Projeto ABCIA

Módulo 04: Deep Learning

Aula 01

Prof. Dr. Luciano Ferreira Silva

UFRR | DCC DEPTO DE CIÊNCIA DA COMPUTAÇÃO | Softex | ABCIA ABC DA INTELIGÊNCIA ARTIFICIAL

Apresentação do Professor

- Graduado em Matemática (UFU) - 2003
- Mestre em Computação Gráfica (UFU) – 2006
- Doutor em Computação Gráfica (UFU) - 2009
- Certificação em Inteligência Artificial (Huawei – HCIA) - 2021

Professor do curso de Ciência da Computação/UFRR desde 2008, atuando nas disciplinas e pesquisa de **Computação Gráfica, RV e RA, Compiladores, IHM, Desenvolvimento de jogos e Visão Computacional.**

Prof. Dr. Luciano Ferreira Silva
E-mail: luciano.silva@ufrr.br

Objetivos e sua relevância

Objetivos de Aprendizagem

1. Definir o conceito de rede neural.

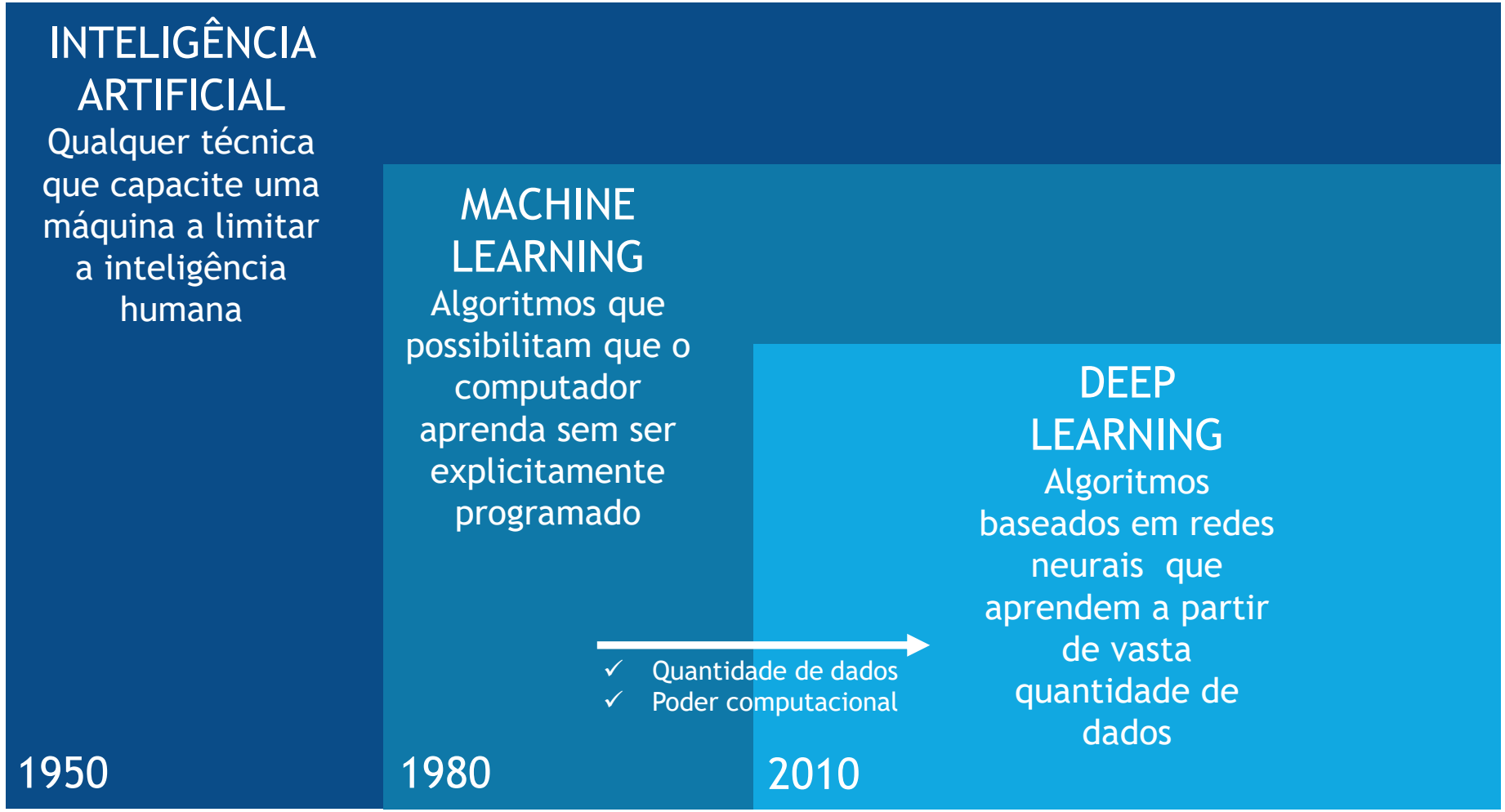
1.1. Apresentar os conceitos de: (1) “*node*”; (2) “*node conection*”; (3) “*weights*”; e (4) “*bias*” no contexto dos parâmetros das redes neurais.

2. Identificar finalidade e limitações das redes neurais no contexto da criação de modelos não-lineares.

2.1. Apresentar a finalidade: (1) das funções de ativação; (2) do algoritmo de “backpropagation”; e (3) do algoritmo de “gradient descent”.

Iniciando nossos estudos em Deep Learning

Deep Learning



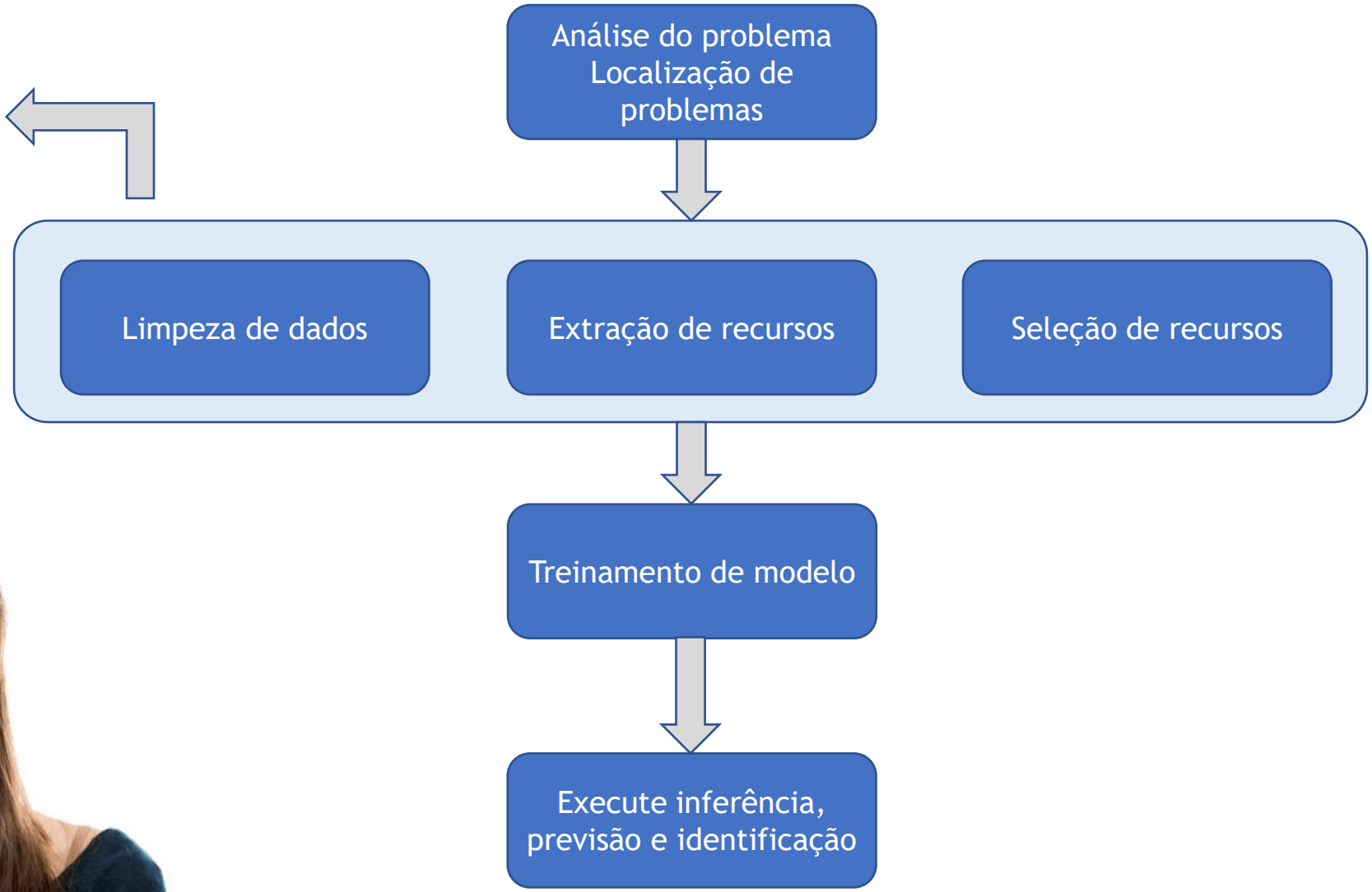
Deep Learning

Machine Learning tradicional	Deep Learning
Baixos requisitos de hardware no computador: dada a quantidade limitada de computação, o computador não precisa de uma GPU para computação paralela em geral.	Requisitos mais elevados de hardware no computador: para executar operações de matriz em dados massivos, o computador precisa de uma GPU para executar computação paralela.
Aplicável ao treinamento com uma pequena quantidade de dados e cujo desempenho não pode ser melhorado continuamente à medida que a quantidade de dados aumenta.	O desempenho pode ser alto quando parâmetros de peso de alta dimensão e dados de treinamento massivos são fornecidos.
Discriminação de problemas nível a nível	Aprendizagem Fim a Fim
Seleção manual de recursos	Extração automática de recursos baseada em algoritmo
Recursos fáceis de explicar	Recursos difíceis de explicar

✓A Deep Learning (Aprendizagem Profunda) tem grandes vantagens em campos como visão computacional, reconhecimento de fala e processamento de linguagem natural.

