbol{y}^{(\boldsymbol{i})} \in \mathbb{R}$
- Camada intermediária
  - $n^{[1]}$ neurônios
  - Ativação de tangente hiperbólica
- Camada de saída
  - $n^{[2]} = 1$ neurônio
  - Ativação de sigmoide

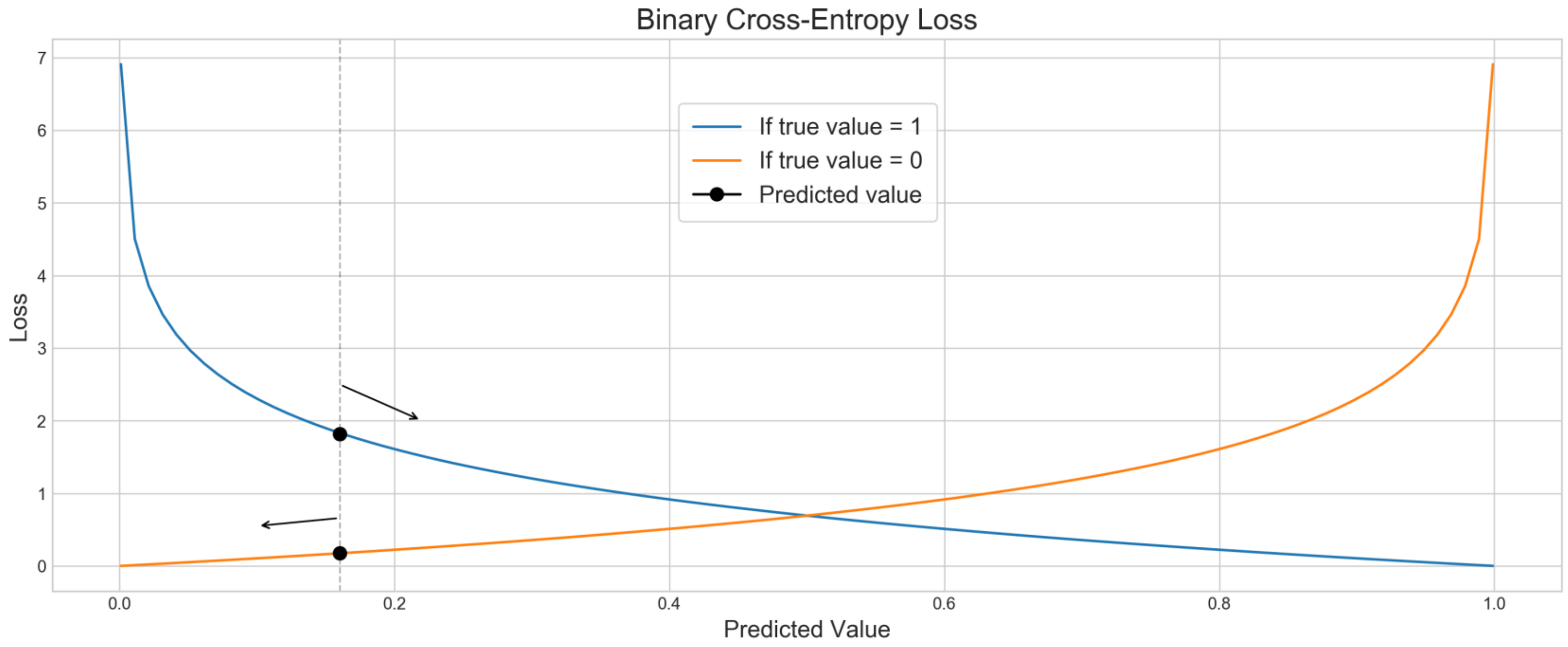


Eduardo Lobo Lustosa Cabral

Função de custo

$$L(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \log \hat{y}^{(i)} - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

- Como comparar a saída da RNA com a "verdade"
- $\hat{y}^{(i)}$ é a previsão da rede no ponto i
- $y^{(i)}$ é o valor de referência no ponto i
- Função de erro logística

