FIAP GRADUAÇÃO

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FIAP GRADUAÇÃO

DISRUPTIVE ARCHITECTURES:
IOT, IOB & GENERATIVE AI

Deep Learning e Redes Neurais Artificiais

Prof. André Tritiack
profandre.farias@fiap.com.br

2024

Redes Neurais Artificiais - Histórico

- Em 1943, Warren McCulloch e Water Pitts **modelaram o primeiro neurônio**;
- Inspirado pelo modelo de McCulloch e Pitts, Frank Rosenblatt irá criar o primeiro neurônio artificial, o **Perceptron** em 1958
- Em 1960 surge uma variação do Perceptron, o **ADALINE**, criado por Bernard Widrow e Ted Hoff.
- Perceptrons e rede ADALINE eram feitas através de implementação física, via hardware, isto é, nessa época elas ainda não eram programas de computador.
- Já em 1974, Paul Werbos irá propor um algoritmo de treinamento para as Redes Neurais Artificiais, chamado de **Backpropagation**

Redes Neurais Artificiais - Histórico

- O *backpropagation* foi redescoberto e difundido por David Rumelhart a partir de 1986.

- Baseado nos estudos de David Hubel e Torsten Wiesel (1968) sobre a visão de organismos vivos, Kunihiko Fukushima irá proporá a primeira **Rede Neural Convolucional** chamada de **Neocognitron.**

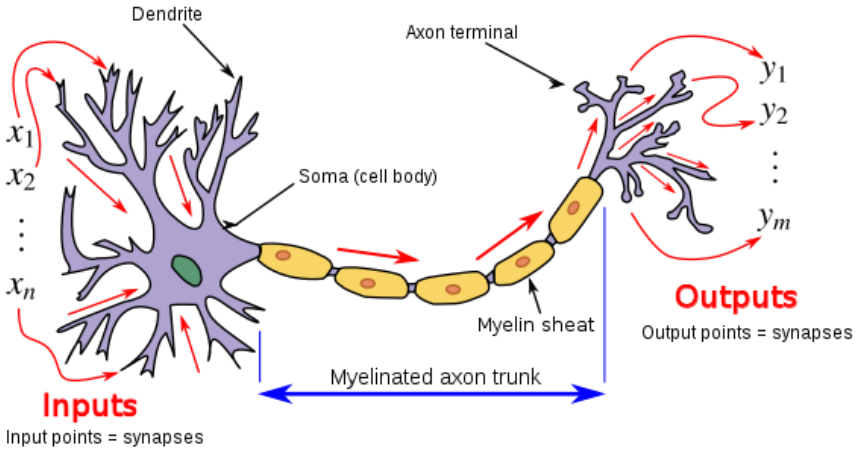
- A primeira **Rede Neural Recorrente** surge em 1997, com os trabalhos de Sepp Hochreiter e Jürgen Schimidhuber;

- A partir dos anos 2000, com muito dinheiro nas empresas de tecnologia do Vale do Silício, as pesquisa em IA e Redes Neurais Artificias sofrem um “boost”.

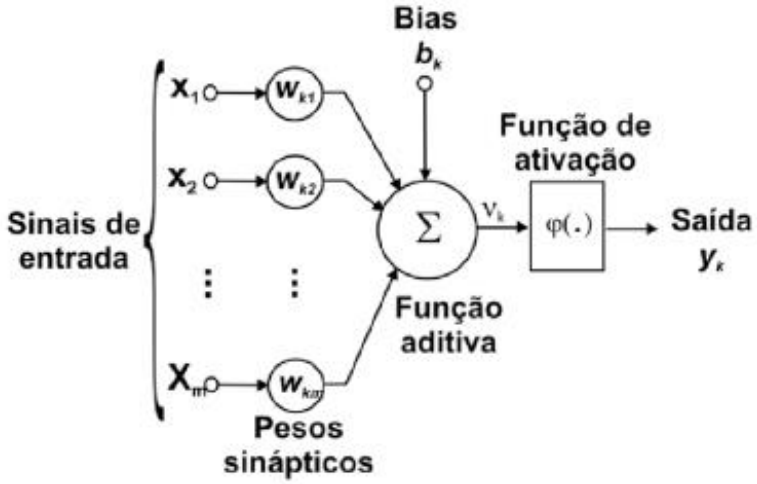
- Hoje, IA e Redes Neurais Artificiais são amplamente aplicadas nas mais variadas áreas do conhecimento e dos negócios.