Machine Learning tradicional

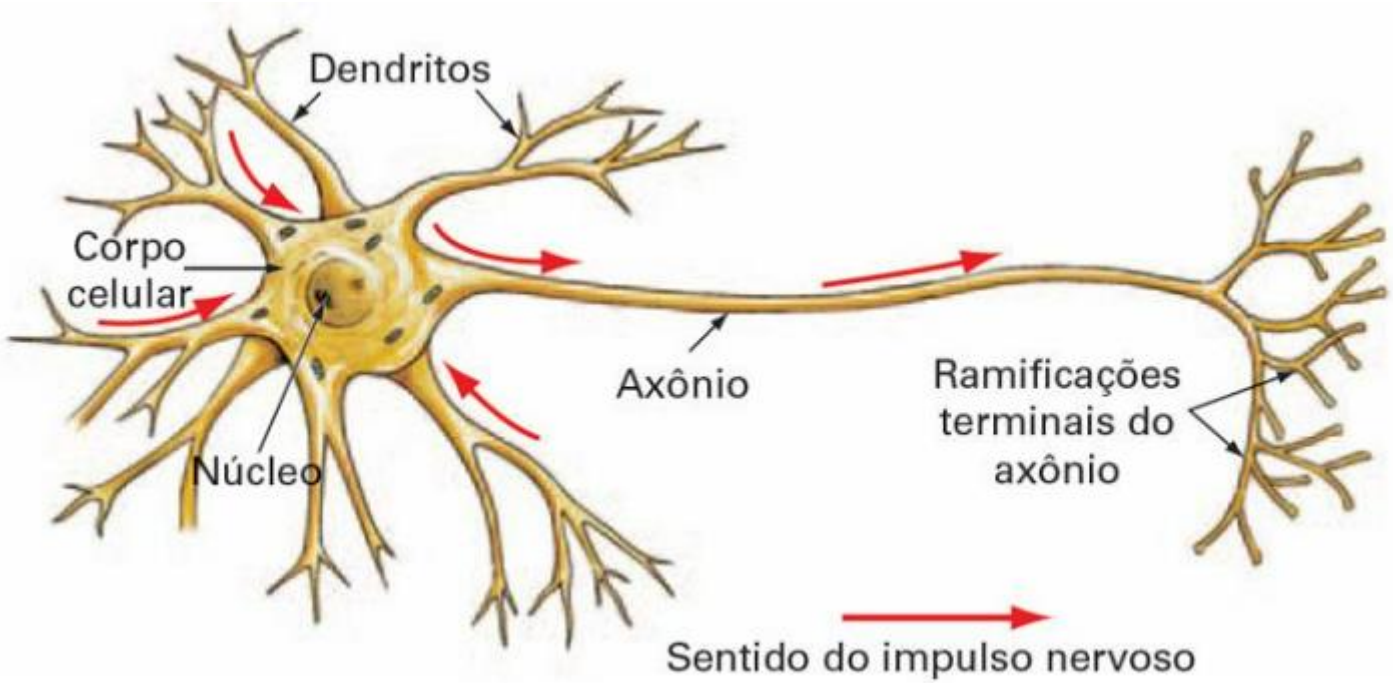
Pergunta: Podemos usar um algoritmo para executar o procedimento automaticamente?



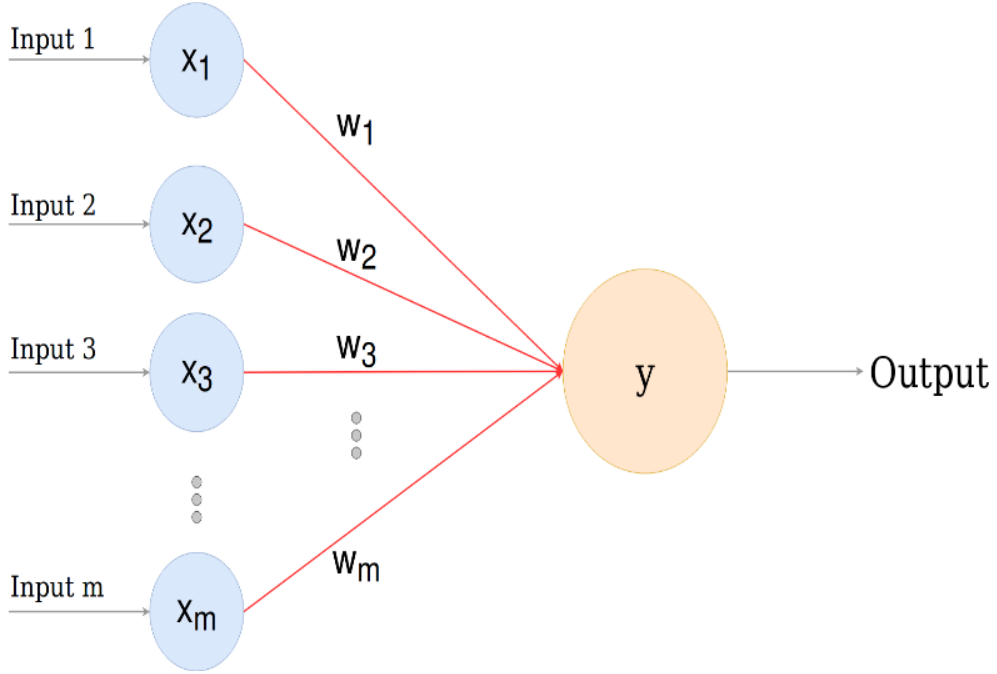
Deep Learning

✓A arquitetura em Deep Learning geralmente é uma rede neural **profunda.**

✓O termo "Profunda" em “Deep Learning" refere-se ao número de camadas da rede neural.



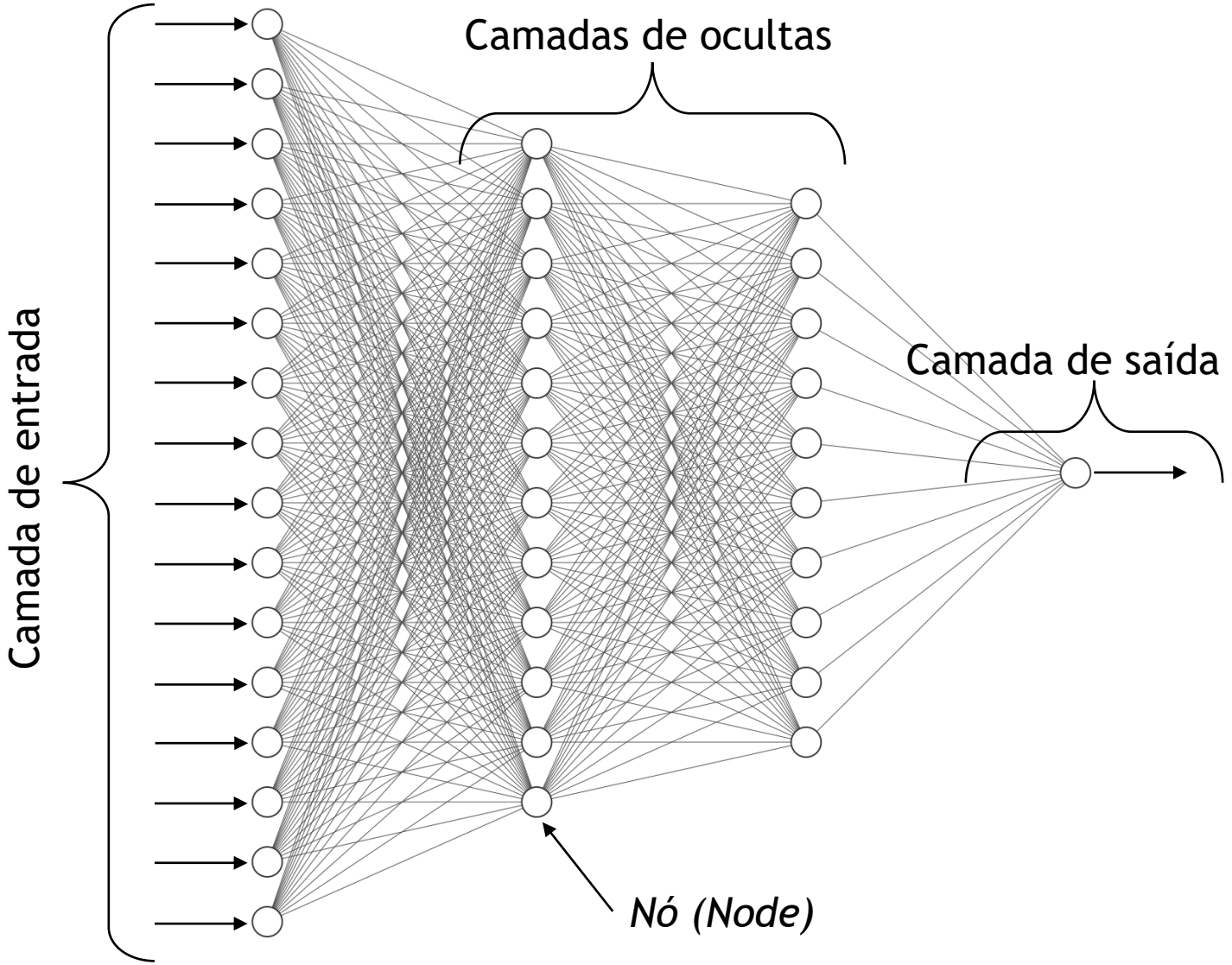
Neurônio Humano



Perceptron (1958)

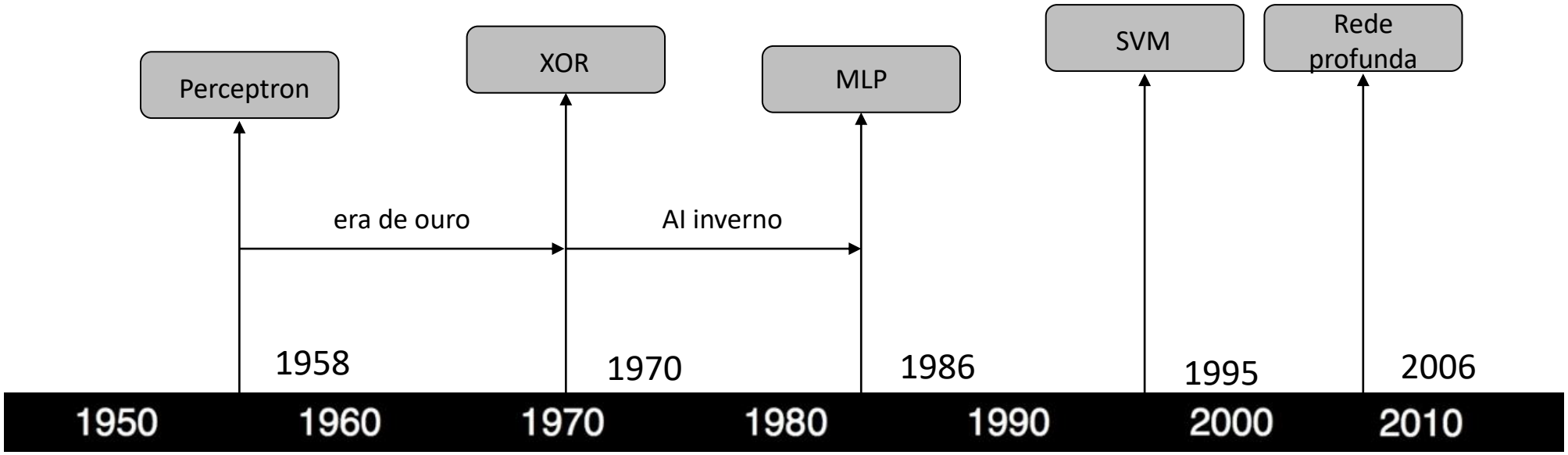
Redes profundas

- ✓**Definição de rede neural:** sistema computacional de processamento de informações, composta por elementos altamente interconectados, projetado para imitar a estrutura e funções do cérebro humano.