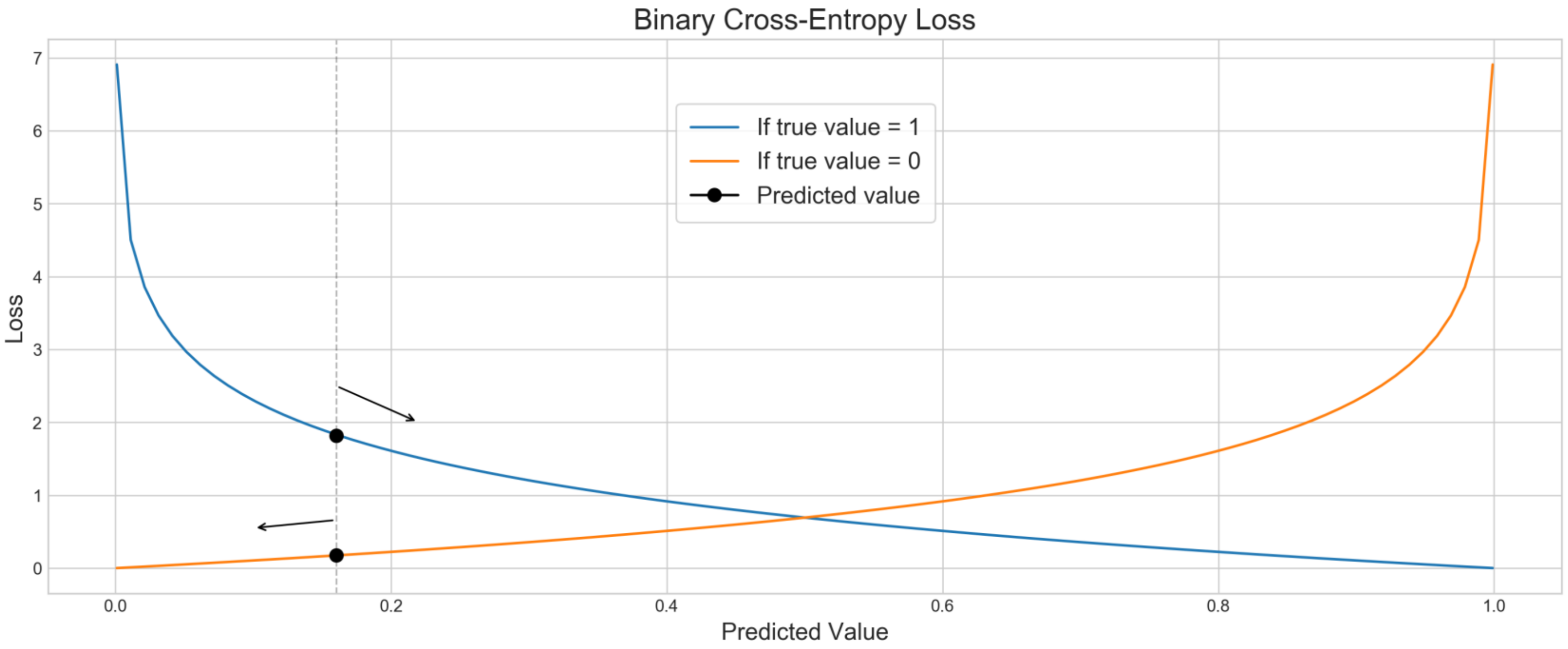


https://towardsdatascience.com/logistic-regression-from-scratch-69db4f587e17

Função de custo

$$L(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \log \hat{y}^{(i)} - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

- Como comparar a saída da RNA com a "verdade"
- $\hat{y}^{(i)}$ é a previsão da rede no ponto i
- $y^{(i)}$ é o valor de referência no ponto i
- Função de erro logística



https://towardsdatascience.com/logistic-regression-from-scratch-69db4f587e17

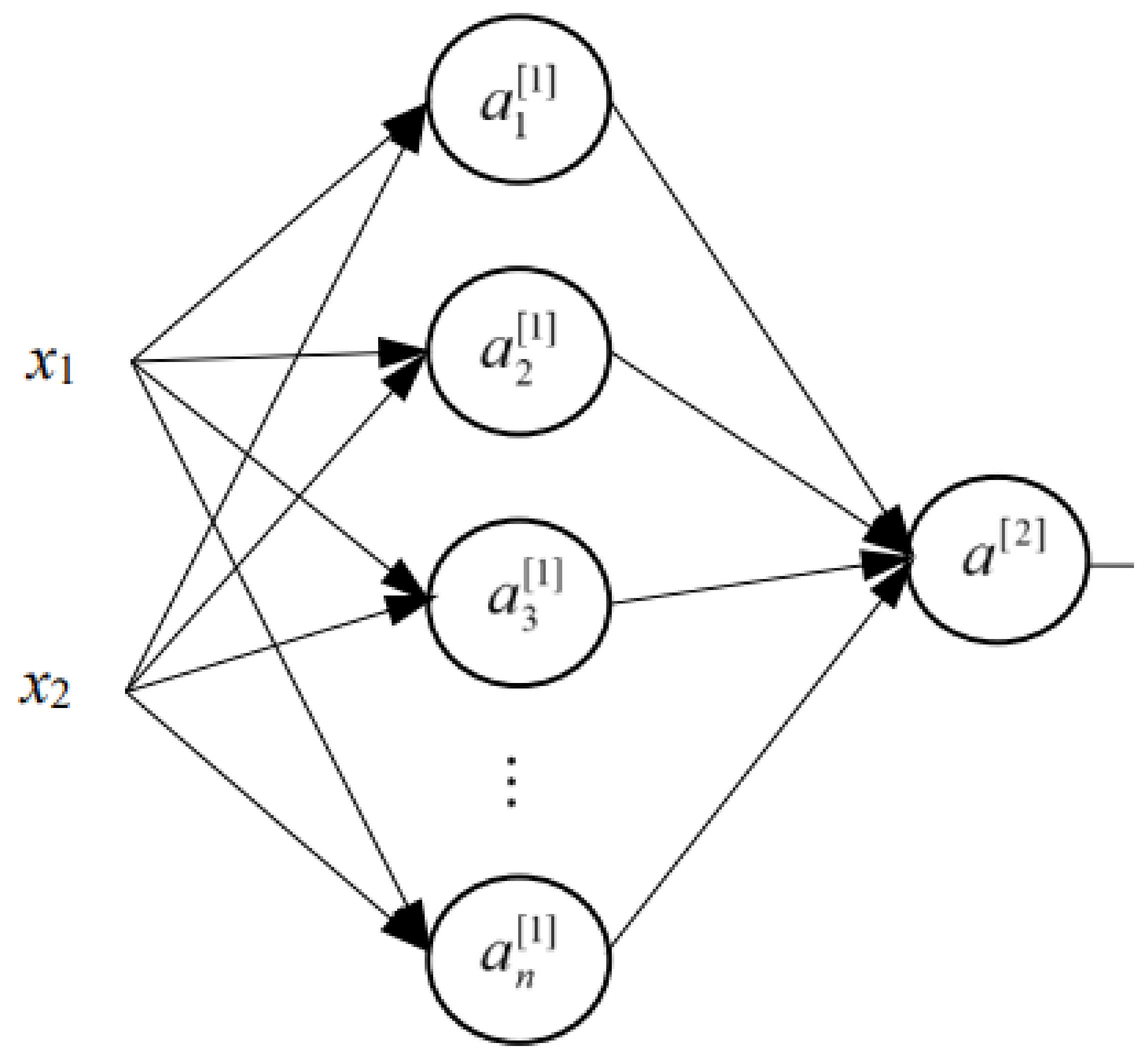
Aplicando a todos os pontos de treino:

$$J(\mathbf{W},\mathbf{B}) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})\right]$$

Propagação para frente

- $z_1^{[1]} = \boldsymbol{W}_{1,1}^{[1]} x_1 + \boldsymbol{W}_{1,2}^{[1]} x_2 + b_1^{[1]}$

- Ou, de forma matricial:
- $\boldsymbol{z}^{[1]} = \boldsymbol{W}^{[1]}\boldsymbol{x} + \boldsymbol{b}^{[1]}$
- $\boldsymbol{a}^{[1]} = \boldsymbol{g}^{[1]}(\boldsymbol{z}^{[1]})$ ← Tangente hiperbólica
- $\boldsymbol{z}^{[2]} = \boldsymbol{W}^{[2]}\boldsymbol{a}^{[1]} + \boldsymbol{b}^{[2]}$
- $\boldsymbol{a}^{[2]} = \boldsymbol{g}^{[2]}(\boldsymbol{z}^{[2]})$ ← Sigmoide
- $\hat{y} = \boldsymbol{a}^{[2]}$



Propagação para frente

$$\boldsymbol{z}^{[1]} = \boldsymbol{W}^{[1]}\boldsymbol{x} + \boldsymbol{b}^{[1]}$$

$$\boldsymbol{a}^{[1]} = \boldsymbol{g}^{[1]}\left(\boldsymbol{z}^{[1]}\right)$$

$$\boldsymbol{z}^{[2]} = \boldsymbol{W}^{[2]}\boldsymbol{a}^{[1]} + \boldsymbol{b}^{[2]}$$

$$\boldsymbol{a}^{[2]} = \boldsymbol{g}^{[2]}\left(\boldsymbol{z}^{[2]}\right)$$

$$\hat{y} = \boldsymbol{a}^{[2]}$$

Variável	Linhas	Colunas
x		
$z^{[1]}$		
$W^{[1]}$		
$b^{[1]}$		
$a^{[1]}$		
$z^{[2]}$		
$W^{[2]}$		
$b^{[2]}$		
$a^{[2]}$		

Propagação para frente

$$\boldsymbol{z}^{[1]} = \boldsymbol{W}^{[1]}\boldsymbol{x} + \boldsymbol{b}^{[1]}$$

$$\boldsymbol{a}^{[1]} = \boldsymbol{g}^{[1]}(\boldsymbol{z}^{[1]})$$

$$\boldsymbol{z}^{[2]} = \boldsymbol{W}^{[2]}\boldsymbol{a}^{[1]} + \boldsymbol{b}^{[2]}$$

$$\boldsymbol{a}^{[2]} = \boldsymbol{g}^{[2]}(\boldsymbol{z}^{[2]})$$

$$\hat{y} = \boldsymbol{a}^{[2]}$$

Variável	Linhas	Colunas
x	2	n_s
$W^{[1]}$	n_h	2
$b^{[1]}$	n_h	1
$z^{[1]}$	n_h	n_s
$a^{[1]}$	n_h	n_s
$W^{[2]}$	1	n_h
$b^{[2]}$	1	1
$z^{[2]}$	1	n_s
$a^{[2]}$	1	n_s

Como treinar a RNA?

