

+

• • • • • .  
• • • • • .  
• + .  
+ •

|

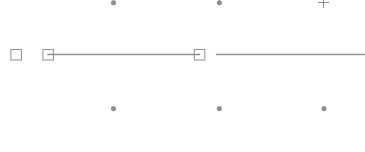
+

# FIAP

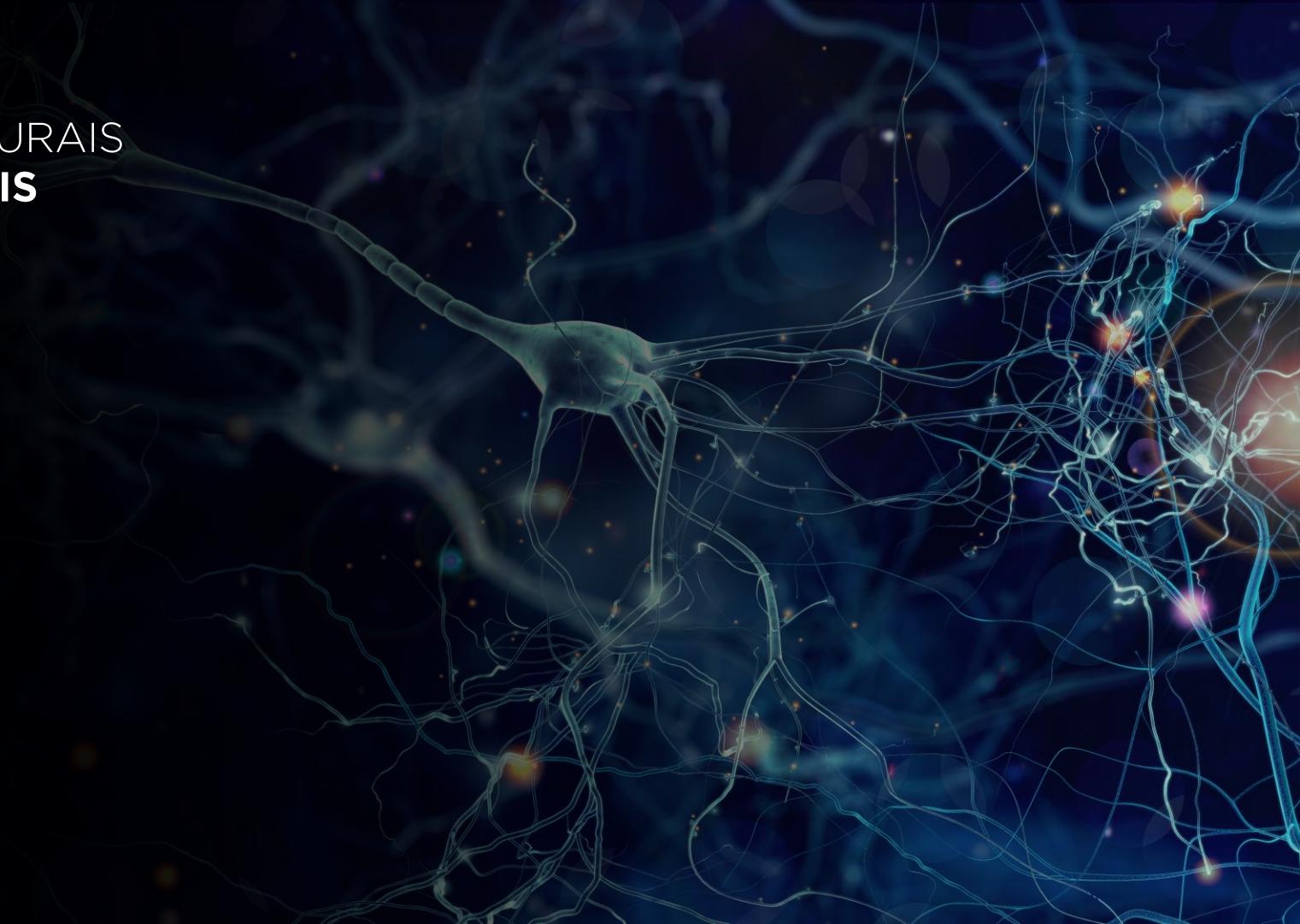
• • +  
■ □ \*  
• • ■  
• + • •



# REDES NEURAIS & **DEEP LEARNING**

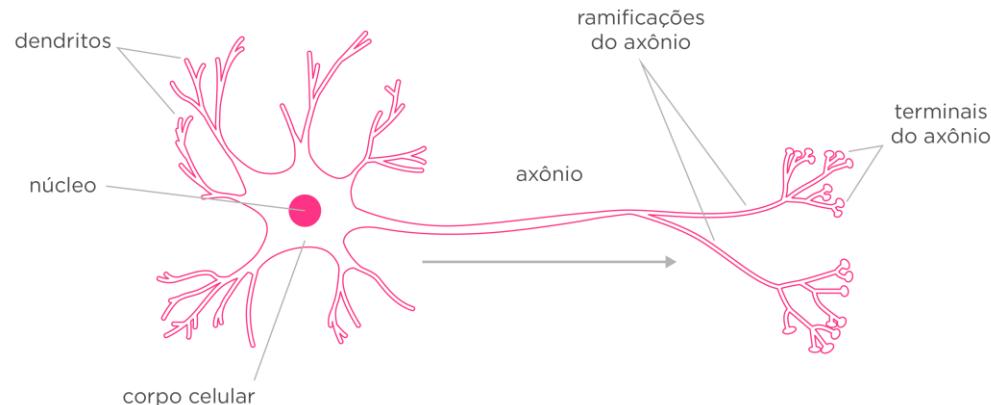


# REDES NEURAIS **ARTIFICIAIS**



## REDES NEURAIS **ARTIFICIAIS**

São sistemas inspirados nos neurônios biológicos e na estrutura maciçamente paralela do cérebro, com capacidade de adquirir, armazenar e utilizar conhecimento experimental.



# REDES NEURAIS **ARTIFICIAIS**

## Cérebro x Computador:

Item	Computador	Cérebro
Complexidade	<ul style="list-style-type: none"><li>• Estrutura ordenada.</li><li>• Processamento serializado.</li></ul>	<ul style="list-style-type: none"><li>• <math>10^{10}</math> Neurônios “processadores” com até <math>10^4</math> conexões.</li></ul>
Velocidade de processamento	<ul style="list-style-type: none"><li>• 10.000.000 de operações por segundo.</li></ul>	<ul style="list-style-type: none"><li>• 100 operações por segundo.</li></ul>
Poder computacional	<ul style="list-style-type: none"><li>• Uma operação por vez com 1 ou 2 entradas.</li></ul>	<ul style="list-style-type: none"><li>• Milhões de operações por vez, com milhares de entradas.</li></ul>

# REDES NEURAIS **ARTIFICIAIS**

## Cérebro x Computador:

- Aritmética: 1 cérebro = 1/10 calculadora de bolso.
- Visão: 1 cérebro = 1000 supercomputadores.
- Memória de detalhes arbitrários: computador ganha.
- Memória de fatos do mundo real: o cérebro vence.
- Um computador deve ser programado explicitamente.
- O cérebro pode aprender experimentando o mundo.

# REDES NEURAIS **ARTIFICIAIS**

## Algumas Curiosidades:

- Nascemos com cerca de 100 bilhões de neurônios.
- Um neurônio pode se conectar a até 100.000 outros neurônios.
- Os sinais “movem-se” via impulsos eletroquímicos.
- As sinapses liberam um transmissor químico - cuja soma faz com que um limiar seja alcançado -, fazendo com que o neurônio “dispare”.
- As sinapses podem ser inibitórias ou excitatórias.

# REDES NEURAIS **ARTIFICIAIS**

## Aplicações:

### Classificação

- Reconhecimento de caracteres
- Reconhecimento de imagens
- Diagnóstico médico
- Análise de crédito
- Detecção de fraudes

### Clusterização

- Agrupamento de sequências de DNA
- Mineração de dados
- Agrupamento de clientes

### Previsão/Regressão

- Previsão do tempo (sistemas complexos)
- Previsão financeira (câmbio, bolsa...)
- Previsão de séries Temporais

# REDES NEURAIS **ARTIFICIAIS**

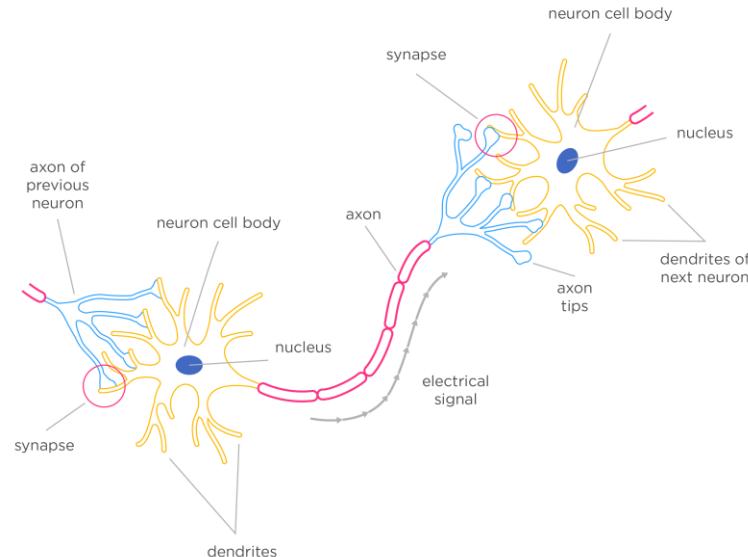
## Inspiração biológica:

- O cérebro faz com que tarefas de classificação pareçam fáceis.
- O processo cerebral é realizado por redes de neurônios.
- Cada neurônio é conectado a vários outros neurônios.

# REDES NEURAIS **ARTIFICIAIS**

## Inspiração biológica:

- O neurônio recebe impulsos (sinais) de outros neurônios por meio dos seus dendritos.
- O neurônio envia impulsos para outros neurônios por meio do seu axônio.
- O axônio termina num tipo de contato chamado sinapse, que conecta-o com o dendrito de outro neurônio.