Perceptron

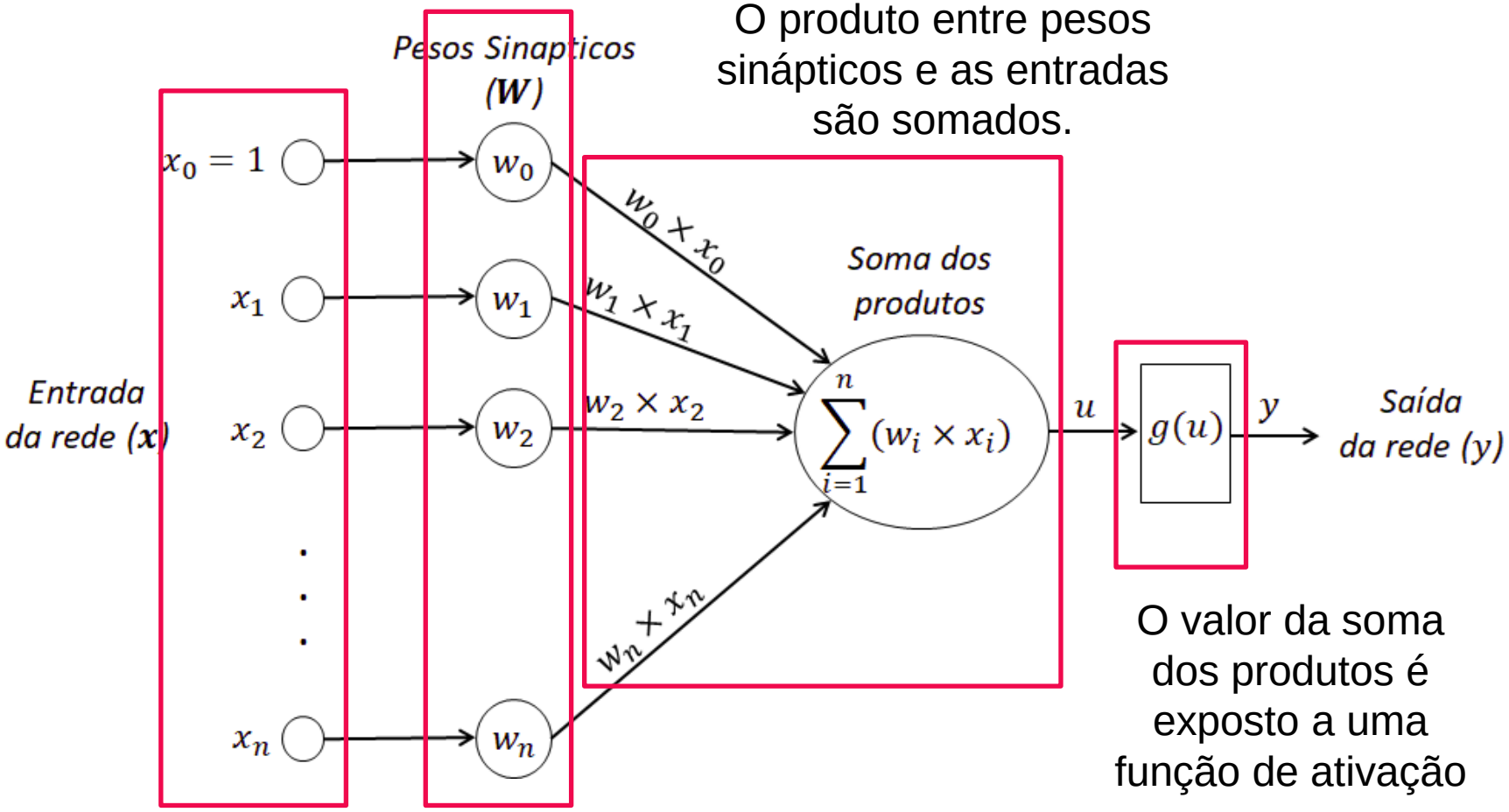
Warren McCulloch e Walter Pitts (1943)