Redes profundas

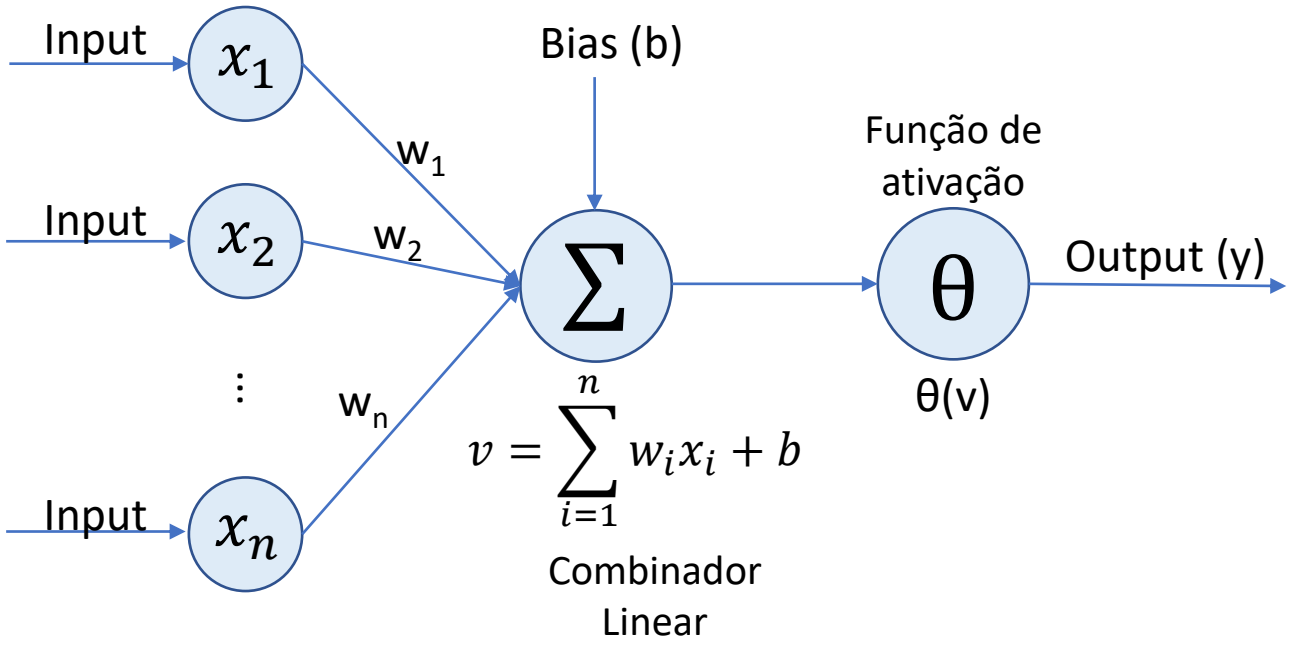
✓Histórico das redes neurais:



Perceptron de camada única

- ✓ Vetor de entrada: X = [x_1, x_2, …, x_n];
- ✓ Pesos (*weights*): W = [w_1, w_2, …, w_n];
- ✓ Bias: b;
- ✓ Função de ativação:

$$y = \theta(v) = sign(\mathrm{v}) = \begin{cases} 1, \mathrm{v} > 0, \\ -1, \mathrm{v} \leq 0. \end{cases}$$



- ✓ O Perceptron de uma camada é um classificador binário, por exemplo:
  - Se y = 1, resultado Classe A
  - Se y = -1, resultado Classe B

Perceptron de camada única

✓Interpretação geométrica do Perceptron de camada única

$$y = \theta(v) = sign(\mathrm{v}) = \begin{cases} 1, \mathrm{v} > 0, & \text{(Classe A)} \\ -1, \mathrm{v} \leq 0 & \text{(Classe B)} \end{cases}$$

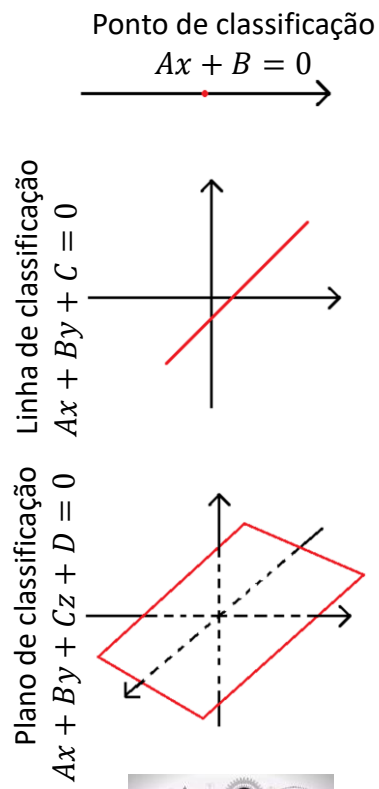
Perceba que toda a classificação ocorre em relação ao Valor 0 (zero)

✓Analisando então v = 0:

$$v = \sum_{i=1}^{n} w_i x_i + b = 0$$

✓Vamos analisar alguns casos para

X = [x_1, x_2, …, x_n];



Caso 1:

X = [x_1] → W = [w_1]

$w_1 x_1 + b = 0$ → Equação de um **Ponto**

Caso 2:

X = [x_1, x_2] → W = [w_1, w_2]

$w_1 x_1 + w_2 x_2 + b = 0$ → Equação de uma **Reta**

Caso 3:

X = [x_1, x_2, x_3] → W = [w_1, w_2 , w_3]

$w_1 x_1 + w_2 x_2 + w_3 x_3 + b = 0$ → Equação de um **Plano**

Caso 4:

X = [x_1, x_2,…, x_n] → W = [w_1, w_2 ,…, w_3]

$w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b = 0$ → Equação de um **Hiperplano**

Perceptron de camada única

✓Observando um pouco mais de perto:

Caso 2:

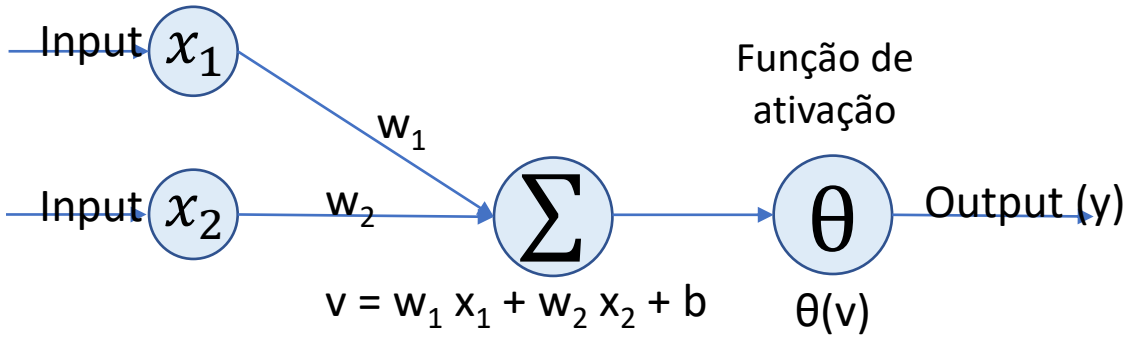
$X = [x_1, x_2] \rightarrow W = [w_1, w_2]$