# REDES NEURAIS **ARTIFICIAIS**

Inspiração biológica:

- vídeo

# REDES NEURAIS **ARTIFICIAIS**

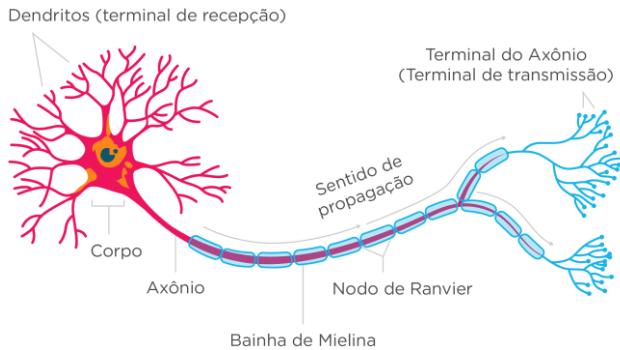
## Inspiração biológica - Aprendizado:

- O aprendizado ocorre por sucessivas modificações nas sinapses que interconectam os neurônios em função da maior ou menor liberação de neurotransmissores.
- À medida em que novos eventos ocorrem, determinadas ligações entre neurônios são reforçadas, enquanto outras são enfraquecidas.
- Este ajuste nas ligações entre os neurônios é uma das características das redes neurais artificiais.

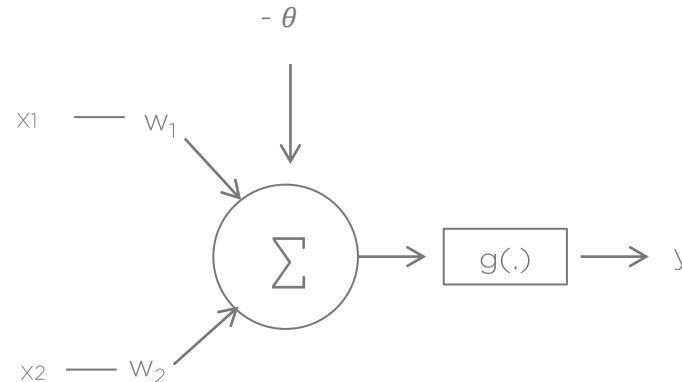
# REDES NEURAIS ARTIFICIAIS

## Inspiração:

- Neurônio biológico

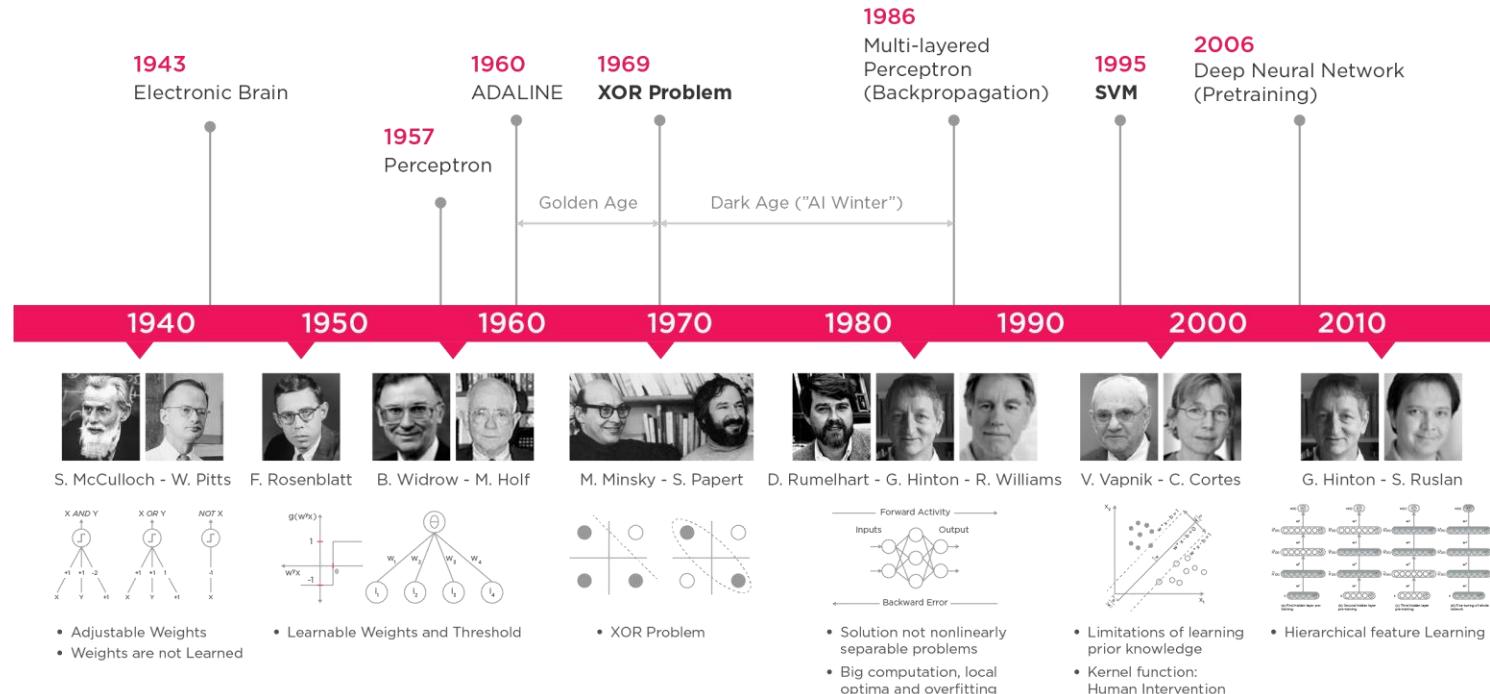


- Neurônio Artificial



# REDES NEURAIS ARTIFICIAIS

## História:



# REDES NEURAIS **ARTIFICIAIS**

## Neurônio de McCulloch e Pitts:

- McCulloch era um psiquiatra e um neuroanatomista. Pitts era um matemático que foi trabalhar com McCulloch na Universidade de Chicago; ambos faziam parte de um dos primeiros grupos do mundo dedicado ao estudo da Biofísica Teórica, criado por Nicolas Rashevsky.
- Em 1943, o conhecimento sobre os neurônios biológicos era muito limitado. As bases iônicas e elétricas da atividade neuronal eram ainda incertas, porém já se sabia da existência de potenciais de ação e da sua natureza “tudo ou nada”.
- McCulloch e Pitts propuseram um modelo de sistema neural em que as unidades básicas, os neurônios, são bastante simples no seu funcionamento.

# REDES NEURAIS **ARTIFICIAIS**

## Neurônio de McCulloch e Pitts:

1. A atividade de um neurônio é binária, ou seja, a cada instante o neurônio ou está disparando (atividade = 1) ou não está disparando (atividade = 0);
2. A rede neural é constituída por linhas direcionadas, com pesos ajustáveis, ligando os neurônios. Essas linhas (inspiradas nas sinapses) podem ser excitatórias ou inibitórias (positivas ou negativas);
3. Cada neurônio tem um limiar fixo  $\Theta$ , de maneira que ele só dispara se a entrada total chegando a ele, num dado instante, for maior ou igual a  $\Theta$ ;
4. A chegada de uma única sinapse inibitória num dado instante evita o disparo do neurônio, independentemente do número de sinapses excitatórias que estejam chegando conjuntamente com a sinapse inibitória.

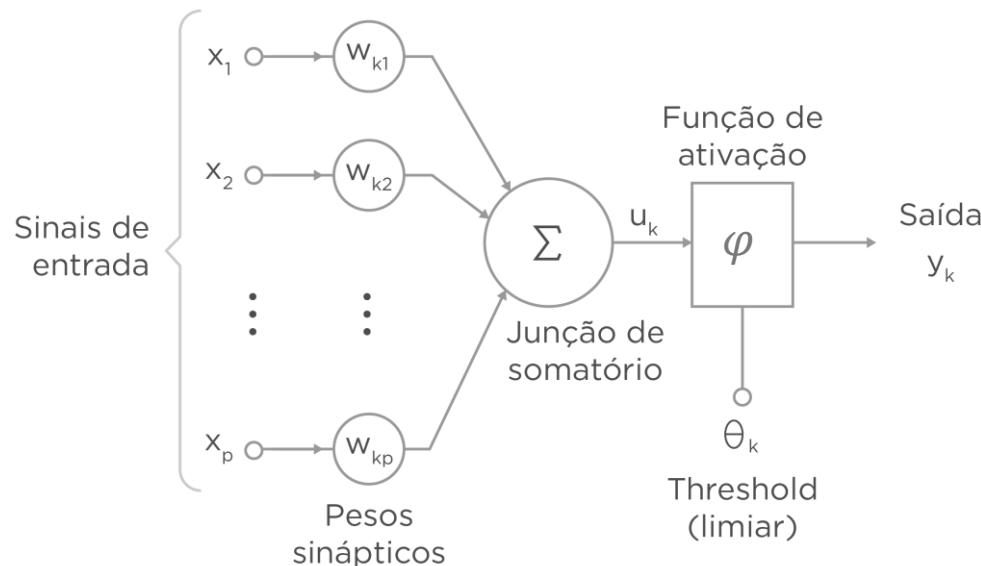
# REDES NEURAIS **ARTIFICIAIS**

## Neurônio de McCulloch e Pitts:

- Uma restrição existente no modelo criado é que as redes desenvolvidas só conseguem implementar funções linearmente separáveis, ou seja, aquelas que podem separar os padrões por meio de uma reta.

# REDES NEURAIS **ARTIFICIAIS**

Neurônio de McCulloch e Pitts:



# REDES NEURAIS **ARTIFICIAIS**

## Neurônio de McCulloch e Pitts:

A saída  $y$  do neurônio de McCulloch-Pitts pode ser equacionada por:

$$y = \varphi \left( \sum_{i=1}^n x_i w_i + \theta_k \right)$$

The diagram shows the mathematical equation for a McCulloch-Pitts neuron. Red arrows point from the text labels below to specific parts of the equation: 'Saída da rede' points to the output variable  $y$ ; 'Função de ativação' points to the activation function symbol  $\varphi$ ; 'Porta do limiar' points to the summation term; and 'Limiar (Threshold)' points to the constant term  $\theta_k$ .