Devemos encontrar os valores de W e b que minimizam o erro J

$$\boldsymbol{z}^{[1]} = \boldsymbol{W}^{[1]}\boldsymbol{x} + \boldsymbol{b}^{[1]}$$

$$\boldsymbol{a}^{[1]} = \boldsymbol{g}^{[1]}\left(\boldsymbol{z}^{[1]}\right)$$

$$\boldsymbol{z}^{[2]} = \boldsymbol{W}^{[2]}\boldsymbol{a}^{[1]} + \boldsymbol{b}^{[2]}$$

$$\boldsymbol{a}^{[2]} = \boldsymbol{g}^{[2]}\left(\boldsymbol{z}^{[2]}\right)$$

$$\hat{y} = \boldsymbol{a}^{[2]}$$

$$J(\mathbf{W},\mathbf{B}) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\hat{y}^{(i)} + (1-y^{(i)})\log(1-\hat{y}^{(i)})\right]$$

Variável	Linhas	Colunas
x	2	n_s
$W^{[1]}$	n_h	2
$b^{[1]}$	n_h	1
$z^{[1]}$	n_h	n_s
$a^{[1]}$	n_h	n_s
$W^{[2]}$	1	n_h
$b^{[2]}$	1	1
$z^{[2]}$	1	n_s
$a^{[2]}$	1	n_s