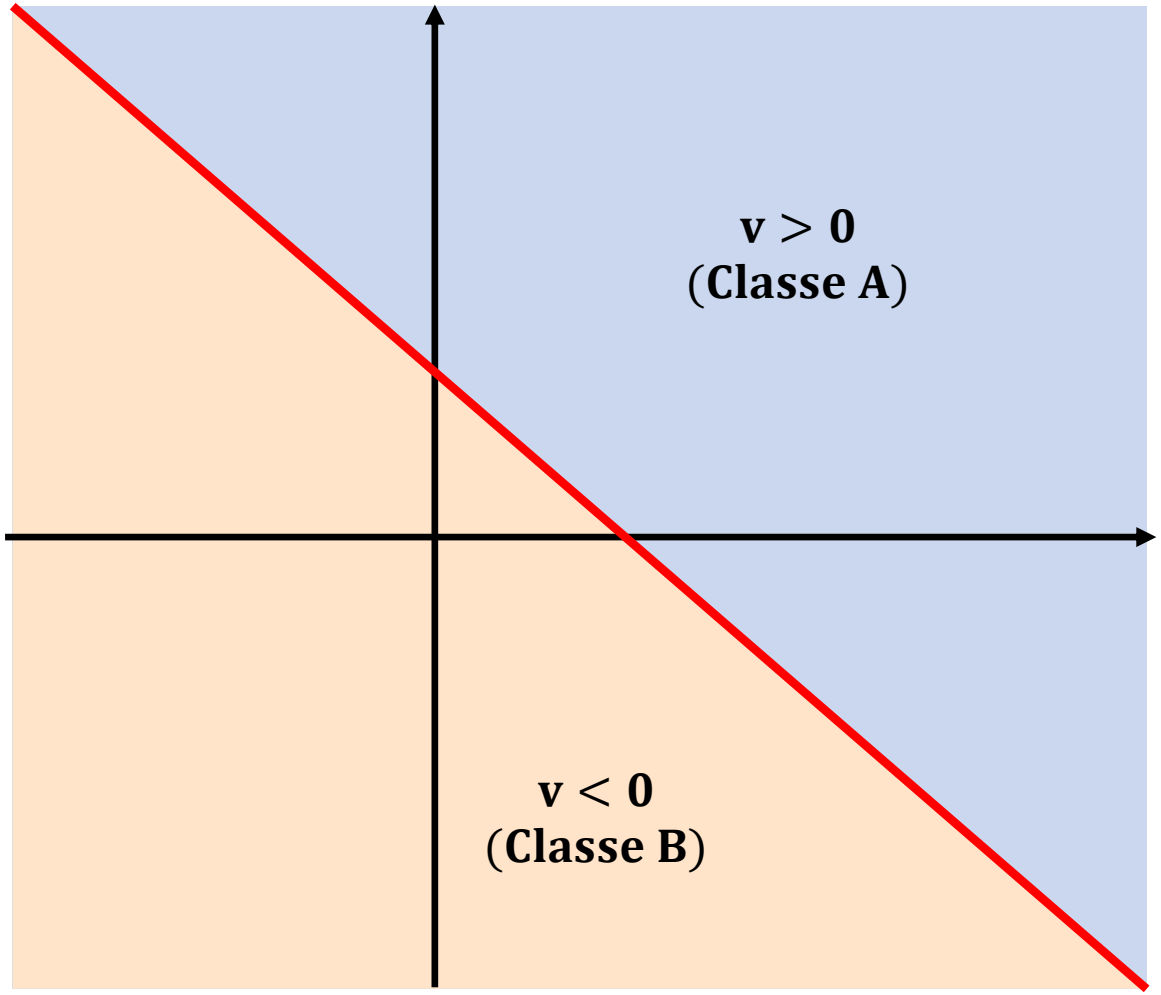
$v = w_1 x_1 + w_2 x_2 + b = 0$

Equação de uma **Reta**

Linha de classificação

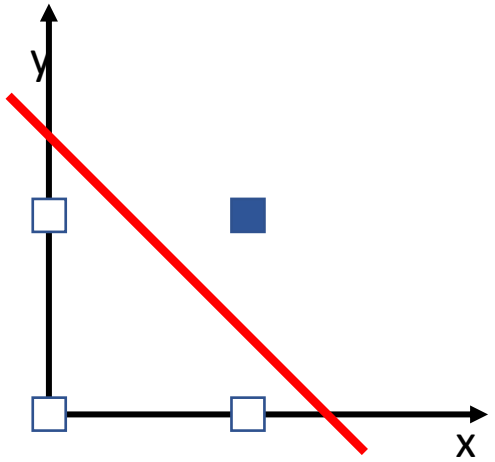
$Ax + By + C = 0$

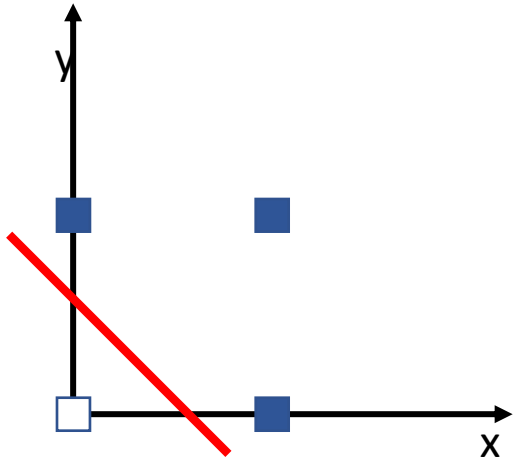


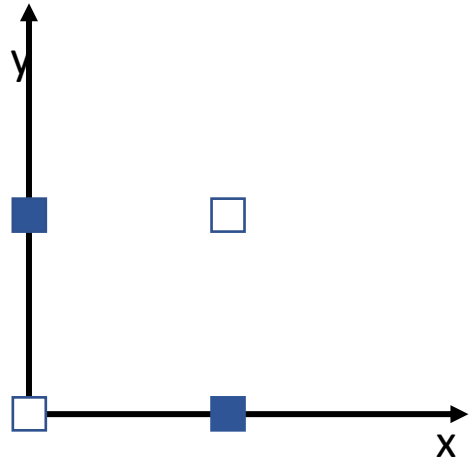


Perceptron de camada única

✓Problema XOR: em 1969, Minsky provou que um perceptron não pode processar dados não lineares.







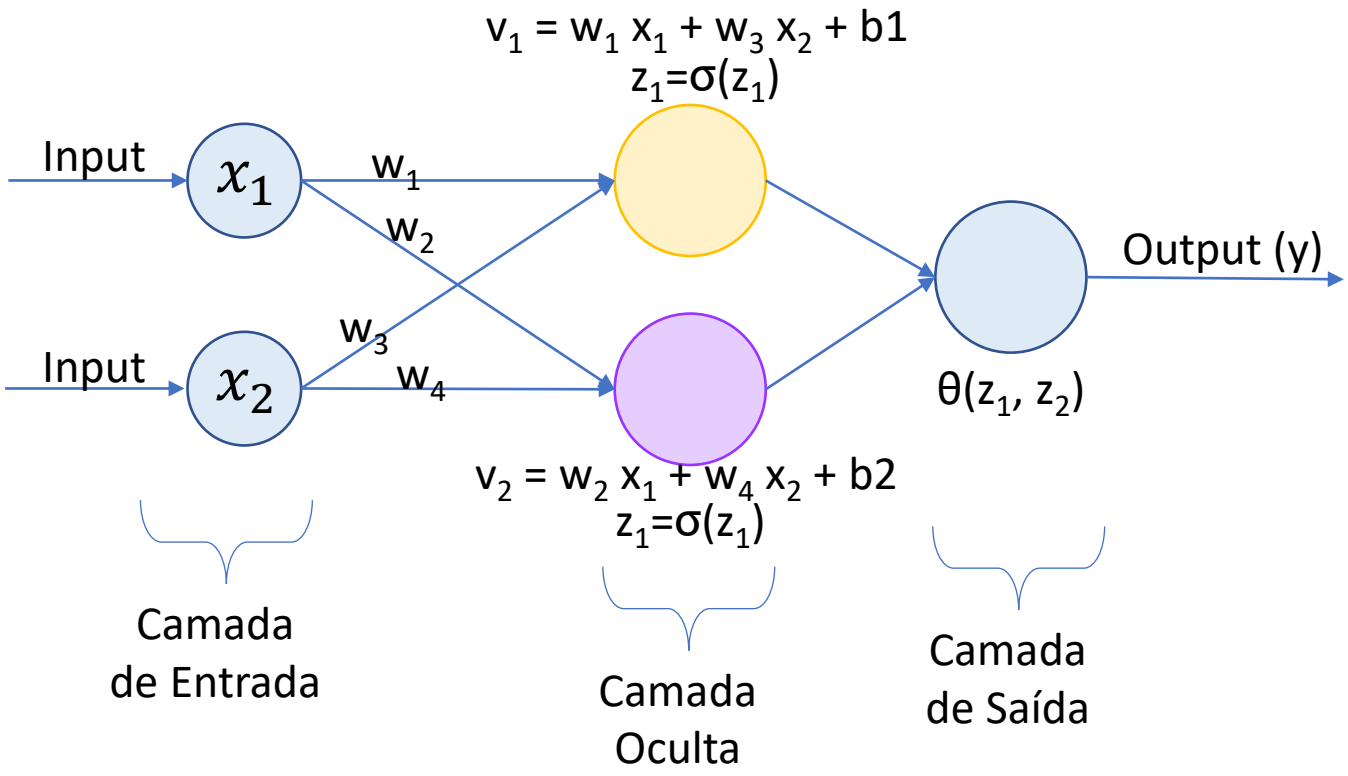
X		Y		X E Y	
V	1	V	1	V	1
V	1	F	0	F	0
F	0	V	1	F	0
F	0	F	0	F	0

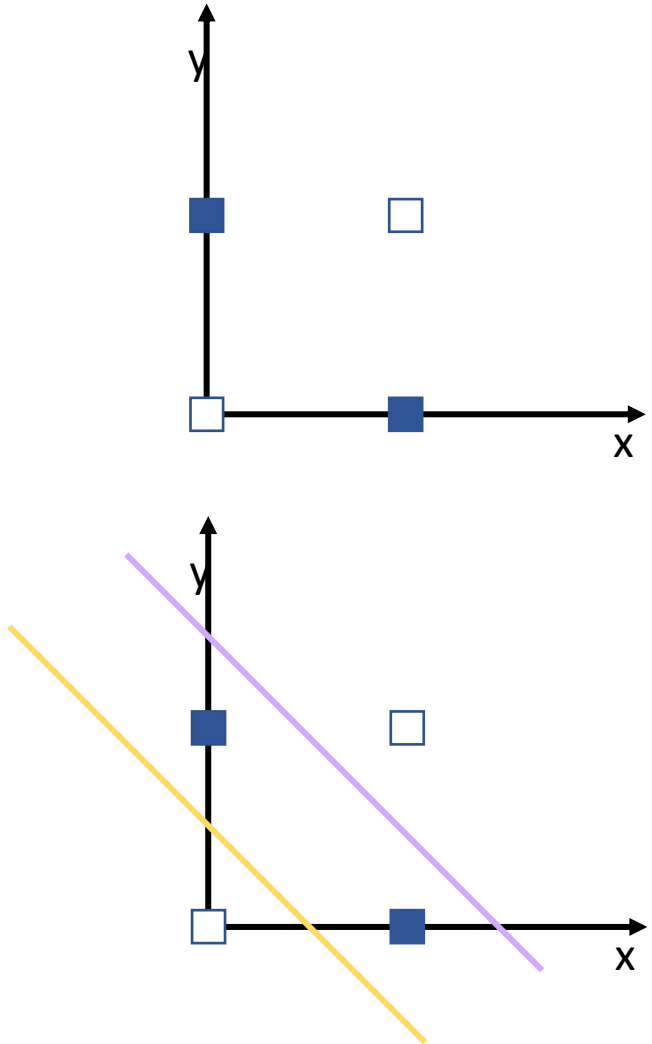
X		Y		X OU Y	
V	1	V	1	V	1
V	1	F	0	F	1
F	0	V	1	F	1
F	0	F	0	F	0