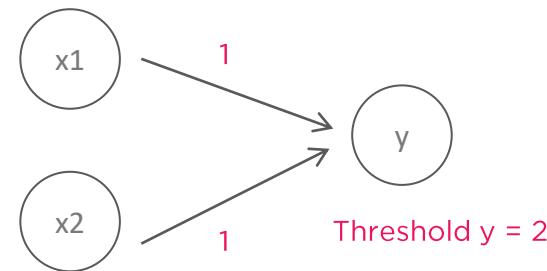
# REDES NEURAIS ARTIFICIAIS

## Neurônio de McCulloch e Pitts:

Exemplo: implementação de portas lógicas:

A função AND resulta na resposta “true” se ambas as entradas são valoradas com “true”; caso contrário a resposta é “false”. Se nós representamos “true” por ‘1’ e “false” por ‘0’, isso nos dá o seguinte conjunto de 4 pares (padrões) de treinamento entrada/saída (alvo):

X1	x2	Y
0	0	0
1	0	0
0	1	0
1	1	1



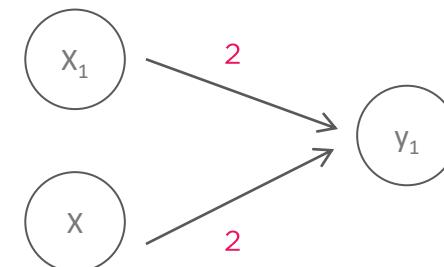
# REDES NEURAIS **ARTIFICIAIS**

Neurônio de McCulloch e Pitts:

Exemplo: Implementação de portas lógicas:

A função **OR** (threshold = 2) :

X1	x2	Y
0	0	0
1	0	1
0	1	1
1	1	1



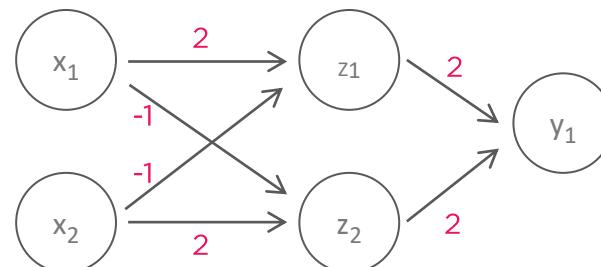
# REDES NEURAIS ARTIFICIAIS

Neurônio de McCulloch e Pitts:

Exemplo: implementação de portas lógicas:

Como implementar a função XOR ?:

X1	x2	Y
0	0	0
1	0	1
0	1	1
1	1	0



$$x_1 \text{ XOR } x_2 \leftrightarrow (x_1 \text{ AND NOT } x_2) \text{ OR } (x_2 \text{ AND NOT } x_1)$$

# REDES NEURAIS **ARTIFICIAIS**

## Aprendizado:

A principal propriedade de uma rede neural é sua capacidade de aprender a partir de estímulos do ambiente e de melhorar seu desempenho por meio do aprendizado.

O processo de aprendizado é iterativo e implica na alteração dos pesos (sinápticos) da rede neural.

# REDES NEURAIS **ARTIFICIAIS**

## Aprendizado:

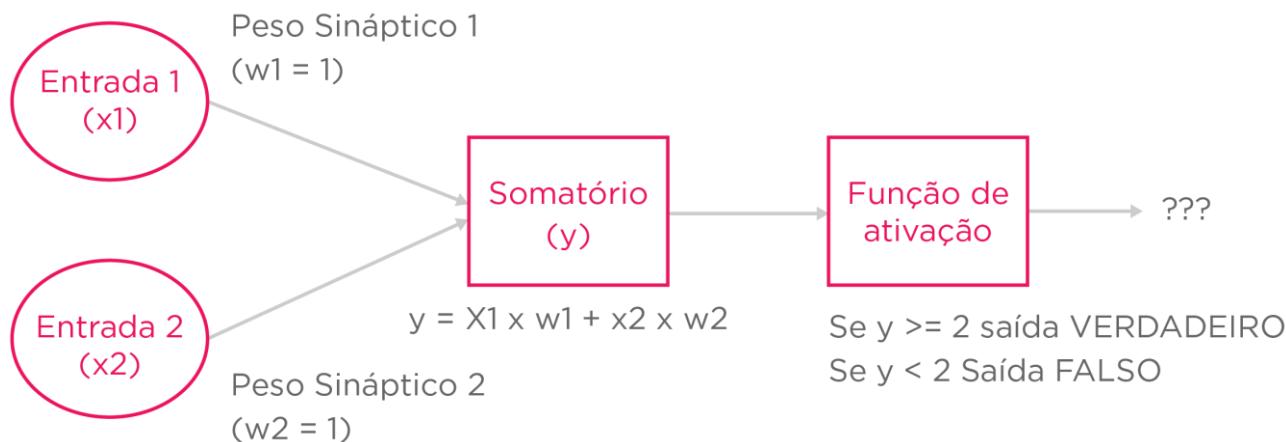
Definição de Mendel e McLaren (1970):

Aprendizado é um processo pelo qual parâmetros livres de uma rede neural são adaptados por meio de um processo de estimulação vindo do ambiente em que a rede está incorporada. O tipo de aprendizado é determinado pela maneira como os parâmetros são alterados.

# REDES NEURAIS **ARTIFICIAIS**

## Aprendizado:

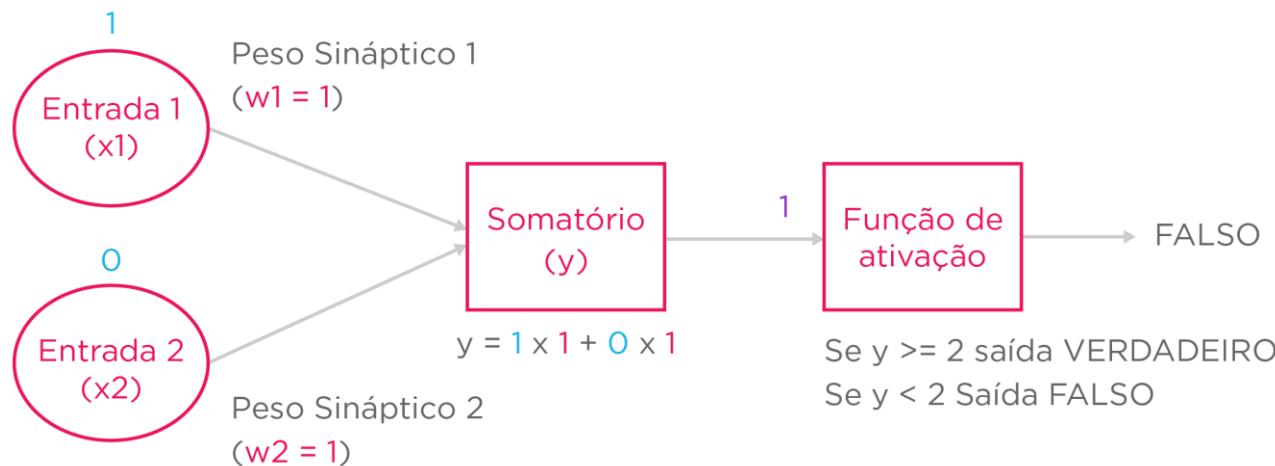
O que queremos:



# REDES NEURAIS ARTIFICIAIS

## Aprendizado:

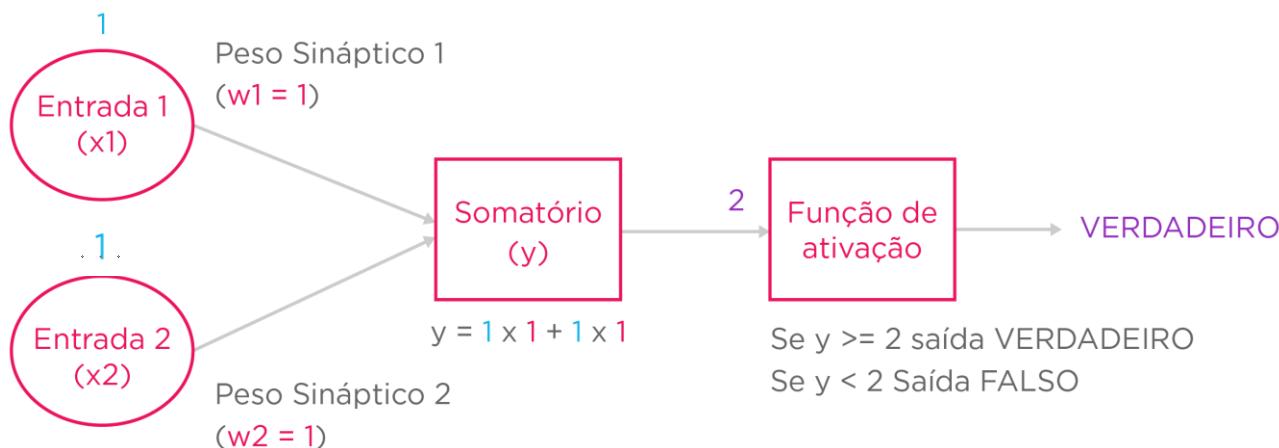
O que queremos:



# REDES NEURAIS ARTIFICIAIS

## Aprendizado:

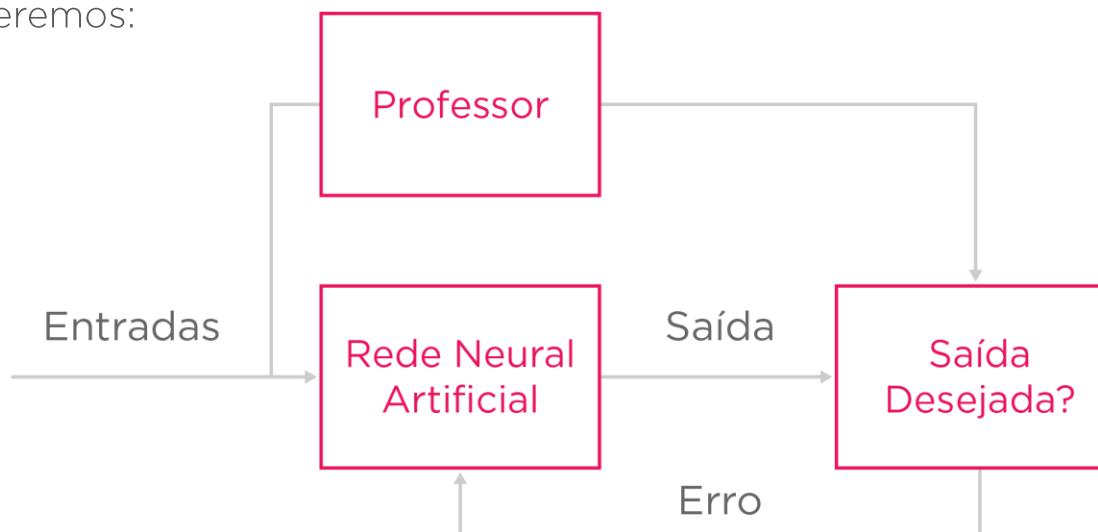
O que queremos:



# REDES NEURAIS **ARTIFICIAIS**

Aprendizado:

O que queremos:

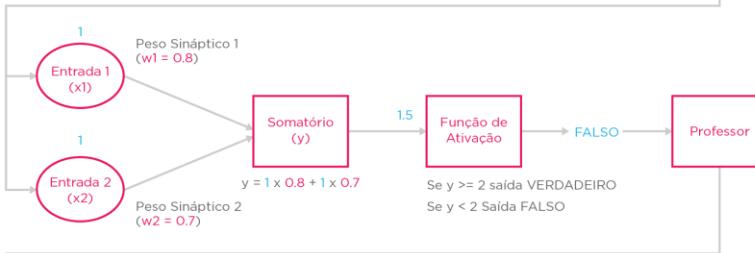


# REDES NEURAIS ARTIFICIAIS

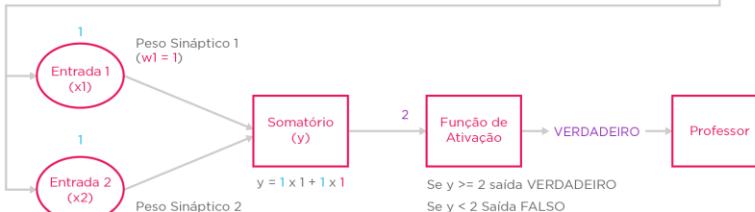


O Neurônio errou: era esperado VERDADEIRO e o Neurônio previu FALSO.  
Os pesos precisam ser atualizados!

