



Preditiva.ai

# Aprendizado Supervisionado

## Redes Neurais Artificiais

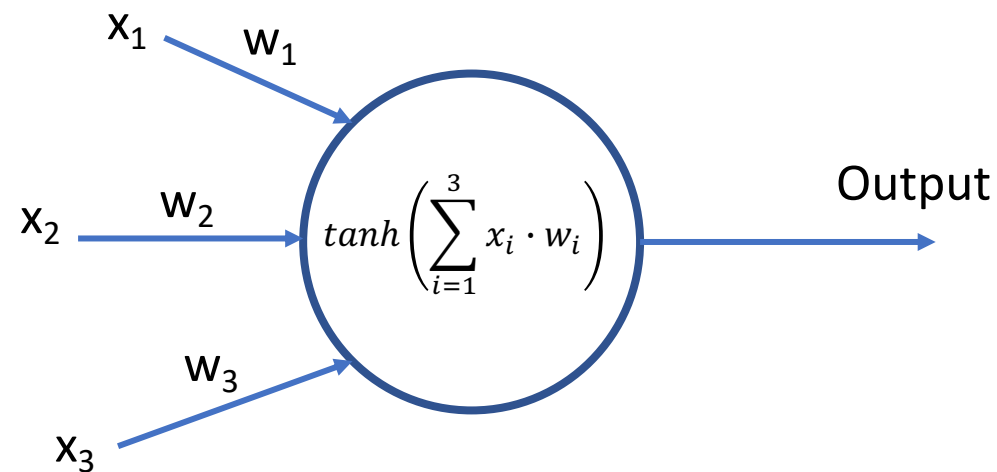
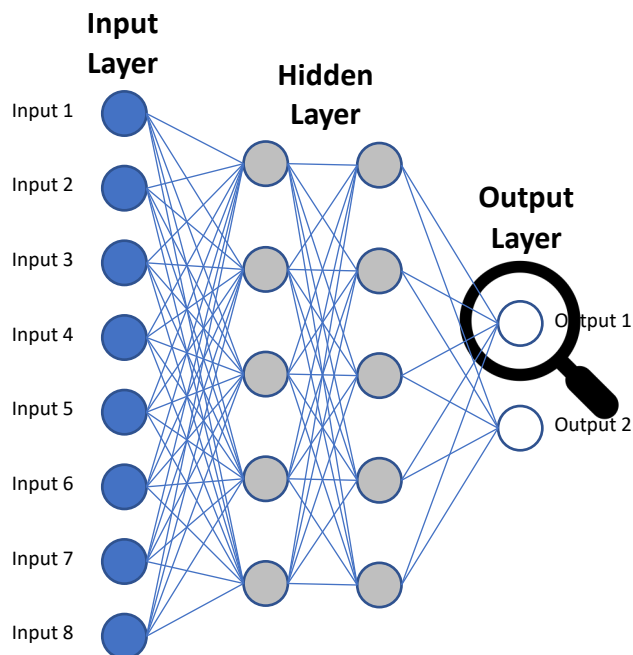
# Redes Neurais Artificiais

## Breve resumo histórico...



As primeiras propostas de um modelo computacional inspirado no cérebro humano surgiram na década de 1940, sendo que em 1960 surgiu a estrutura conhecida como **Single Layer Perceptron**, capaz de reconhecer **padrões linearmente separáveis**.

Apenas em 1970 foi proposta a estrutura de Redes Neurais Artificiais (RNA) que é uma das mais utilizadas até hoje: **Multi Layer Perceptron** ou MLP, capaz de reconhecer **padrões não-lineares**.



# Demonstração

Exemplos de utilização de uma Rede Neural Artificial

<https://playground.tensorflow.org/>

# Redes Neurais Artificiais

## Resultados importantes

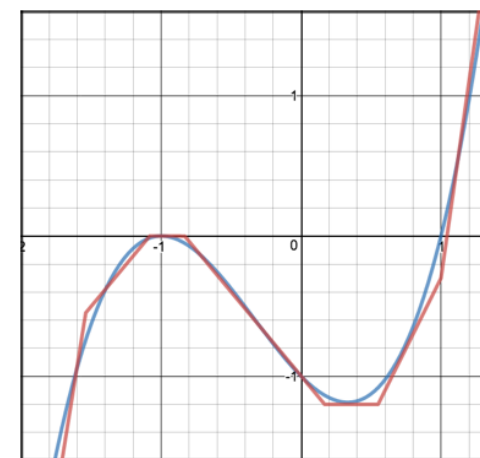


Um dos resultados mais importantes que utilizaremos nas RNA é o **Teorema da Aproximação Universal** de George Cybenko (1989) que demonstra que:

*Uma MLP com uma **única camada oculta**, um **número finito de neurônios artificiais** nessa camada e qualquer **função contínua sigmoideal** como função de ativação pode **aproximar qualquer função contínua**.*

Este teorema demonstra o **poder de captura de padrões** inerente as RNA do tipo MLP, porém não é apresentada qualquer forma de cálculo desse número finito de neurônios.