X		Y		X XOR Y	
V	1	V	1	V	0
V	1	F	0	F	1
F	0	V	1	F	1
F	0	F	0	F	0

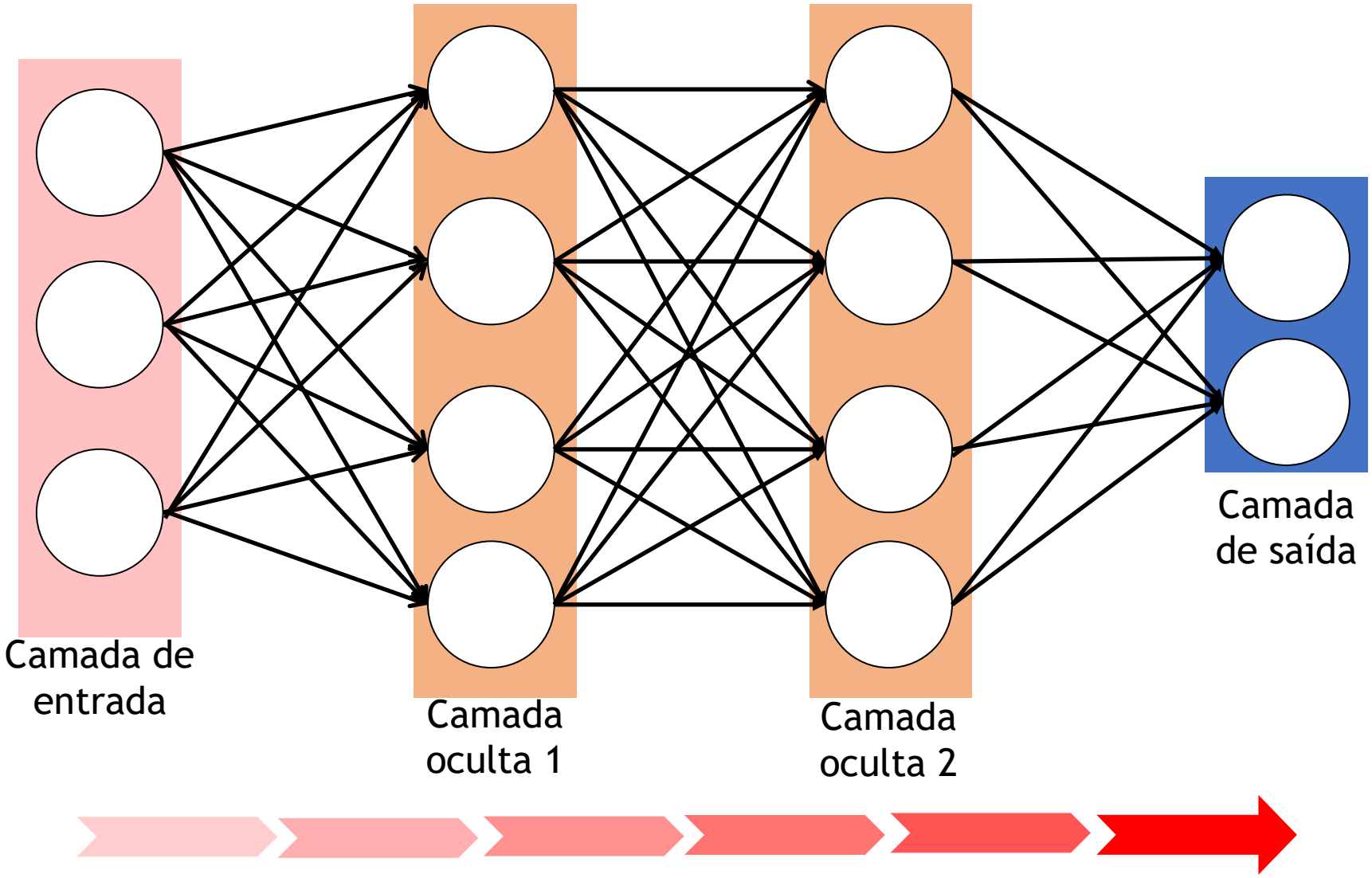
Perceptron multicamadas

✓Solução do problema XOR:





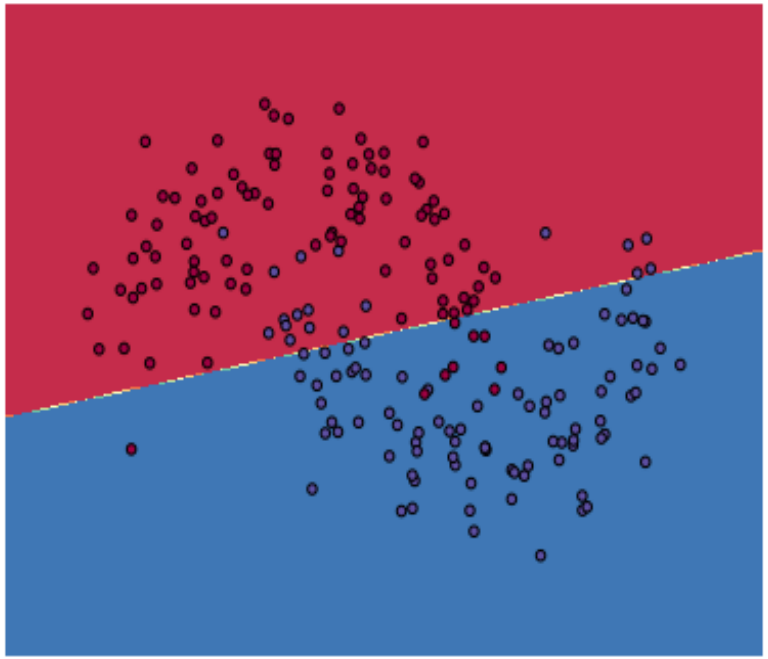
Rede Neural Feedforward



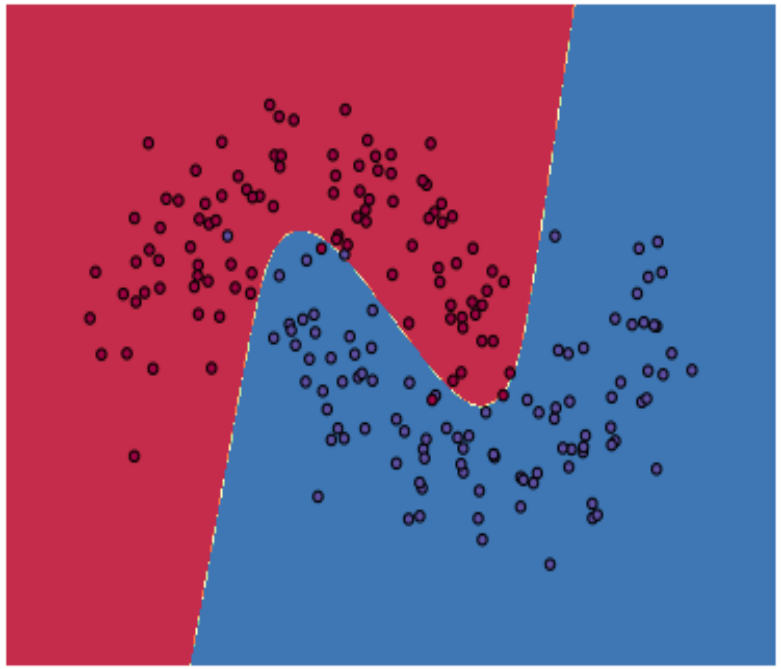
- ✓Cada camada se conecta à próxima camada, porém não há caminho de volta.
- ✓Todas as conexões portanto, têm a mesma direção, partindo da camada de entrada rumo a camada de saída.

Rede Neural Feedforward

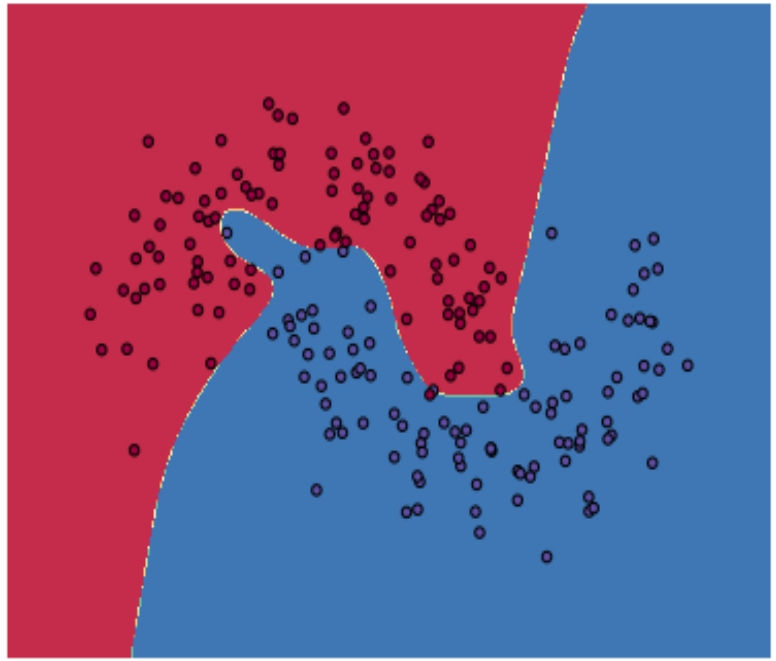
✓Impactos das camadas ocultas em uma rede neural



0 camadas ocultas



3 camadas ocultas



20 camadas ocultas

Função de Ativação

- ✓Possibilitam a rede neural aprender e compreender modelos não-lineares complexos.
- ✓Os neurônios realizam a seguinte **combinação linear**:

$$\Sigma(\text{pesos} * \text{inputs}) + \text{bias}$$

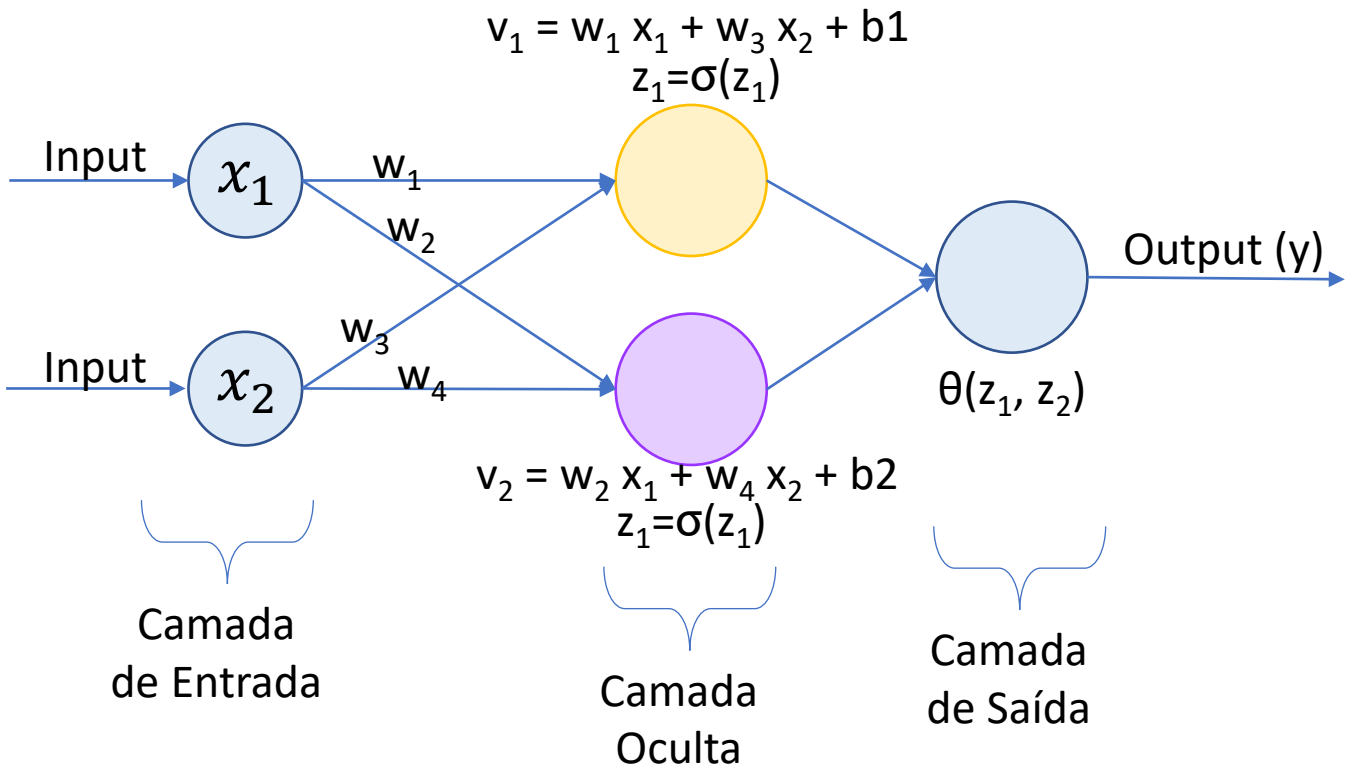
- ✓A função de ativação é a transformação **não-linear** desse valor.

$$Y = \text{Função\_Ativação}(\Sigma(\text{pesos} * \text{inputs}) + \text{bias})$$

- ✓Esta saída transformada é então enviada para a próxima camada de neurônios como entrada.