A saída do neurônio não é a saída desejada.  
Os pesos  $w_1$  e  $w_2$  gerados aleatoriamente não foram capazes de resolver nosso problema.



Melhorou, mas ainda não é a saída desejada.



Saída do Neurônio é a Saída Desejada!  
O Neurônio foi treinado com sucesso!

# REDES NEURAIS **ARTIFICIAIS**

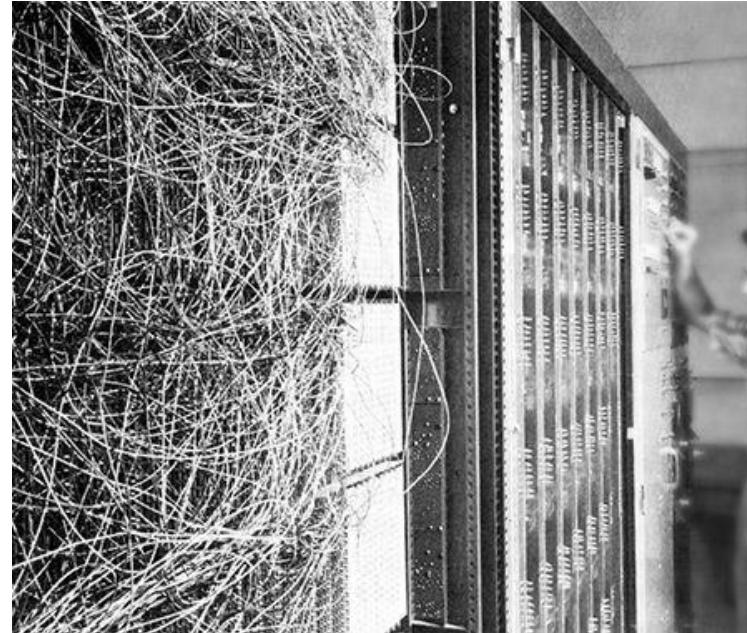
## Algoritmos de aprendizagem:

Um conjunto de regras (passos) bem definidas para solucionar o problema de aprendizado é chamada de algoritmo de aprendizado.

- Existem vários algoritmos de aprendizado diferentes.
- Eles diferem-se entre si pela maneira como ajustam (alteram) os pesos.

# REDES NEURAIS **ARTIFICIAIS**

Perceptron:



# REDES NEURAIS **ARTIFICIAIS**

## Perceptron:

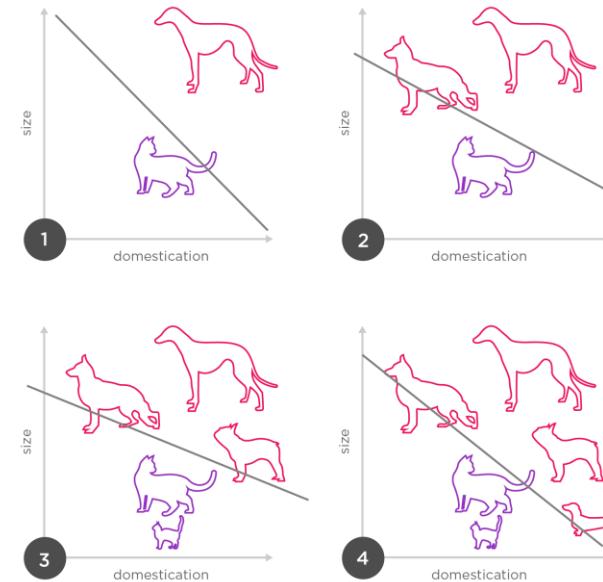
No final da década de 1950, Rosenblatt, na Universidade de Cornell, criou uma genuína rede de múltiplos neurônios do tipo discriminadores lineares e chamou essa rede de perceptron.

Perceptron pode ser visto como o tipo mais simples de rede neural feedforward: um classificador linear.

# REDES NEURAIS **ARTIFICIAIS**

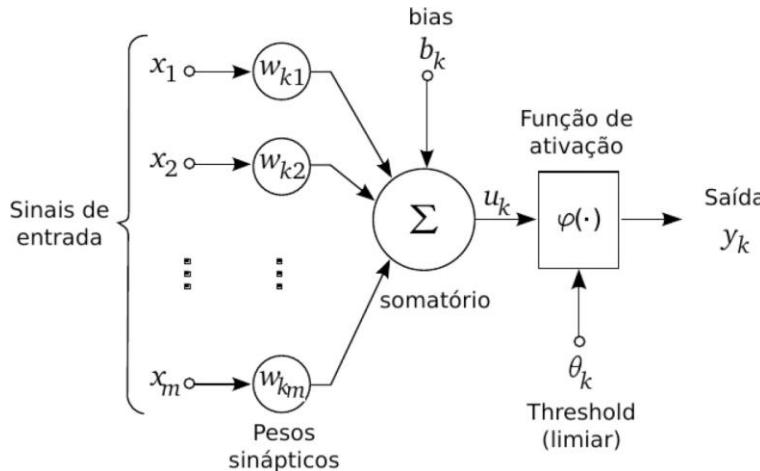
## Perceptron:

O Perceptron, proposto por Rosenblatt, é composto pelo neurônio de McCulloch-Pitts, com função de limiar e aprendizado supervisionado. Sua arquitetura consiste na entrada e uma camada de saída.



# REDES NEURAIS ARTIFICIAIS

## Perceptron:



### Considerações:

$$\vec{x} = [1, x_1, x_2, \dots, x_n]$$

$$\vec{w} = [-b, w_1, w_2, \dots, w_n]$$

$$\sum_i w_i x_i = \vec{w} \cdot \vec{x}$$

produto interno

$$\vec{w}(t+1) = \vec{w}(t) + ? \Rightarrow$$

COMO ATUALIZAR OS PESOS  
DA REDE NEURAL?

MAS ANTES...

## COMPONENTES DE **REDES NEURAIS ARTIFICIAIS**

O processo de aprendizagem pode variar de acordo como a Rede Neural é composta, por isso precisamos determinar:

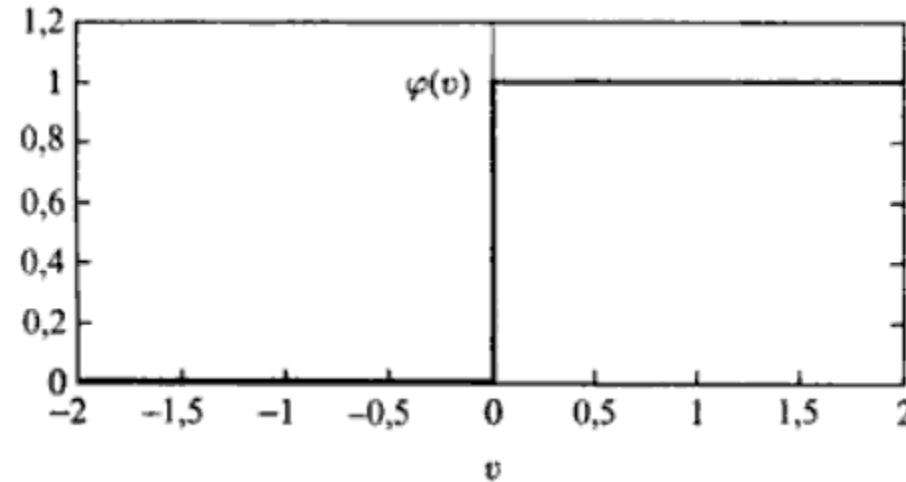
- A função de ativação;
- A arquitetura da Rede Neural
- O otimizador (gradiente);
- A taxa de aprendizagem;
- E a função de erro (também conhecida como função de perda - loss function).

## REDES NEURAIS ARTIFICIAIS - FUNÇÃO DE ATIVAÇÃO

Função de Etapa Binária ou Função Limiar

$$y_k = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases}$$

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k$$

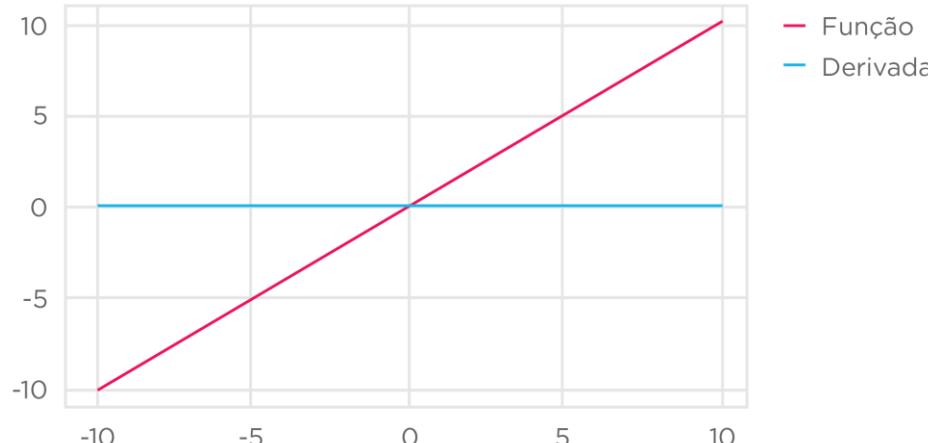


# REDES NEURAIS ARTIFICIAIS - FUNÇÃO DE ATIVAÇÃO

## Função Linear

A derivada de uma função linear é constante, isto é, não depende do valor de entrada  $x$ .