Frank Rosenblatt (1958)



Perceptron

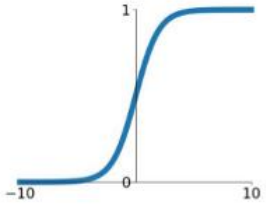

Pesos Sinapticos
(W)
O produto entre pesos sinápticos e as entradas são somados.
$x_0 = 1$
x_1
x_2
x_n
w_0
w_1
w_2
w_n
$w_0 \times x_0$
$w_1 \times x_1$
$w_2 \times x_2$
$w_n \times x_n$
Soma dos produtos
$\sum_{i=1}^{n} (w_i \times x_i)$
u
$g(u)$
y
Entrada da rede (x)
Saída da rede (y)
O valor da soma dos produtos é exposto a uma função de ativação
Atributos de Entrada
Pesos sinápticos são os parâmetros do nosso modelo. Eles que serão aprendidos na fase de treinamento.

Função de Ativação

- Existem muitas funções de ativação usadas em redes neurais artificiais. Para problemas de classificação, em geral, usa-se funções sigmoidais e para problemas de regressão usa-se funções ReLU.

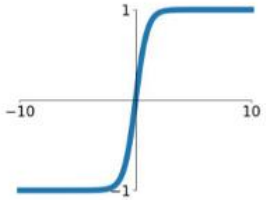
Sigmoid

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



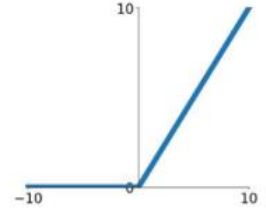
tanh

$$\tanh(x)$$



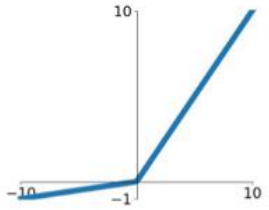
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

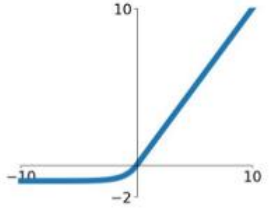


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

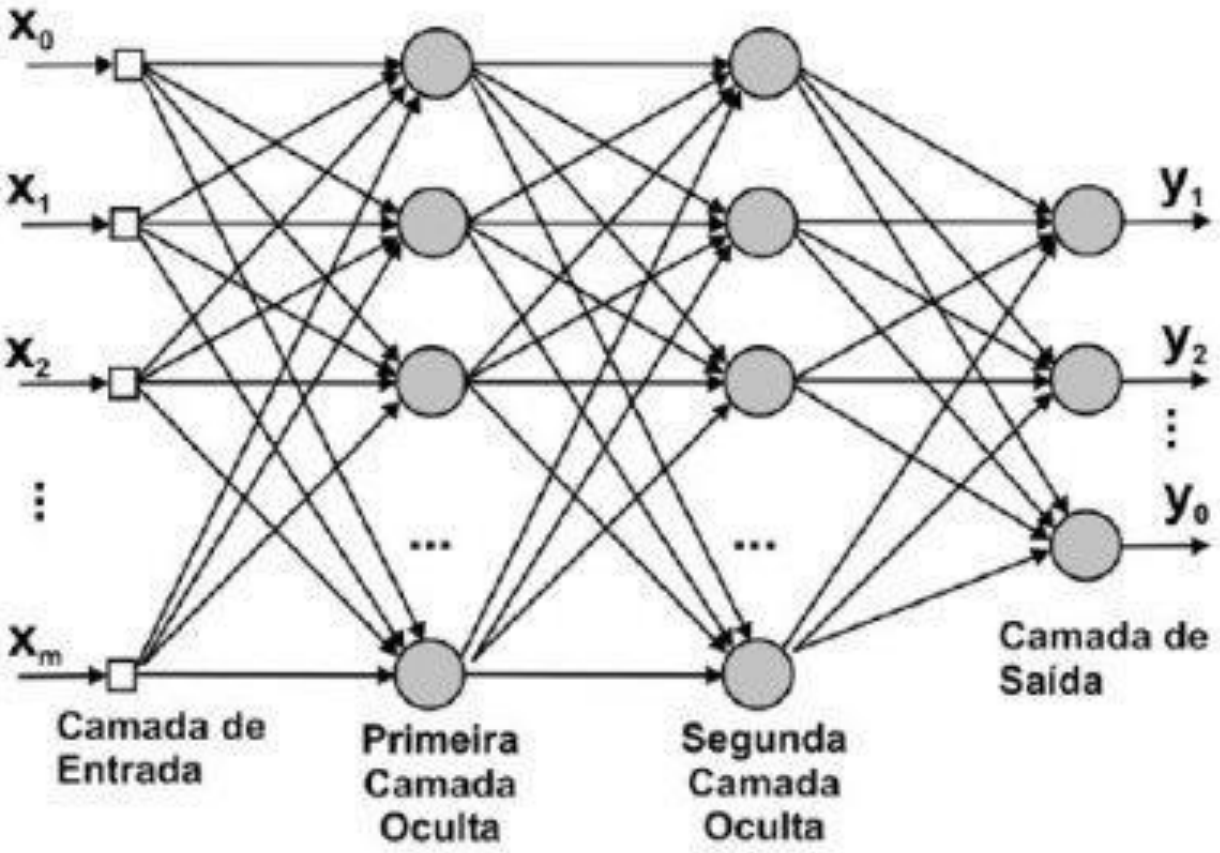
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