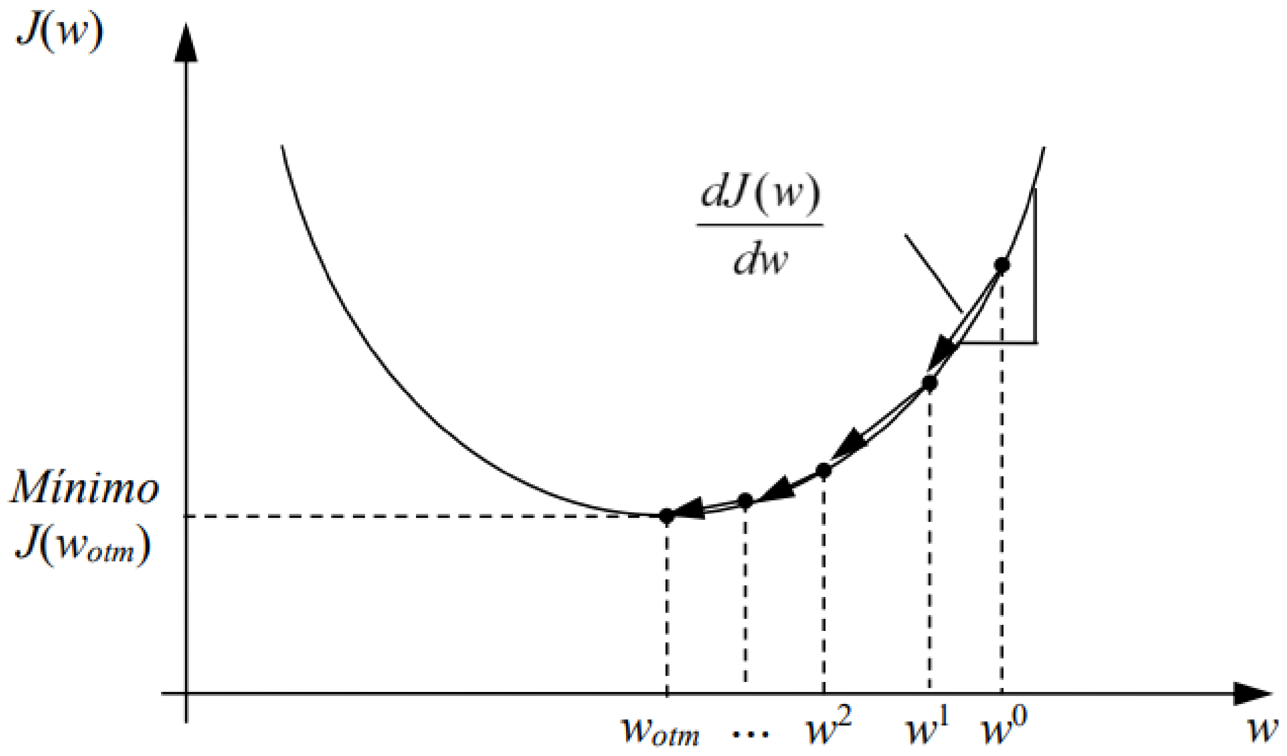
Gradiente descendente

- Problema de otimização
- Minimizar uma função

- Queremos achar W que minimiza J

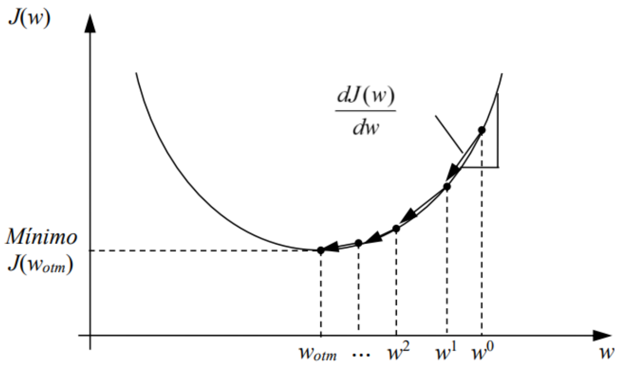
$$\frac{\partial J}{\partial W}$$

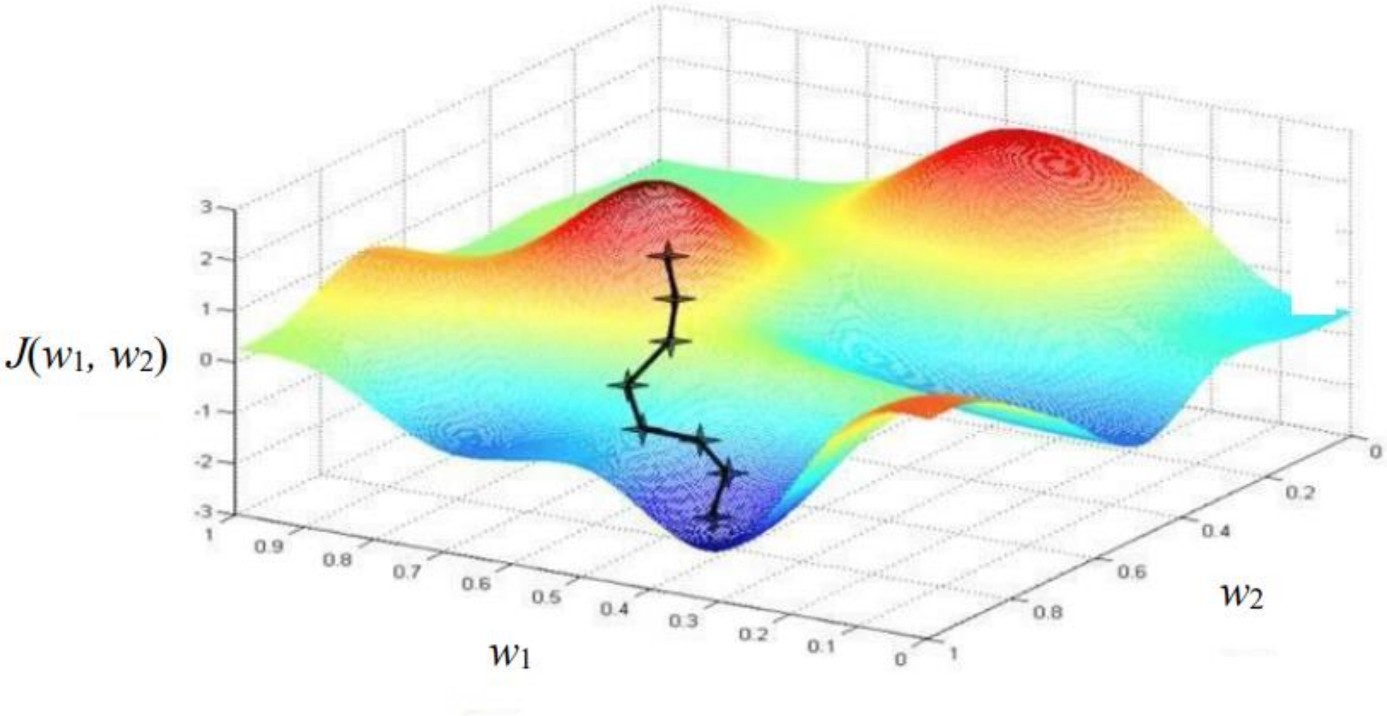
Qual o formato dessa derivada?



Gradiente descendente

- Problema de otimização
- Minimizar uma função



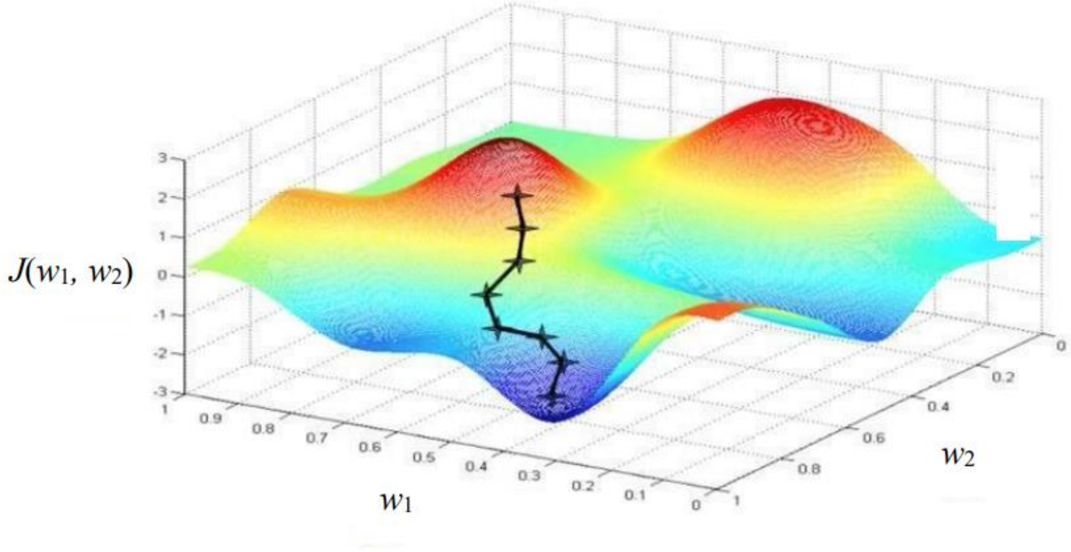


https://github.com/RNogales94/ISI-Tensorflow

Gradiente descendente

- Precisamos do gradiente do custo em relação a cada um dos pesos

- Aqui entra o *backpropagation*



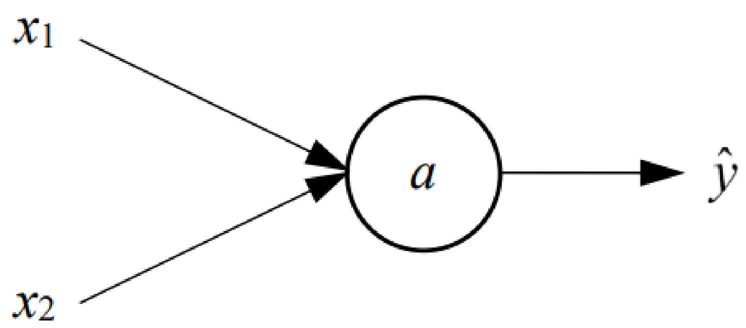
https://github.com/RNogales94/ISI-Tensorflow