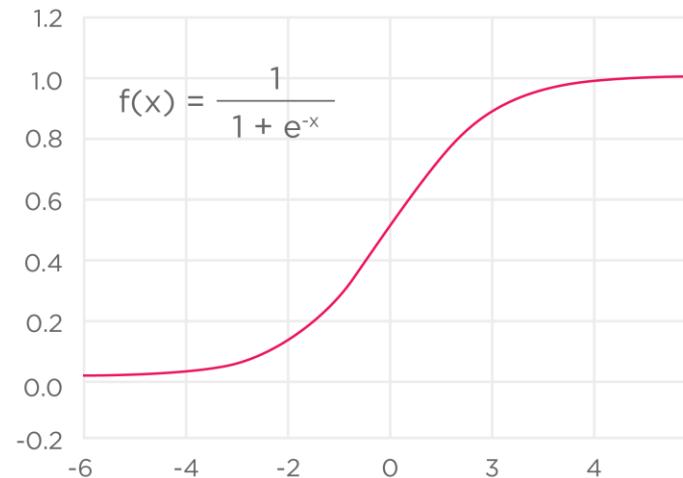
Função Linear



# REDES NEURAIS ARTIFICIAIS - FUNÇÃO DE ATIVAÇÃO

## Função Sígnioide (Sigmoid Function):

- Boa para interpretação da “força” do estímulo do neurônio.
- Função não linear consegue “afastar” os dados no eixo y.
- Trabalha apenas com valores positivos na saída, o que pode ser um problema.

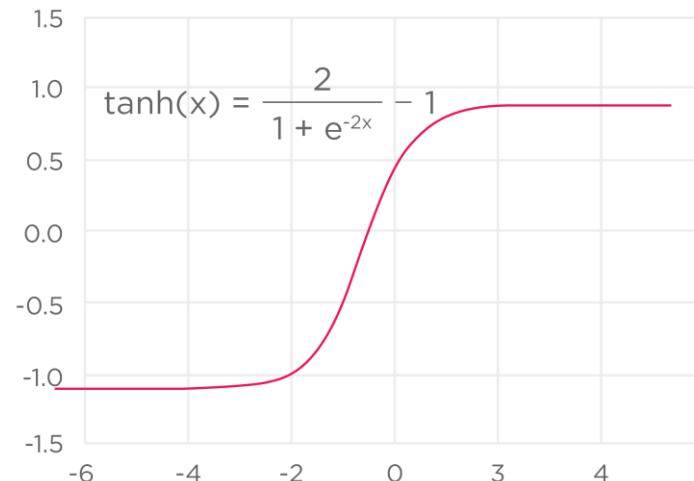


# REDES NEURAIS ARTIFICIAIS - FUNÇÃO DE ATIVAÇÃO

## Função Tangente Hiperbólica (tanh):

Não é centralizada no “zero”.

É uma versão escalonada da Função Sigmoid.

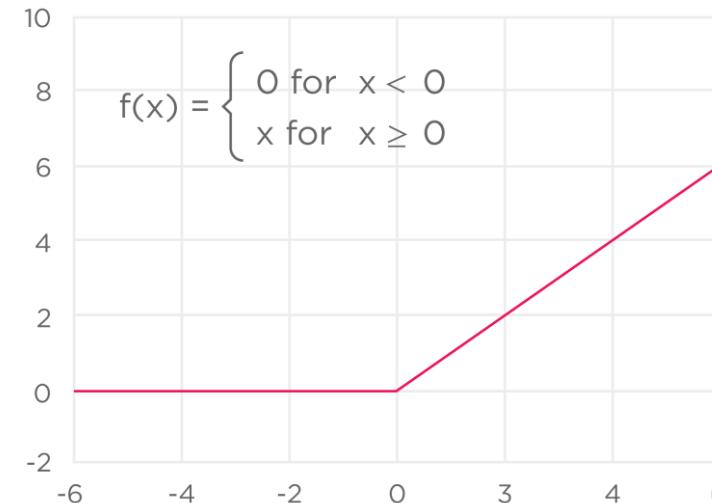


# REDES NEURAIS ARTIFICIAIS - FUNÇÃO DE ATIVAÇÃO

## Função ReLU (Unidade Linear

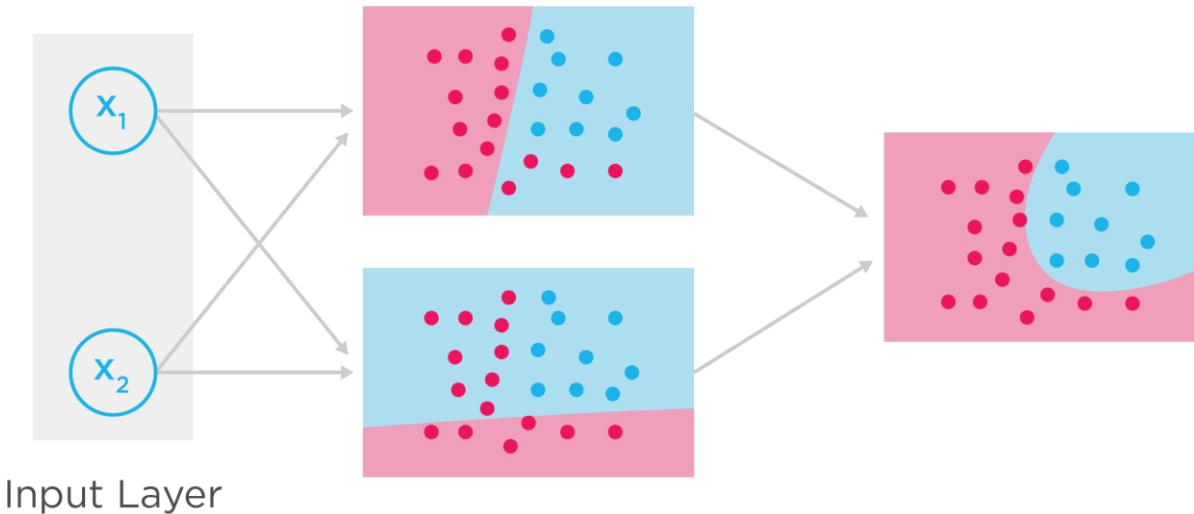
Retificada):

- Largamente utilizada em redes neurais convolucionais (Deep Learning).
- Acelera a convergência para a descida do gradiente.
- Menor complexidade computacional.
- Não ativa todos os neurônios ao mesmo tempo.



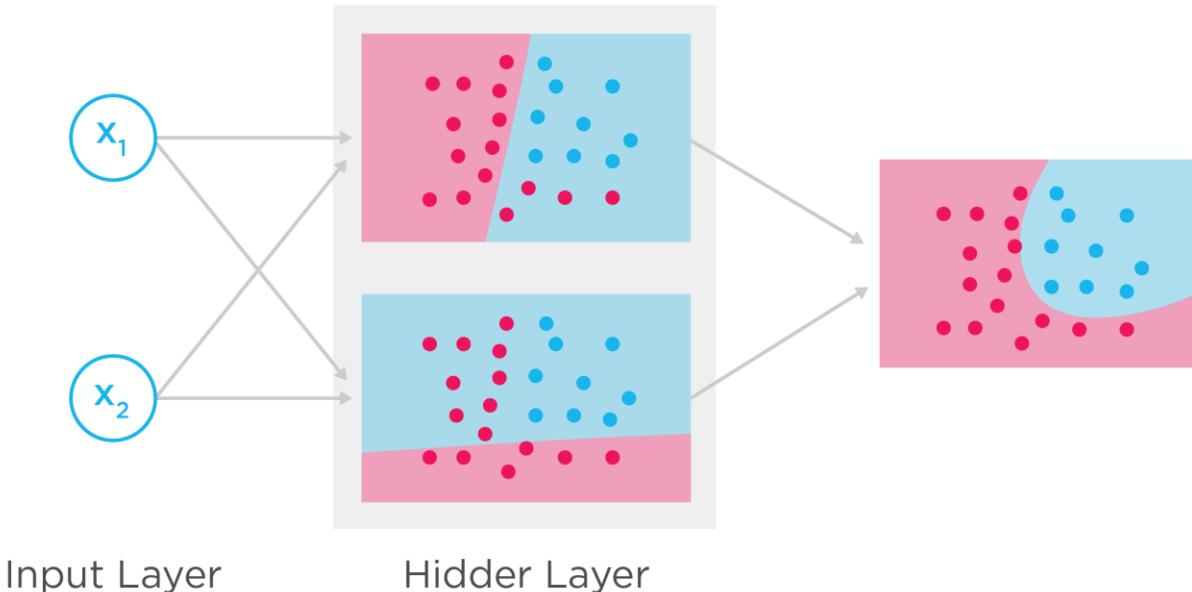
# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

## Neural Network



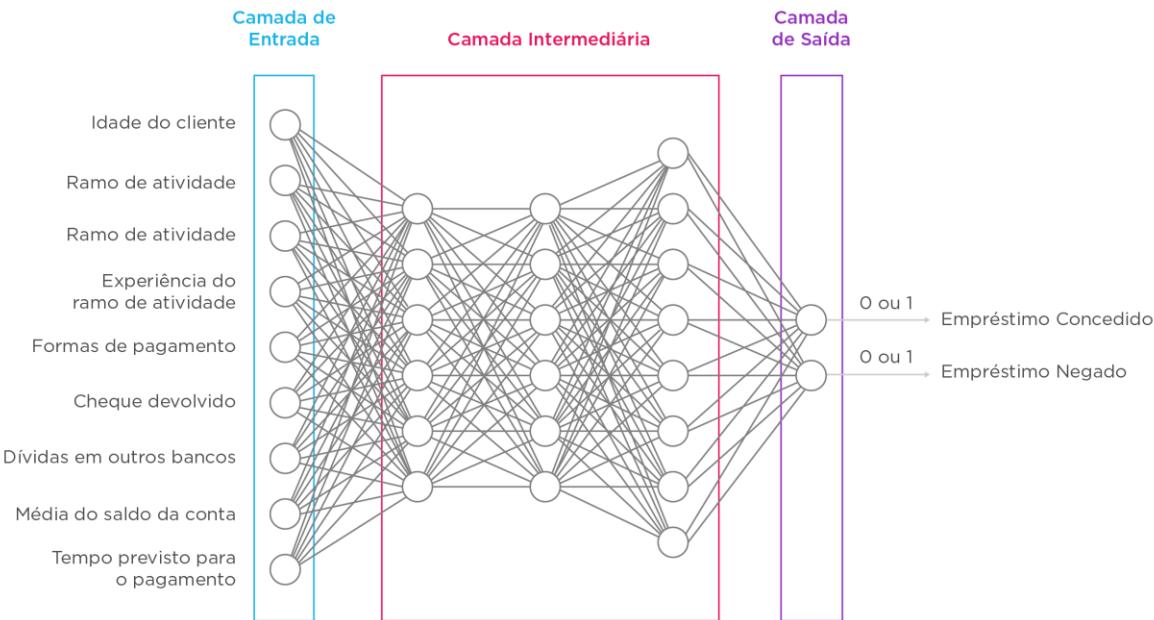
# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Neural Network