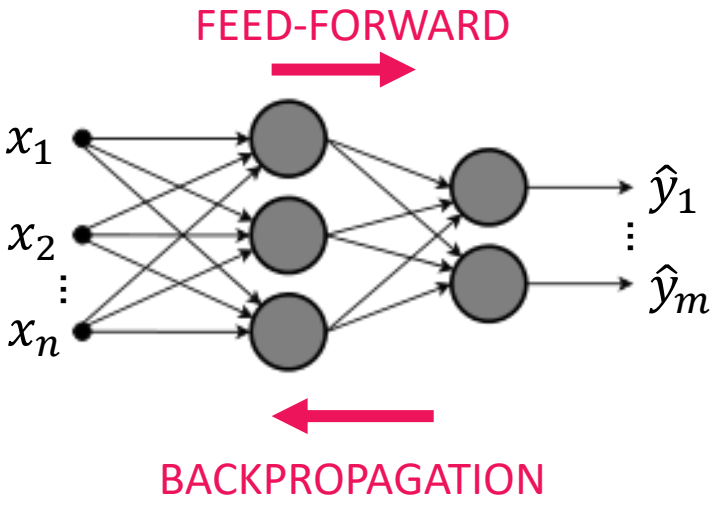


Fonte: https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092

Multilayer Perceptron (MLP)



Treinamento em uma RNA



Atributo alvo observado

Atributos descritivos

Atributo alvo previsto

Entrada	x_1	x_2	...	x_n	y	$\hat{y}$
1	70.52	30	...	0.584	90	100
2	60.96	27	...	1.254	81	90
...	...	...	...	...	...	...
k	97.48	35	...	0.758	122	120

$erro(y, \hat{y})$

Treinamento em uma RNA

- Na etapa de **Feed-Forward**, propagamos a entrada da rede, calculando a saída de cada neurônio, indo da primeira camada até a última camada.

- Comparamos a saída obtida $\hat{y}$ com a saída esperada y.

- Computamos o erro a partir de uma função de erro (como o MSE) para todas as entradas calculadas.

- Na etapa de **Backpropagation**, verificamos qual foi o erro da camada atual e atualizamos os pesos sinápticos da rede para minimizar o erro;

Treinamento em uma RNA

FIAP

- Para ajustar os pesos da rede precisamos medir o quanto estamos errando.
- Para medir isso podemos usar uma função chamada de função custo ou função erro.
- Nosso objetivo será minimizar o erro da rede, logo desejamos minimizar a função custo.