Função de Ativação

✓Relembrando a solução do problema XOR:



✓Mas qual o porquê do nome "função de ativação"?

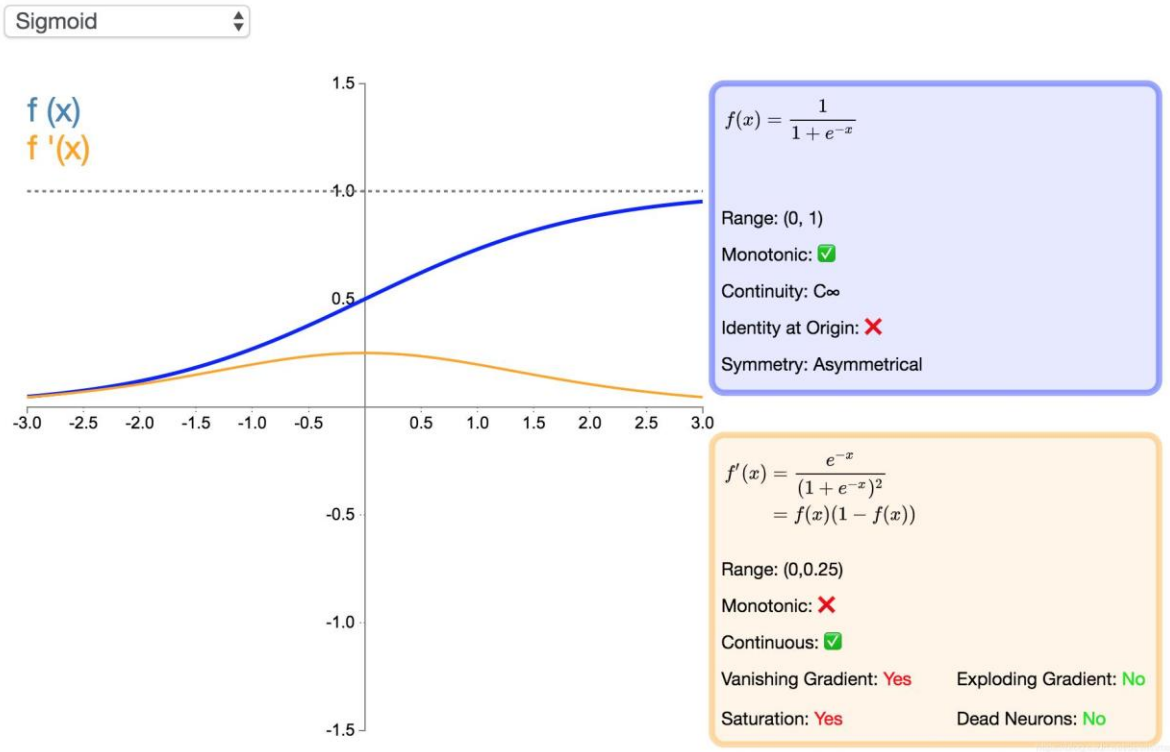
✓Ela possui a capacidade de "ativar" ou "desativar" um neurônio da próxima camada.

✓**Observação:** as camadas podem ter funções de ativação diferentes.

Função de Ativação

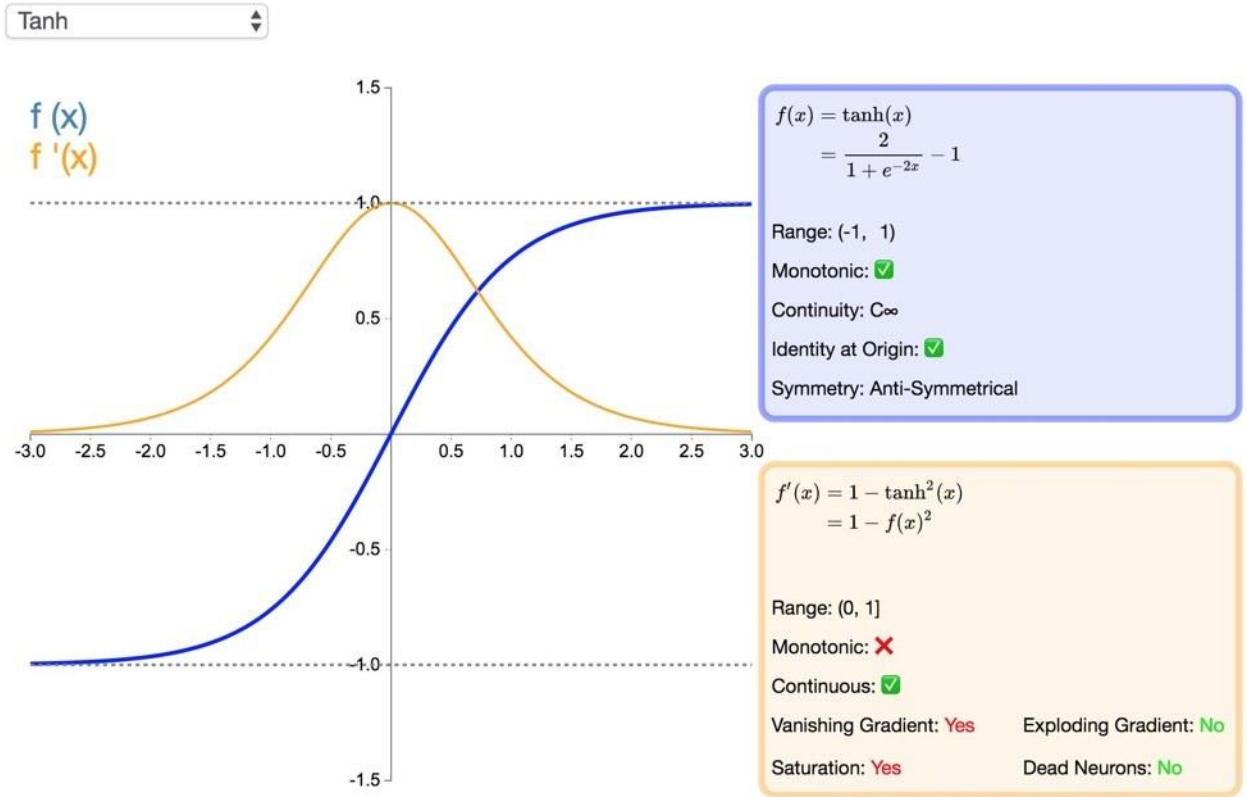
✓Sigmóide

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



✓Tanh

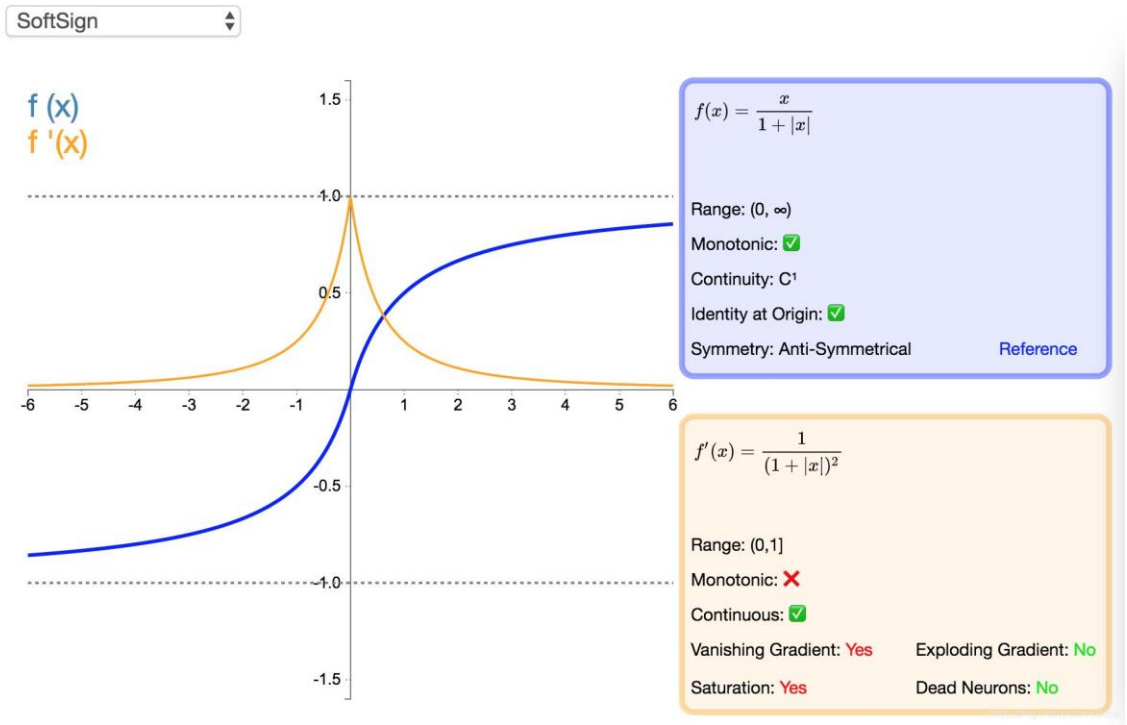
$$\tanh x = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