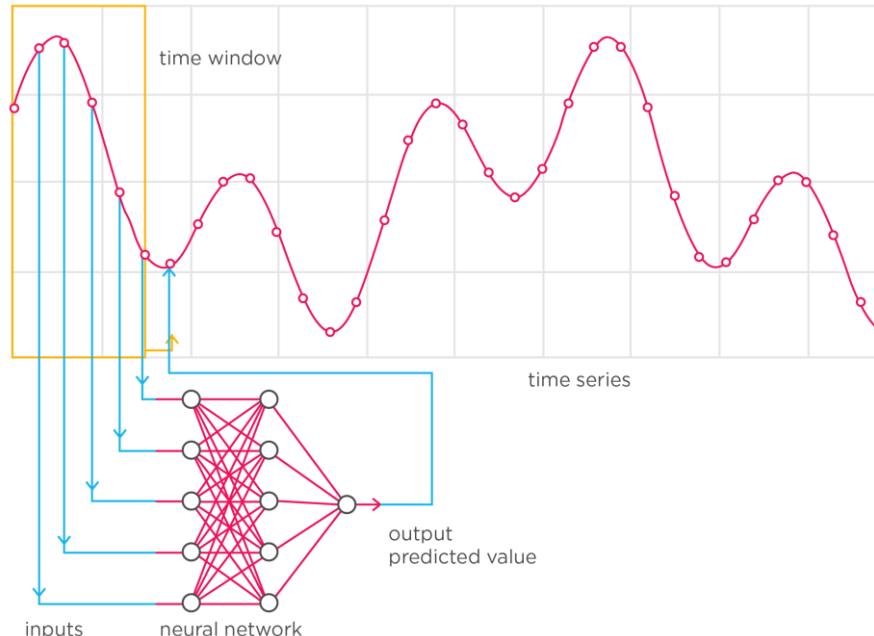
# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

## Multi Layer Perceptron (MLP) :



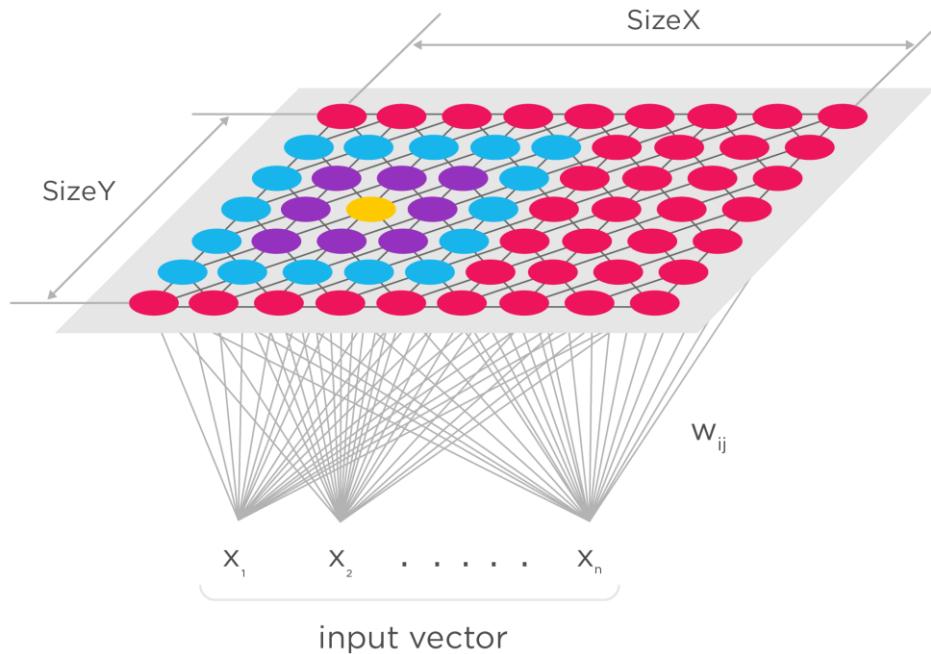
# REDES NEURAIS ARTIFICIAIS - FUNÇÃO DE ATIVAÇÃO

## Redes Recorrentes:



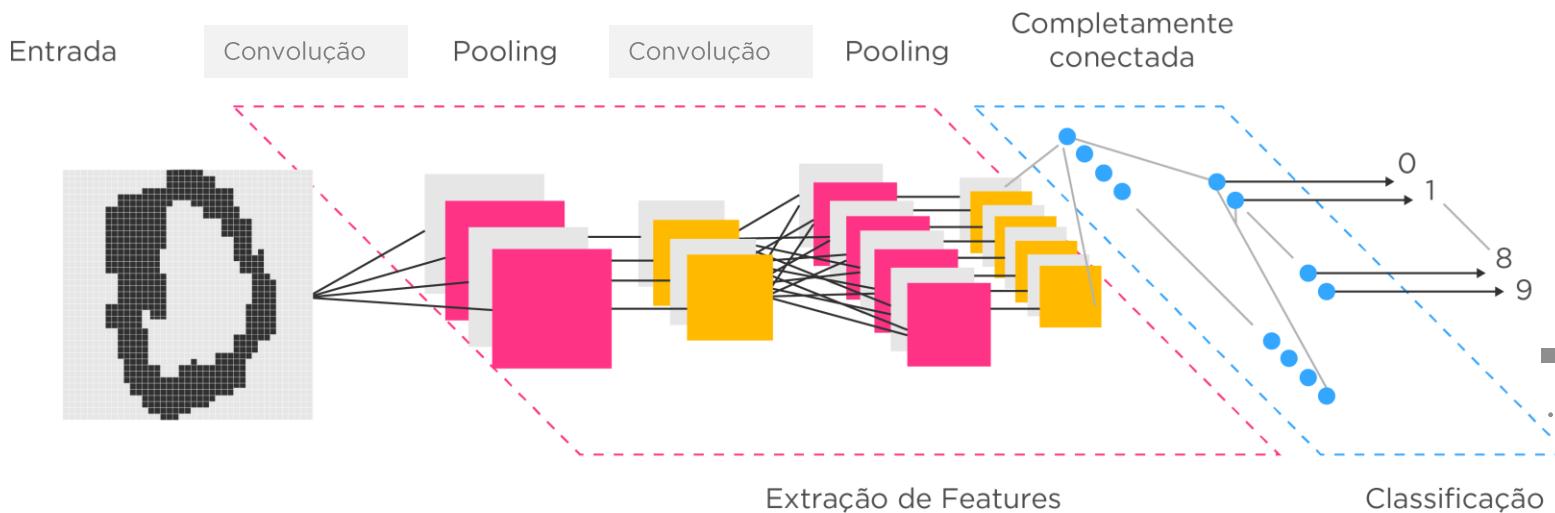
# REDES NEURAIS ARTIFICIAIS - FUNÇÃO DE ATIVAÇÃO

Self Organized Maps:



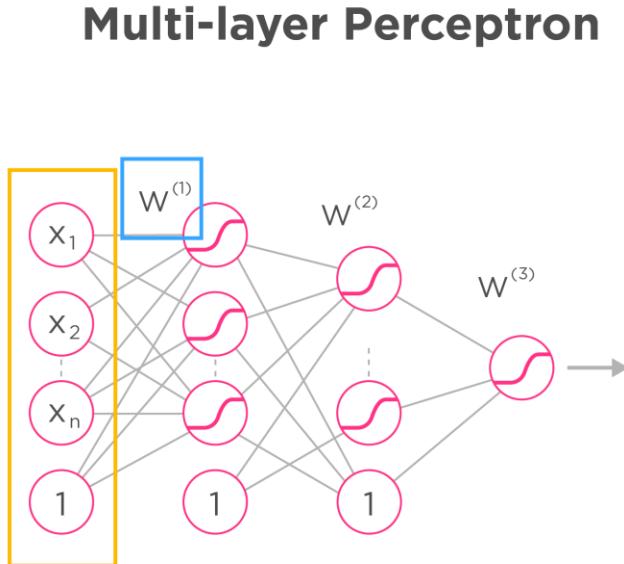
# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Redes Neurais Convolucionais :



# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Topologia FeedForward:



PREDICTION

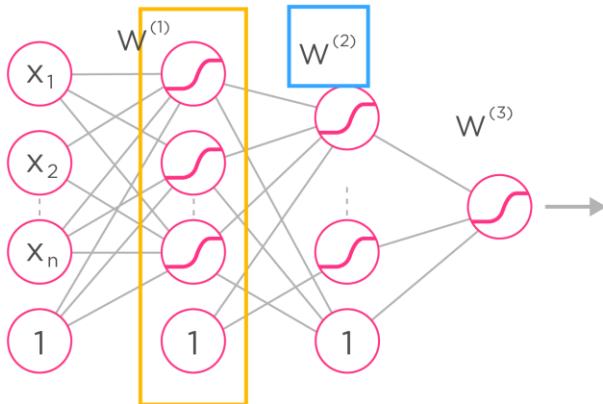
$$\hat{y} =$$

$$\sigma \circ W^{(1)}(x)$$

# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Topologia FeedForward:

## Multi-layer Perceptron



## PREDICTION

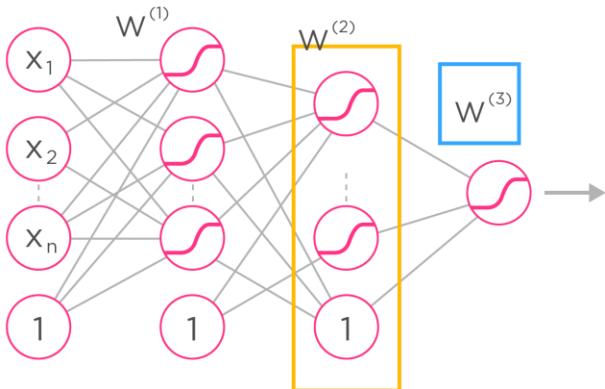
$$\hat{y} =$$

$$\sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Topologia FeedForward:

## Multi-layer Perceptron

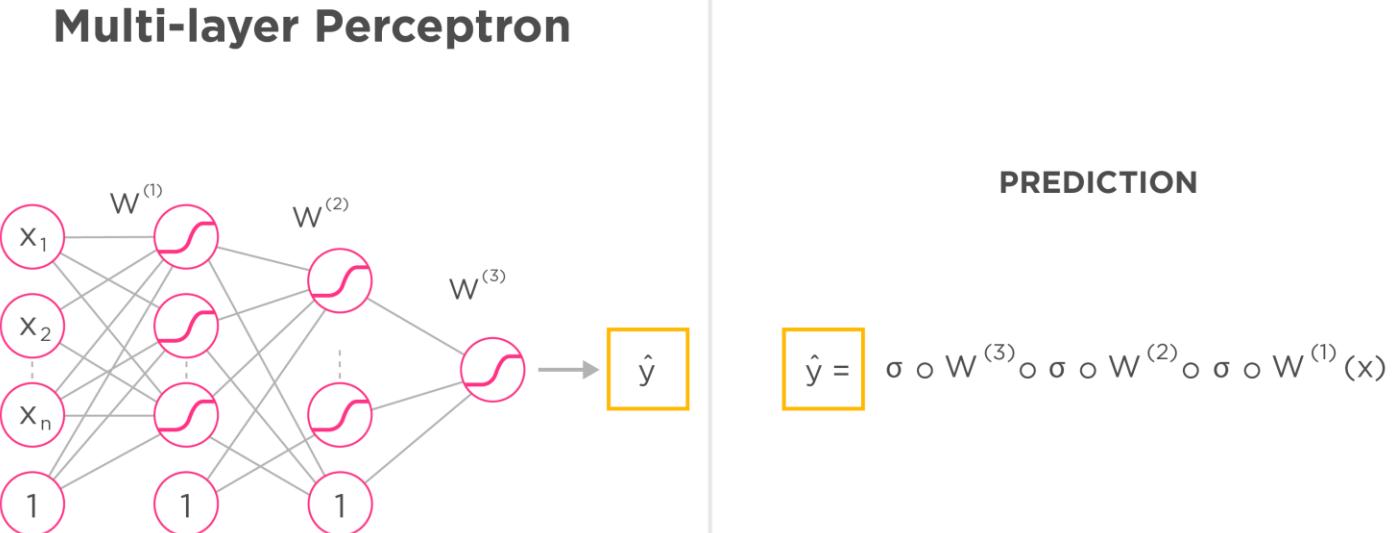


## PREDICTION

$$\hat{y} = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Topologia FeedForward:



## REDES NEURAIS **ARTIFICIAIS MULTICAMADAS**

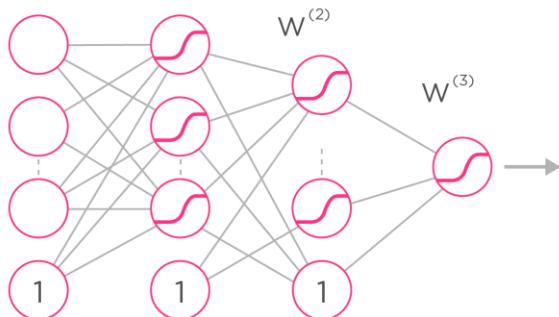
Topologia FeedForward:

Mas agora que temos diversos neurônios na rede neural, como fazer para ajustar todos os seus pesos?

# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Aprendizado e Função de Erro:

## Multi-layer Perceptron

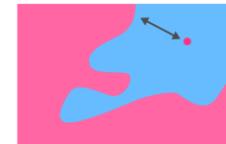


### PREDICTION

$$y = \sigma \circ W^{(3)} \circ \sigma \circ W^{(2)} \circ \sigma \circ W^{(1)}(x)$$

### ERROR FUNCTION

$$E(W) = -\frac{1}{m} \sum_{i=1}^m y_i \ln(\hat{y}_i) + (1 - y_i) \ln(1 - \hat{y}_i)$$



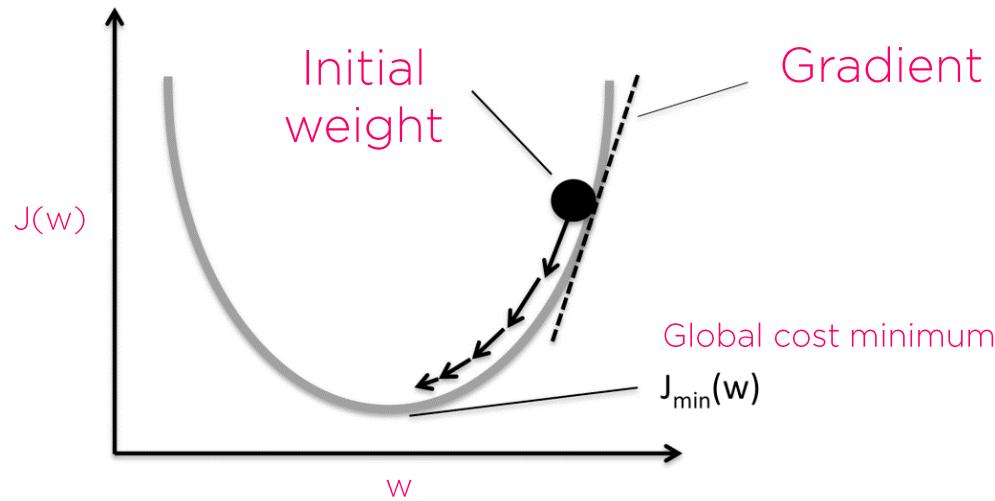
## REDES NEURAIS **ARTIFICIAIS MULTICAMADAS**

Descida do Gradiente (otimizadores)



# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente (otimizadores)



## REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente (otimizadores)

custo = avaliar (f (coeficiente))

delta = derivado (custo)

coeficiente = coeficiente - (alfa \* delta)

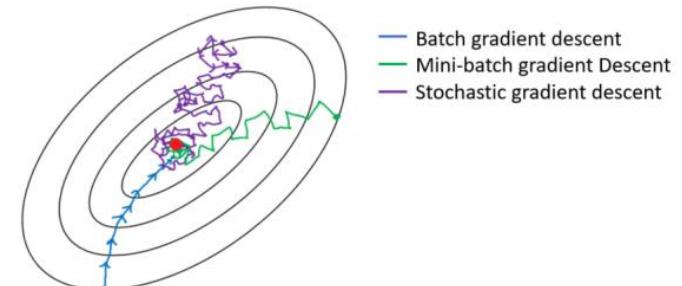
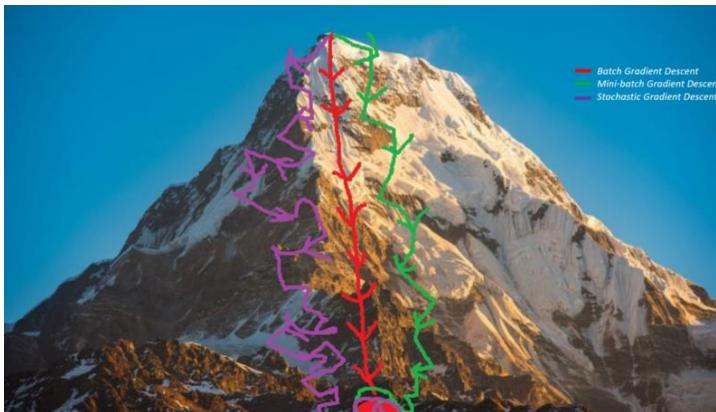
- Nossos coeficientes são o vetor de peso w e b;
- Onde f é a função de ativação;
- A função Avaliar calcula o erro da função em relação ao coeficiente;
- A derivada de f previamente calculada retorna a inclinação delta;
- Alfa (ou eta) é a taxa de aprendizagem.

# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente (otimizadores)

Descida do Gradiente – Diferentes algoritmos:

- Descida do Gradiente em “Batch”
- Descida do Gradiente Estocástico
- Descida do Gradiente em “Mini-Batch”



# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente (otimizadores)

Descida do Gradiente – Diferentes algoritmos:

- Descida do Gradiente em “Batch”
- Descida do Gradiente Estocástico
- Descida do Gradiente em “Mini-Batch”

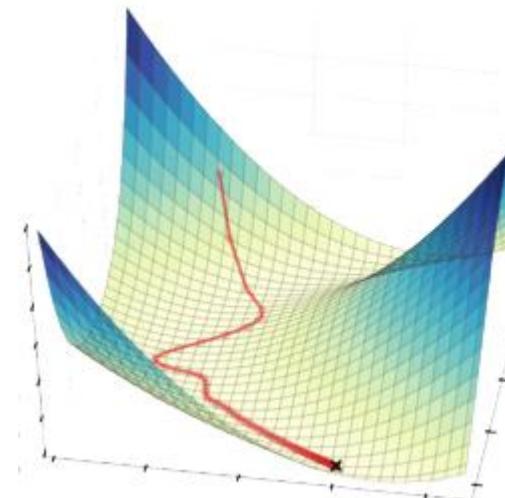
Method	Accuracy	Time	Memory Usage	Online Learning
<b>Batch</b> gradient descent	○	Slow	High	✗
<b>Stochastic</b> gradient descent	△	High	Low	○
<b>Mini-batch</b> gradient descent	○	Midium	Midium	○

# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente (otimizadores)

Descida do Gradiente – Diferentes algoritmos:  
• Descida do Gradiente com Adam Optimizer