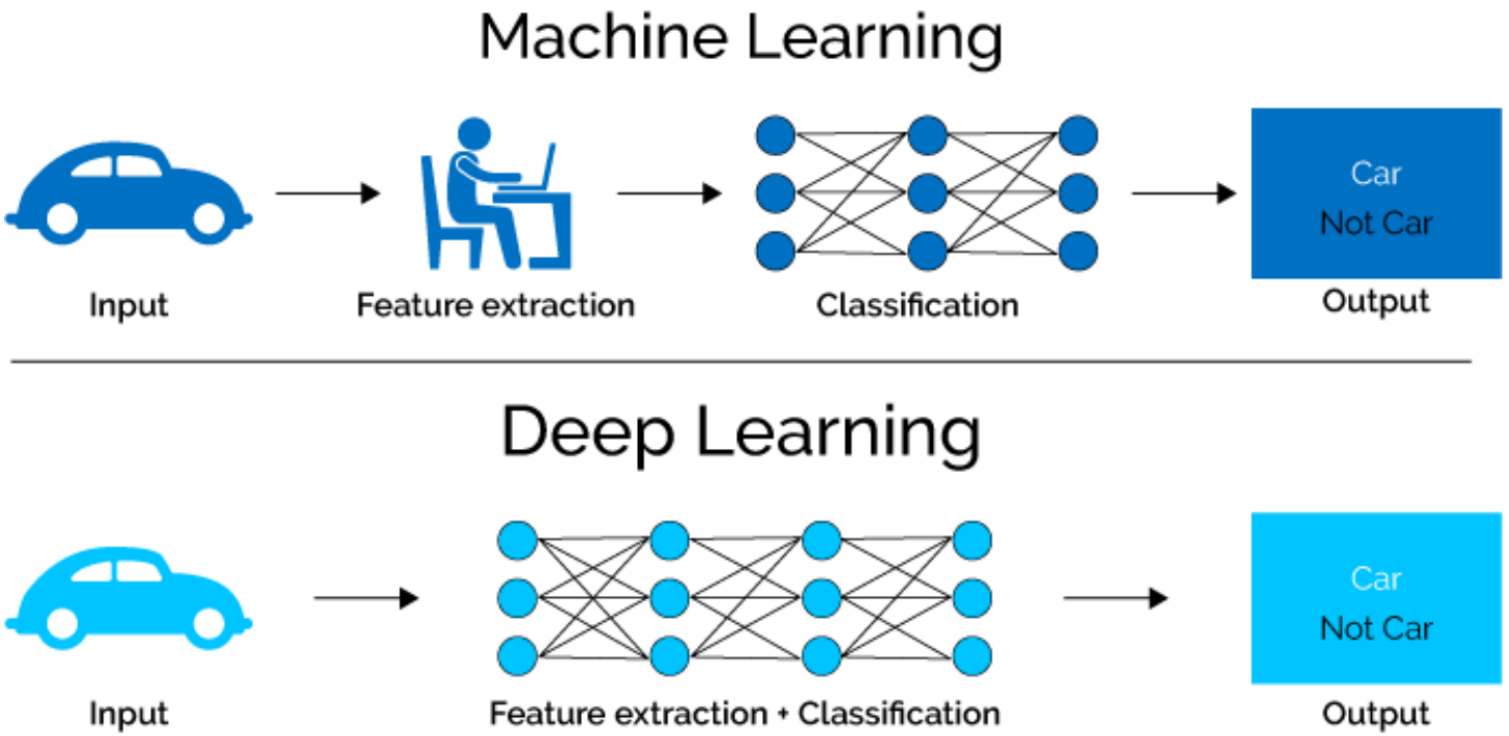
Exemplo de Função custo:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}^{(i)} - y^{(i)}\right)^2$$

$\hat{y}^{(i)}$ saída i estimada pelo modelo
$y^{(i)}$ saída i verdadeira ou esperada

Vantagens das Redes Neurais

Feature Engineering



[1] Towards Data Science

Vantagens das Redes Neurais

- Redes neurais também conseguem “aprender mais” quando recebem mais dados;