Função de Ativação

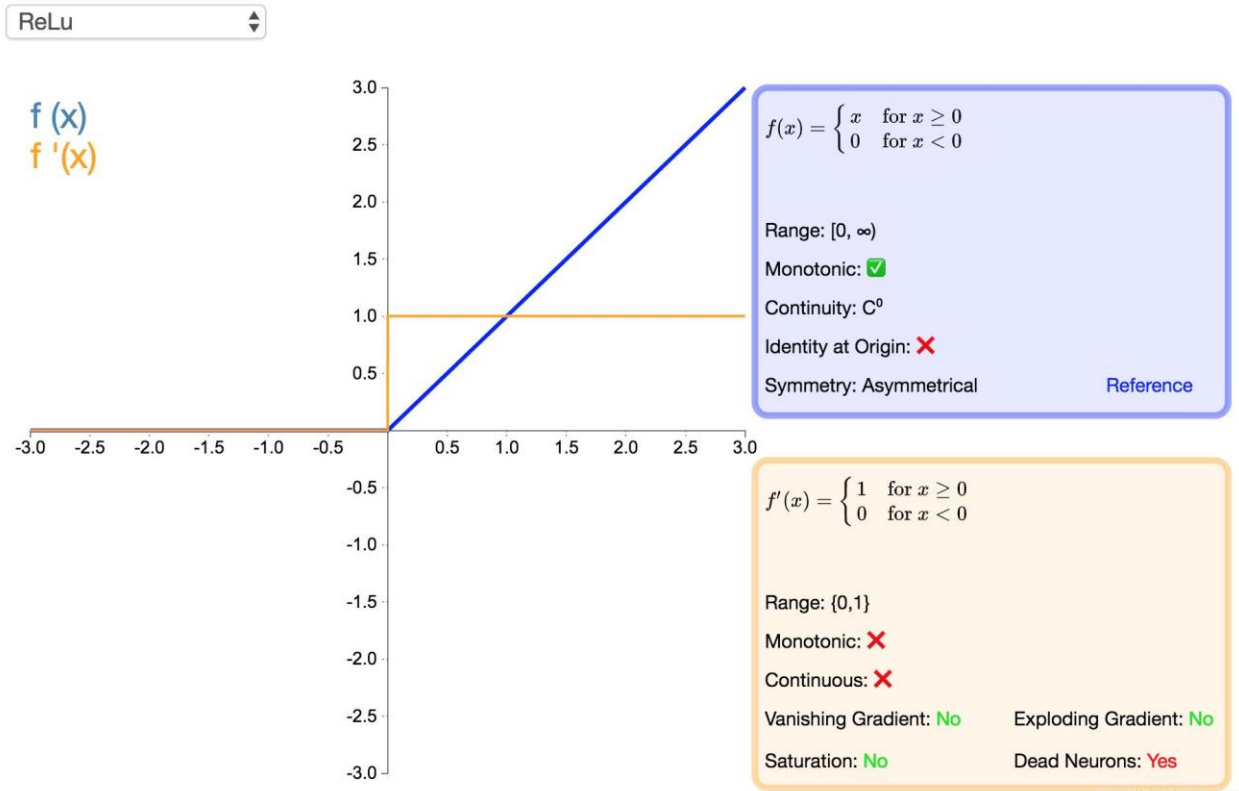
✓Softsign

$$f(x) = \frac{x}{|x| + 1}$$



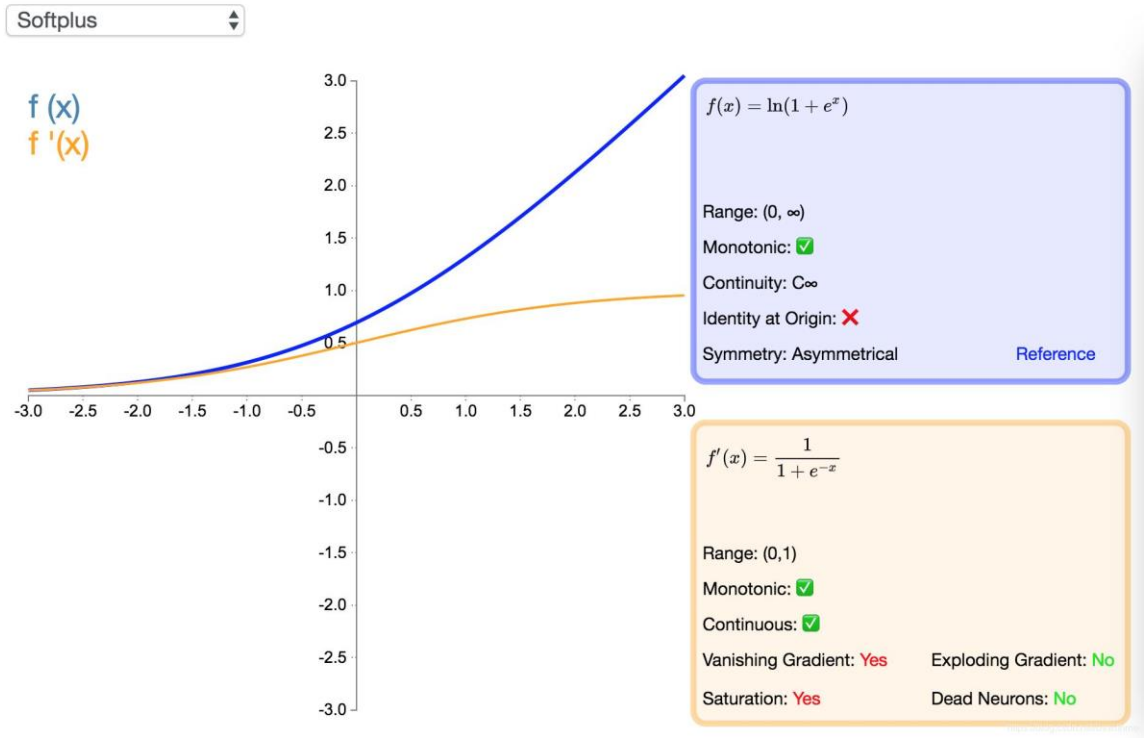
✓ReLU

$$y = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



Função de Ativação

✓**Softplus** $f(x) = \ln(e^x + 1)$



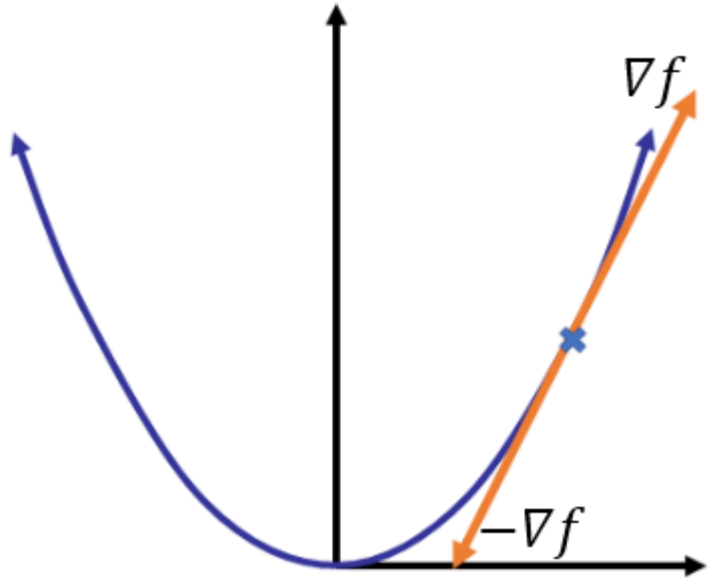
✓**Softmax** $\sigma(z_j) = \frac{e^{z_j}}{\sum_k e^{z_k}}$

- ✓É frequentemente usada como a camada de saída de uma tarefa de classificação multiclasse.
- ✓Mapeia um vetor real K-dimensional para outro vetor K-dimensional, onde cada elemento está no intervalo (0, 1). A soma de todos os elementos resultam 1.
- ✓O valor $\sigma(z_j)$ representa a probabilidade de z_j acontecer.

Gradiente Descendente

- ✓ O **Gradiente** (ou **vetor gradiente**) é um vetor que indica o sentido e a direção na qual, por deslocamento a partir do ponto especificado, obtém-se o maior incremento possível no valor de uma grandeza.
- ✓ O gradiente da função multivariada $f(x) = f(x_0, x_1, \dots, x_n)$ no $X' = [x_0', x_1', \dots, x_n']$ é mostrado da seguinte forma:

$$\nabla f(x_0', x_1', \dots, x_n') = [\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n}]$$



- ✓ Como a direção de ∇f é a de crescimento mais rápido da função, a direção do vetor gradiente negativo $-\nabla f$ é a direção de descida mais rápida da função.
- ✓ $-\nabla f$ recebe o nome de **Gradiente Descendente - GD** (*Gradient Descent*).

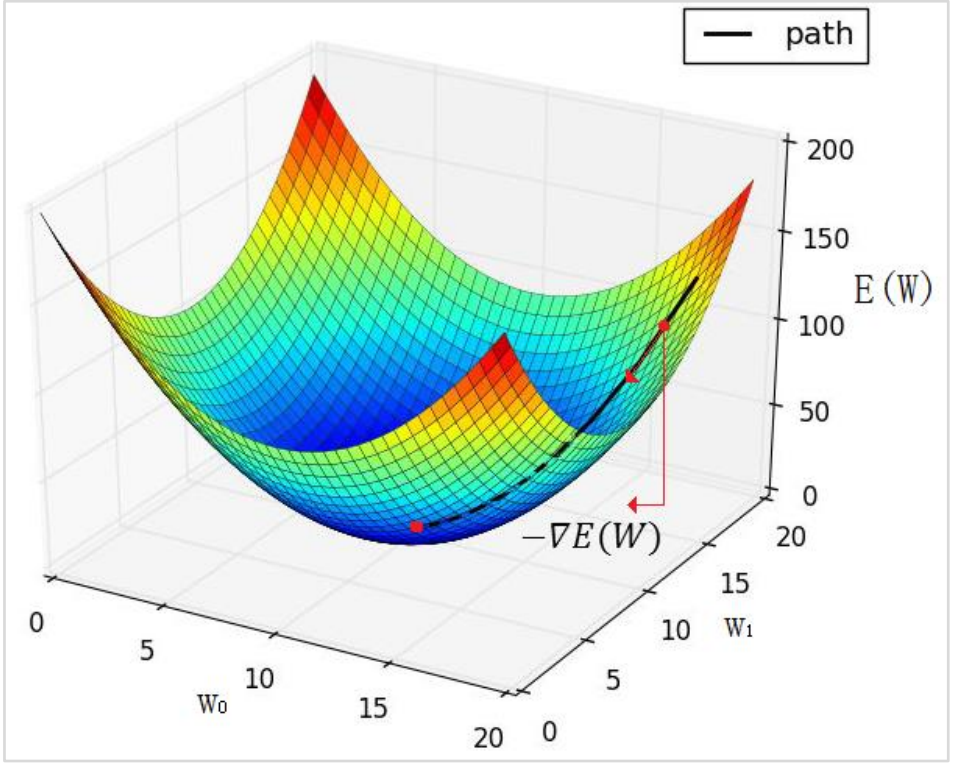
Gradiente descendente e Função de erro

- ✓ Durante o treinamento uma **função de erro (função de perda)** é usada para refletir o erro entre a saída alvo e a saída real do neurônio.
- ✓ A função de erro mais comum é a **Função de custo quadrático (*Quadratic cost function*)**.

$$E(w) = \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2,$$

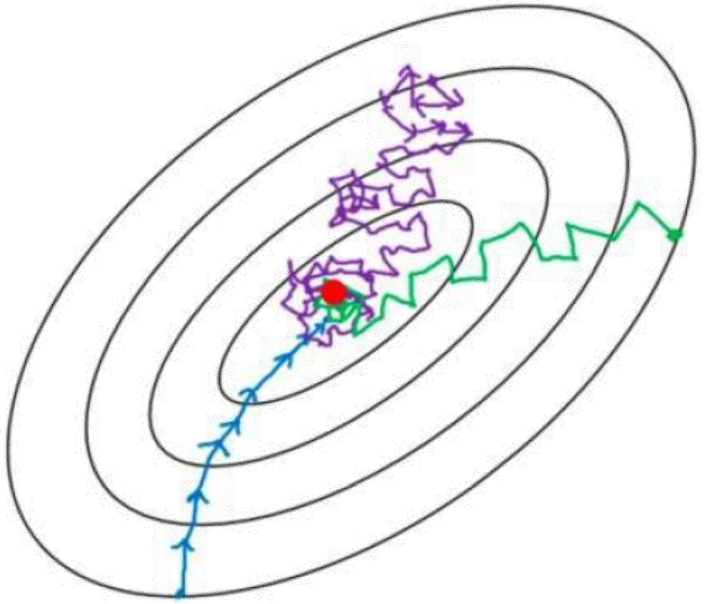
Sendo, d é um neurônio na camada de saída, *D* são todos os neurônios na camada de saída, t_d é a saída de destino, e o_d é a saída real.