<https://www.youtube.com/watch?v=qPKKtVkJAjY>

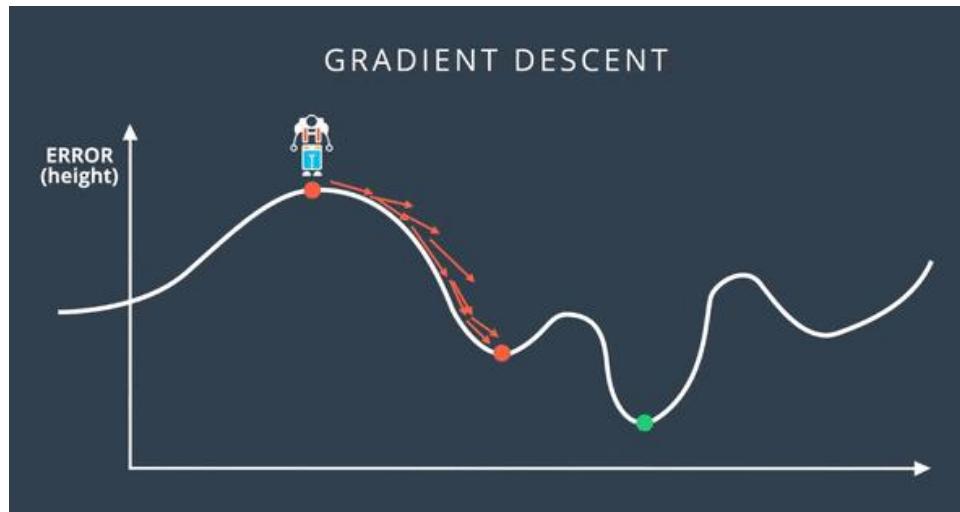


# REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente (otimizadores)

Descida do Gradiente – Diferentes algoritmos:

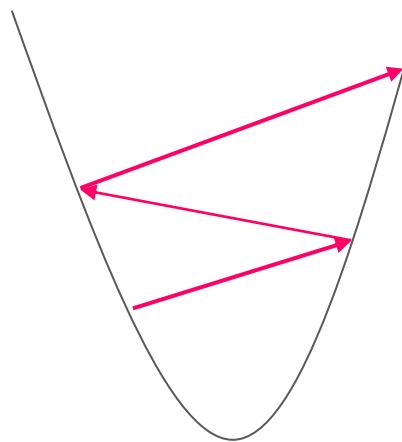
- Descida do Gradiente com Adam Optimizer



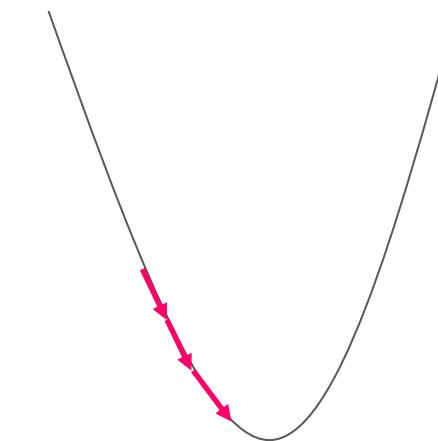
## REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente e taxa de aprendizagem

Big learning rate

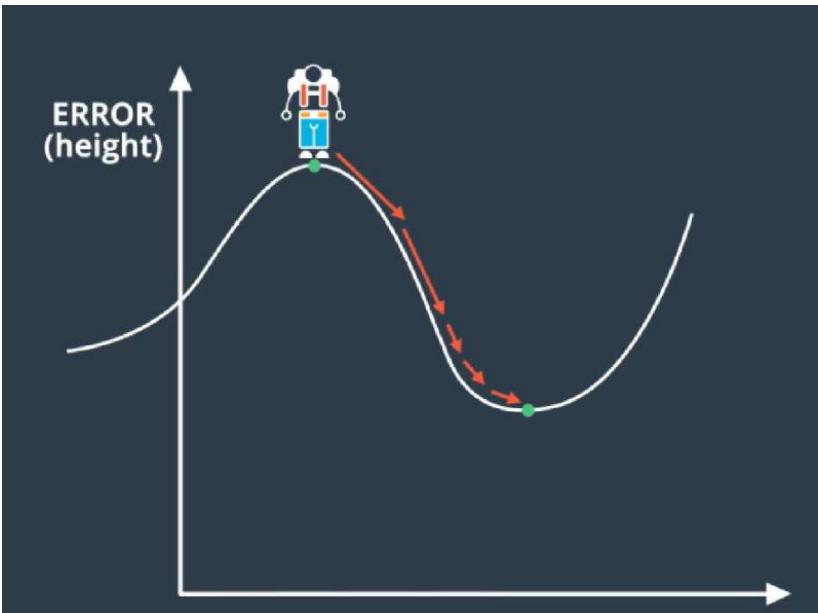


Small learning rate



## REDES NEURAIS ARTIFICIAIS MULTICAMADAS

Descida do Gradiente e taxa de aprendizagem



### Decreasing Learning Rate

Rule:

- If steep: long steps
- If plain: small steps

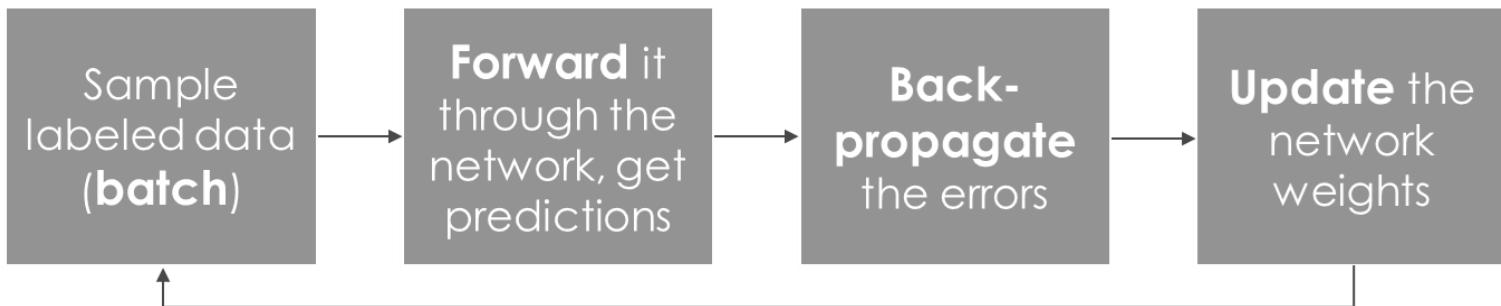
## REDES NEURAIS ARTIFICIAIS MULTICAMADAS

O processo de aprendizagem pode variar de acordo com a maneira como a Rede Neural é composta, por isso precisamos determinar:

- a função de ativação ✓
- a arquitetura da Rede Neural ✓
- o otimizador (gradiente) ✓
- a taxa de aprendizagem ✓
- A Função de Erro (também conhecida como função de perda - *loss function*) ✓

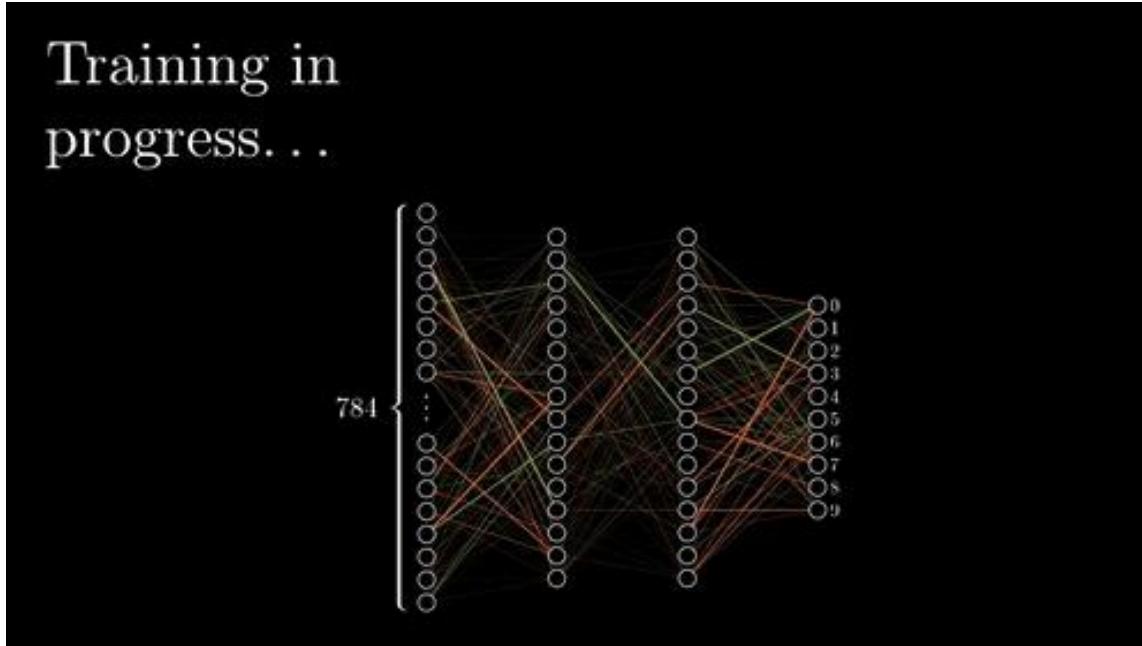
# REDES NEURAIS **ARTIFICIAIS**

Backpropagation:



# REDES NEURAIS **ARTIFICIAIS**

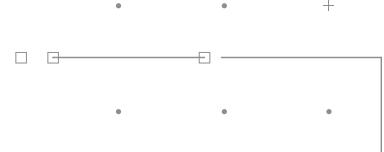
Backpropagation:



REDES NEURAIS  
**ARTIFICIAIS**

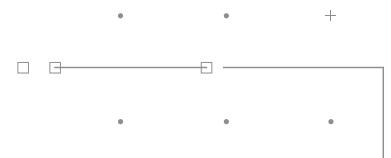


HANDS ON #1:  
**TENSORFLOW PLAYGROUND**



# REDES NEURAIS ARTIFICIAIS

HANDS ON #2:  
**REDE NEURAL MLP PARA  
CLASSIFICAÇÃO**



# REDES NEURAIS ARTIFICIAIS

## HANDS ON #3: REDE NEURAL MLP PARA REGRESSÃO

# OBRIGADO

FIAP

Copyright © 2020 | Professor Felipe Gustavo Silva Teodoro

Todos os direitos reservados. A reprodução ou divulgação total ou parcial deste documento é expressamente proibida sem consentimento formal, por escrito, do professor(a)/autor(a).

+

• • • • • .  
• • • • • .  
• + .  
+ •

|

+

# FIAP

• • +  
■ □ \*  
• • •  
• + • •