$$w_{k,j}^{[l]} = w_{k,j}^{[l]} - \alpha \frac{\partial J(\mathbf{W},\mathbf{B})}{\partial w_{k,j}^{[l]}}, \text{ para } l = 1,\dots,L,\ k = 1,..,n^{[l]},\ j = 1,..,n^{[l-1]}$$

$$b_k^{[l]} = b_k^{[l]} - \alpha \frac{\partial J(\mathbf{W},\mathbf{B})}{\partial b_k^{[l]}}, \text{ para } l = 1,\dots,L,\ k = 1,..,n^{[l]}$$

Exemplo em uma RNA de um neurônio

- 2 entradas
- 1 saída
- m exemplos de treinamento
- 1 camada
- 1 neurônio



Eduardo Lobo Lustosa Cabral

Exemplo em uma RNA de um neurônio

Propagação para frente

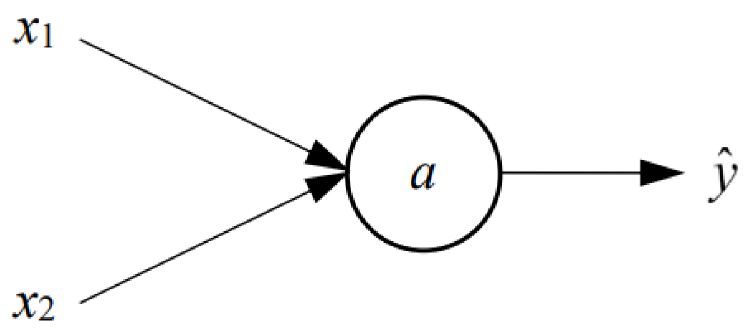
- $z = w_1 x_1 + w_2 x_2 + b$
- $\hat{y} = a = g(z)$

Parâmetros:

- w_1, w_2, b

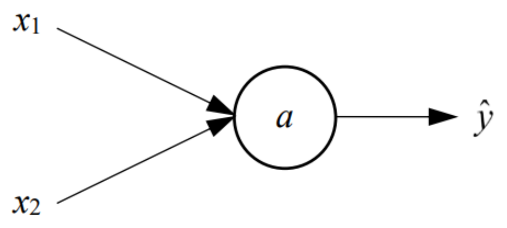
Custo:

- $J(w_1, w_2, b) = \frac{1}{m} \sum_{i=1}^{m} E\left(\hat{y}^{(i)}, y^{(i)}\right)$



Eduardo Lobo Lustosa Cabral

Exemplo em uma RNA de um neurônio



Propagação para frente

- $z = w_1 x_1 + w_2 x_2 + b$
- $\hat{y} = a = g(z)$

Parâmetros:

- w_1, w_2, b

Custo:

- $J(w_1, w_2, b) = \frac{1}{m} \sum_{i=1}^{m} E\left(\hat{y}^{(i)}, y^{(i)}\right)$

$\left[\frac{\partial E(a,y)}{\partial a}\right]$ = derivada da função de erro em relação à saída calculada ($\hat{y} = a$);

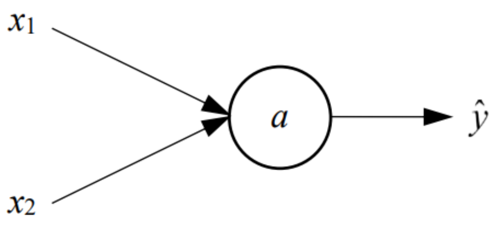
$\frac{\partial a}{\partial z} = \frac{d[g(z)]}{dz} \Rightarrow$ derivada da função de ativação g

$$\frac{\partial z}{\partial w_1} = x_1$$

$$\frac{\partial z}{\partial w_2} = x_2$$

$$\frac{\partial z}{\partial b} = 1$$

Exemplo em uma RNA de um neurônio



$\left[\dfrac{\partial E(a,y)}{\partial a}\right]$ = derivada da função de erro em relação à saída calculada ($\hat{y} = a$);

$\dfrac{\partial a}{\partial z} = \dfrac{d[g(z)]}{dz} \Rightarrow$ derivada da função de ativação g

$$\frac{\partial z}{\partial w_1} = x_1$$

$$\frac{\partial z}{\partial w_2} = x_2$$

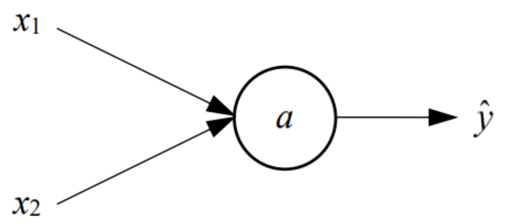
$$\frac{\partial z}{\partial b} = 1$$

$$\boxed{\frac{\partial E(a,y)}{\partial w_1} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right] x_1}$$

$$\boxed{\frac{\partial E(a,y)}{\partial w_2} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right] x_2}$$

$$\boxed{\frac{\partial E(a,y)}{\partial b} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right]}$$