- ✓ O GD de $E\,(-\nabla E(w))$ permite a atualização dos pesos de forma iterativa, minimizando a função de erro.



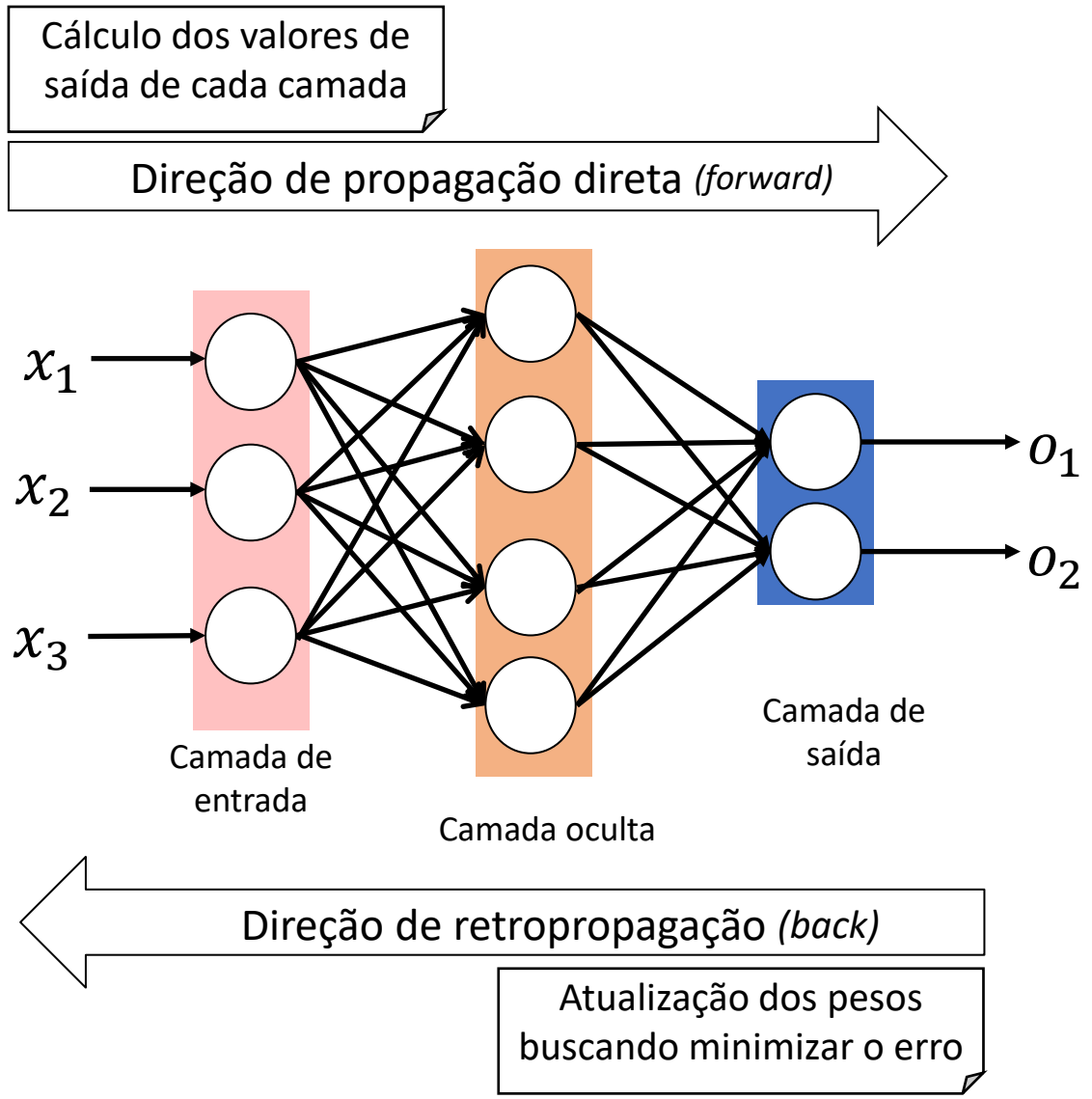
Tipos de algoritmos de Gradiente Descendente

Algoritmo	N° de amostras de treino usadas por interação	Velocidade de convergência	Difusão / Aplicação
*Batch Gradient Descent (**BGD**)*	Todas	Lento	Pouco usado
*Stochastic Gradient Descent (**SGD**)*	Uma amostra aleatória	Variável / Aleatório	Pouco usado
*Mini-Batch Gradient Descent (**MBGD**)*	Um lote aleatório de amostras	Intermediário	Muito usado



Algoritmo Backpropagation

- ✓ O objetivo do backpropagation é otimizar os pesos para que a rede neural possa aprender a mapear corretamente as entradas para as saídas.
- ✓ O algoritmo backpropagation emprega o Gradiente Descendente na Função de Erro para minimizar o erro entre a saída da rede e os valores alvos para estas saídas.
- ✓ O algoritmo realiza esse processo partindo da camada de saída até a camada de entrada.



Praticando exercícios

Exemplo de Questão

1) Qual dos seguintes problemas não pode ser tratado pelo Perceptron?

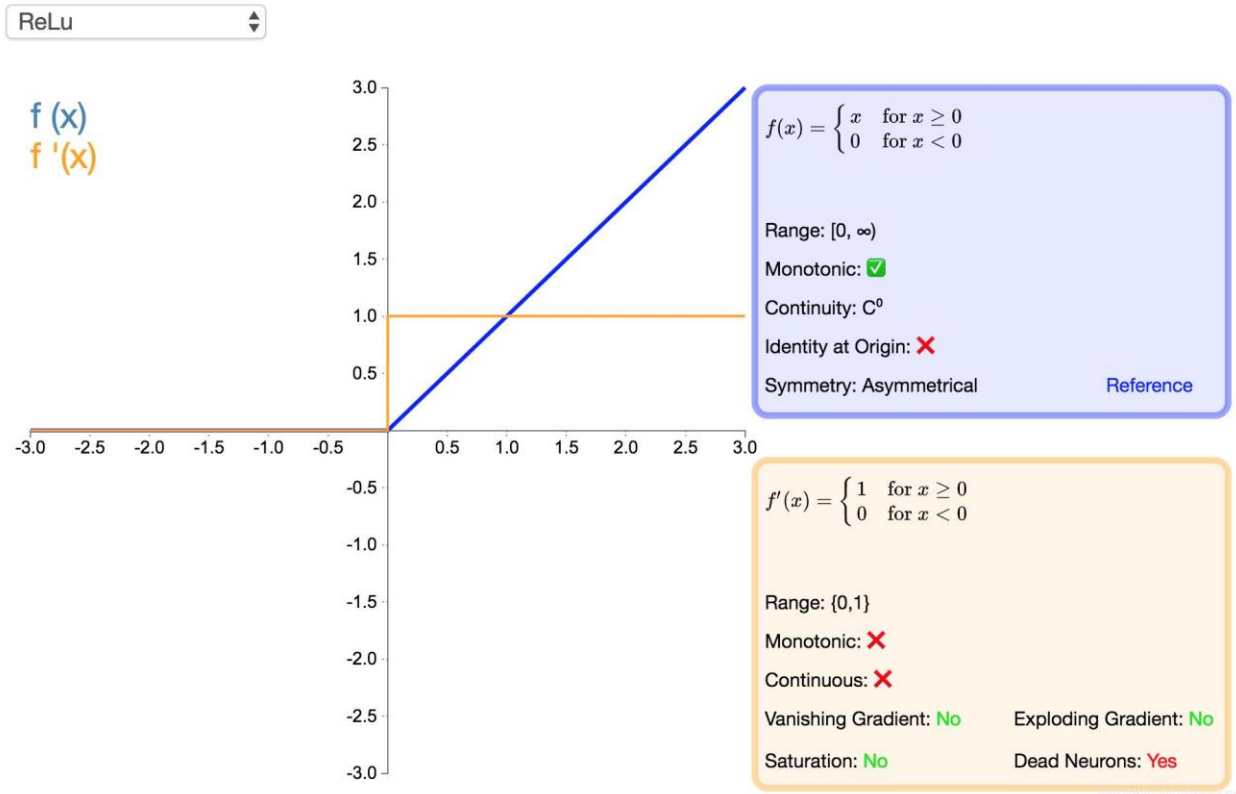
a) AND
b) OR
c) XOR
d) NOT

Exemplo de Questão

2) Qual das alternativas a seguir é o intervalo de saída da função ReLU?

a) [0, inf)
b) [0, 1]
c) [-1, 1]
d) [-1, 0]

$$y = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$



Exemplo de Questão

3) Na rede neural de aprendizado profundo, o perceptron é a rede neural mais simples. A declaração correta sobre sua estrutura é:

a) Existem apenas duas camadas ocultas
b) Possui apenas uma camada oculta
c) Sua rede usa a função de ativação sigmóide
d) Sua rede usa a função de ativação do RelU

Exemplo de Questão

4) Nas redes neurais, quais dos seguintes métodos são usados para atualizar os parâmetros ao treinar a rede para minimizar a função de perda?

a) Algoritmo de propagação Forward
b) Cálculo de agrupamento
c) Cálculo da convolução
d) Algoritmo Backpropagation

Exemplo de Questão

5) No algoritmo de descida de gradiente, qual dos seguintes algoritmos é o algoritmo mais confuso para a trajetória na superfície da função de perda?

a) SGD
b) BGD
c) MGD
d) MBGD

Exemplo de Questão

6) A Rede Neural Feedforward é uma rede neural simples, onde cada neurônio é organizado hierarquicamente, sendo, atualmente, uma das redes neurais artificiais mais utilizadas. Qual das seguintes declarações sobre as redes neurais Feedforward está correta:

a) Os neurônios com potência de computação estão conectados às camadas superior e inferior
b) Os nós de entrada tem poder de computação
c) Neurônios de uma mesma camada são conectados
d) As informações percorrem a rede da camada de entrada para a de saída

Exemplo de Questão

7) Quais são as funções de ativação comumente usadas em redes neurais?

a) Sigmoid
b) Tanh
c) ReLU
d) Danish

UFRR

DEPTO DE CIÊNCIA DA COMPUTAÇÃO

Softex

ABCIA
ABC DA INTELIGÊNCIA ARTIFICIAL

Até a próxima aula!

Continue praticando :)