Exemplo em uma RNA de um neurônio



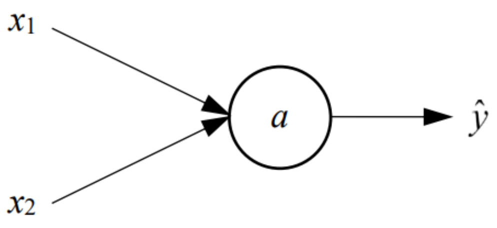
$$\boxed{\frac{\partial E(a,y)}{\partial w_1} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right]x_1}$$

Para função logística

$$\frac{\partial E(a,y)}{\partial a} = \begin{cases} y = 0 \rightarrow & -1/(1-a) \\ y = 1 \rightarrow & -1/a \end{cases}$$

$$L(\hat{y}^{(i)}, y^{(i)}) = -y^{(i)} \log \hat{y}^{(i)} - (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

Exemplo em uma RNA de um neurônio

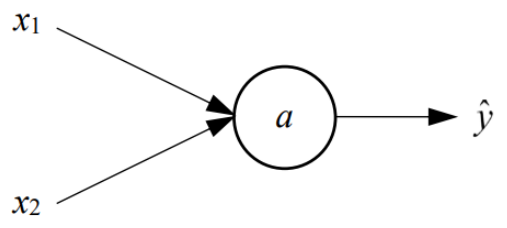


$$\boxed{\frac{\partial E(a,y)}{\partial w_1} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right]x_1}$$

Para tangente hiperbólica

$$\frac{dg(z)}{dz} = \frac{d}{dz}\left[\frac{e^z - e^{-z}}{e^z + e^{-z}}\right] = \frac{(e^z + e^{-z})^2 - (e^z - e^{-z})^2}{(e^z + e^{-z})^2} = \underbrace{\frac{(e^z + e^{-z})^2}{(e^z + e^{-z})^2}}_{1} - \underbrace{\left[\frac{(e^z - e^{-z})}{(e^z + e^{-z})}\right]}_{\tanh(z)}^2 = 1 - a^2$$

Exemplo em uma RNA de um neurônio

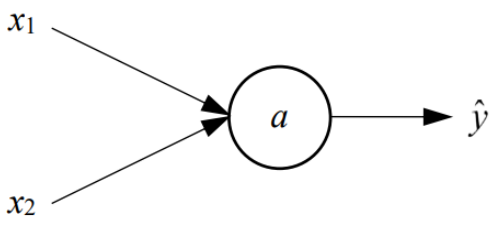


$$\boxed{\frac{\partial E(a,y)}{\partial w_1} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right]x_1}$$

Para sigmoide

$$\frac{d\sigma(z)}{dz} = \frac{d}{dz}\left[\frac{1}{1+e^{-z}}\right] = \frac{e^{-z}}{(1+e^{-z})^2} = \underbrace{\frac{1}{(1+e^{-z})}}_{a}\underbrace{\left[1-\frac{1}{(1+e^{-z})}\right]}_{(1-a)} = a(1-a)$$

Exemplo em uma RNA de um neurônio



$$\boxed{\frac{\partial E(a,y)}{\partial w_1} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right]x_1}$$

$$\boxed{\frac{\partial E(a,y)}{\partial w_2} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right]x_2}$$

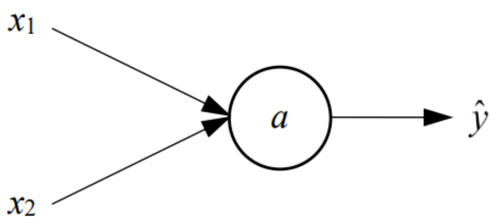
$$\boxed{\frac{\partial E(a,y)}{\partial b} = \left[\frac{\partial E(a,y)}{\partial a}\right]\left[\frac{d[g(z)]}{\partial z}\right]}$$

Lembrando que:

$$J(w_1, w_2, b) = \frac{1}{m}\sum_{i=1}^{m} E\left(\hat{y}^{(i)}, y^{(i)}\right)$$

Podemos somar $\frac{\partial E}{\partial w_i}$ para cada ponto e obter $\frac{\partial J(w_1,w_2,b)}{\partial w_i}$

Exemplo em uma RNA de um neurônio



Lembrando que:

$$J(w_1, w_2, b) = \frac{1}{m} \sum_{i=1}^{m} E\left(\hat{y}^{(i)}, y^{(i)}\right)$$

Podemos usar o $\frac{\partial E}{\partial w_i}$ de cada ponto e obter

$$\frac{\partial J(w_1, w_2, b)}{\partial w_i} = \frac{1}{m} \sum_{i=1}^{m} \frac{\partial E\left(\hat{y}^{(i)}, y^{(i)}\right)}{\partial w_i}$$

Finalmente, para atualizar os pesos, usamos:

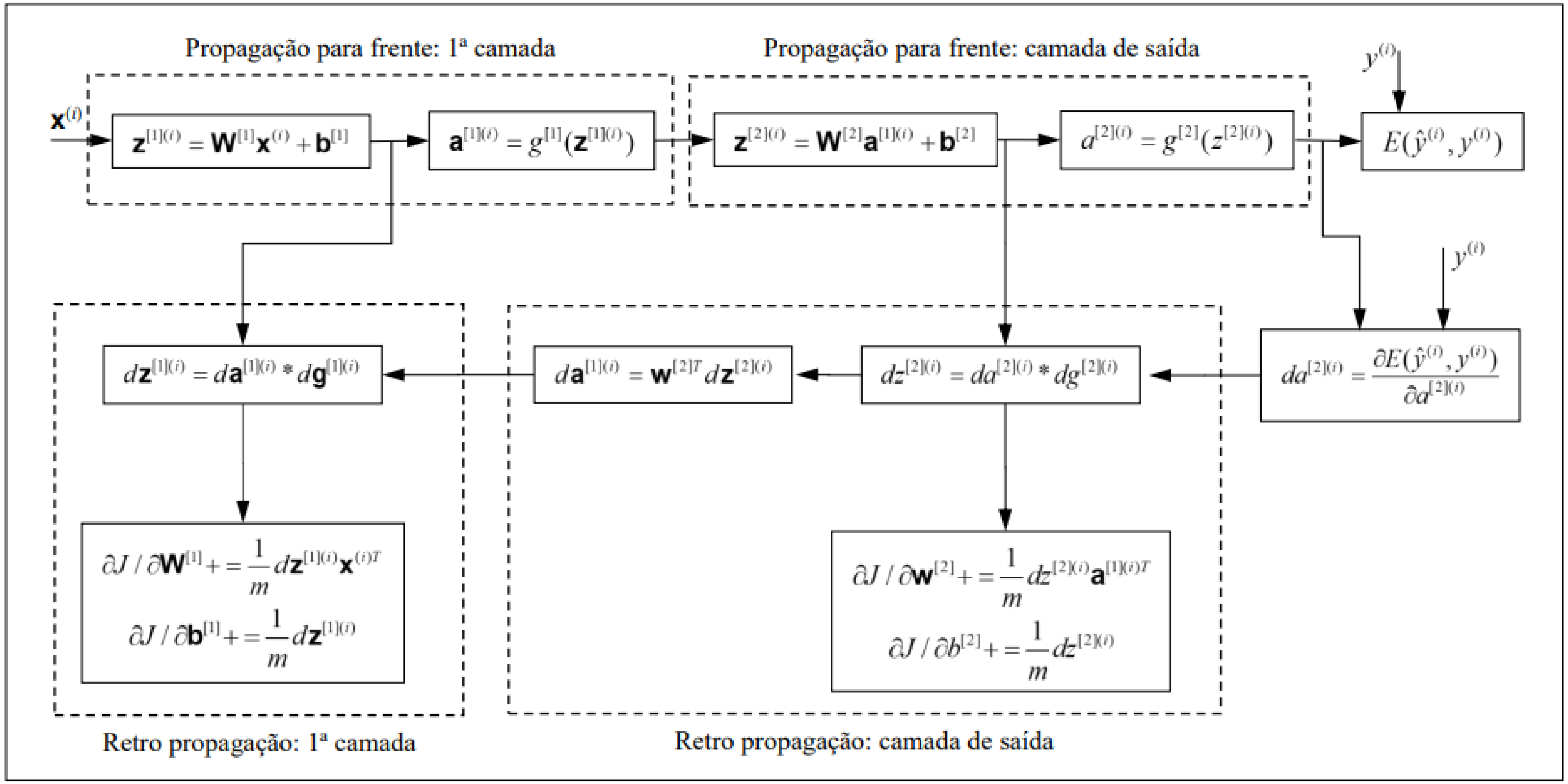
$$w_1 = w_1 - \alpha \frac{\partial J(w_1, w_2, b)}{\partial w_1},$$

$$w_2 = w_2 - \alpha \frac{\partial J(w_1, w_2, b)}{\partial w_2},$$

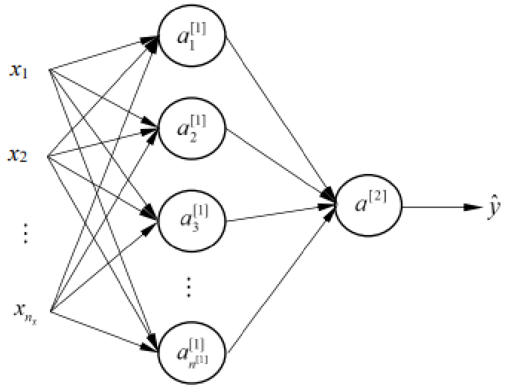
$$b = b - \alpha \frac{\partial J(w_1, w_2, b)}{\partial b}.$$

Onde α é a taxa de aprendizagem

E com mais camadas?



Eduardo Lobo Lustosa Cabral



Trabalho 1

- Classificação binária
- Implementado num notebook de Python
- Sem usar bibliotecas de ML
- Rede neural rasa