Além disso, existem diversos outros hiperparâmetros que afetam o desempenho de uma RNA MLP.



$$\begin{aligned}n_1(x) &= \text{Relu}(-5x - 7.7) \\n_2(x) &= \text{Relu}(-1.2x - 1.3) \\n_3(x) &= \text{Relu}(1.2x + 1) \\n_4(x) &= \text{Relu}(1.2x - .2) \\n_5(x) &= \text{Relu}(2x - 1.1) \\n_6(x) &= \text{Relu}(5x - 5)\end{aligned}$$

$$\begin{aligned}Z(x) &= -n_1(x) - n_2(x) - n_3(x) \\&\quad + n_4(x) + n_5(x) + n_6(x)\end{aligned}$$

# Redes Neurais Artificiais

## Hiperparâmetros

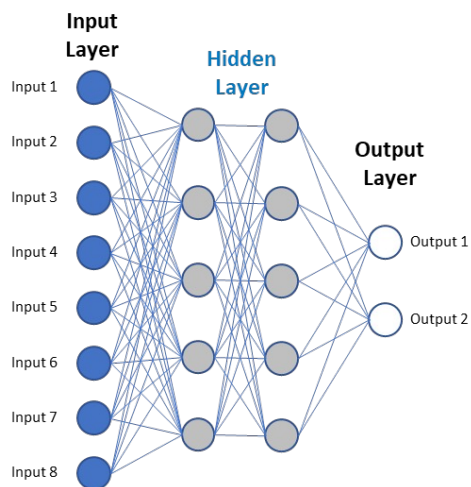


As RNA MLP possuem diversos hiperparâmetros que afetam seu desempenho e podemos organizá-los em 3 principais grupos:

### 1. Topologia

Define a estrutura da RNA MLP e ajusta sua capacidade de reconhecimento de padrões, mais ou menos complexos.

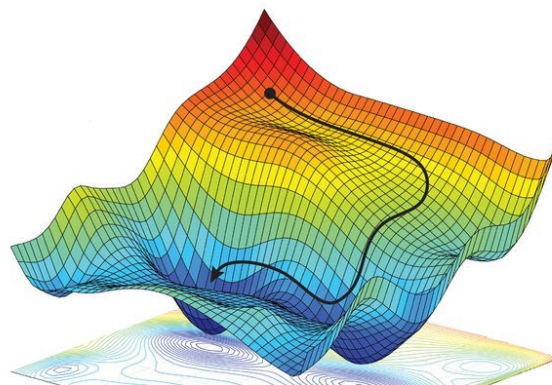
- Nº camadas
- Nº neurônios por camada
- Função de ativação



### 2. Aprendizagem

Define como a RNA MLP reconhecerá os padrões utilizando diferentes métodos de otimização dos pesos sinápticos.

- Otimizador
- Learning Rate
- Batch Size

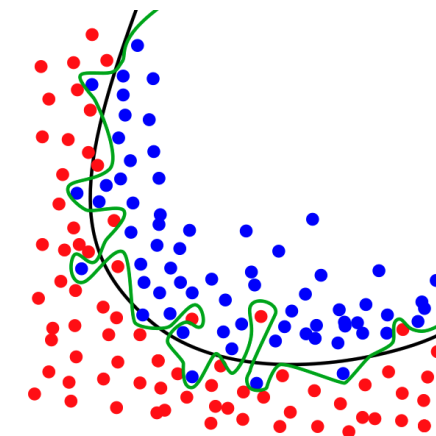


Fonte: <https://towardsdatascience.com/coding-deep-learning-for-beginners-linear-regression-gradient-descent-fcd5e0fc077d>

### 3. Regularização

Aplica restrições no treinamento e nos valores dos pesos sinápticos visando melhor capacidade de generalização.

- Dropout
- L1
- L2



Fonte: <https://laptrinhx.com/machine-learning-model-regularization-in-practice-an-example-with-keras-and-tensorflow-2-0-2844625608/>

# Redes Neurais Artificiais

## Hiperparâmetros

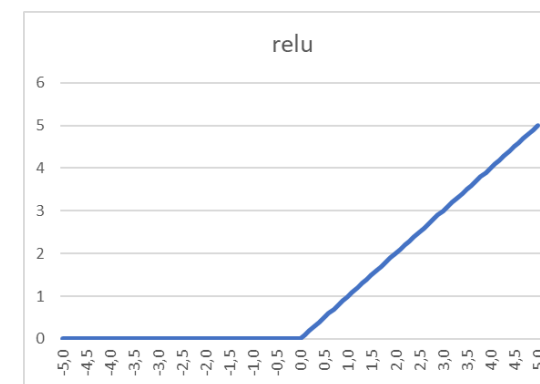
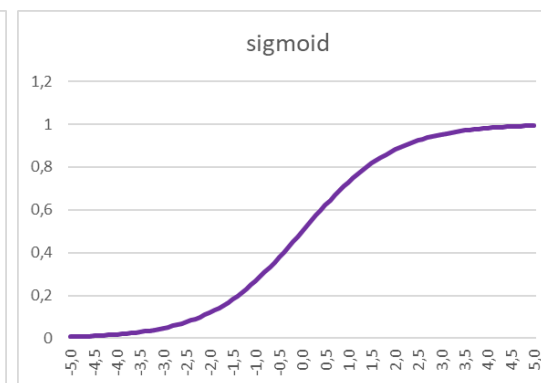
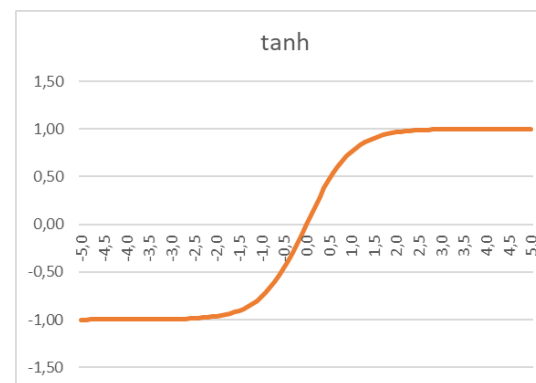
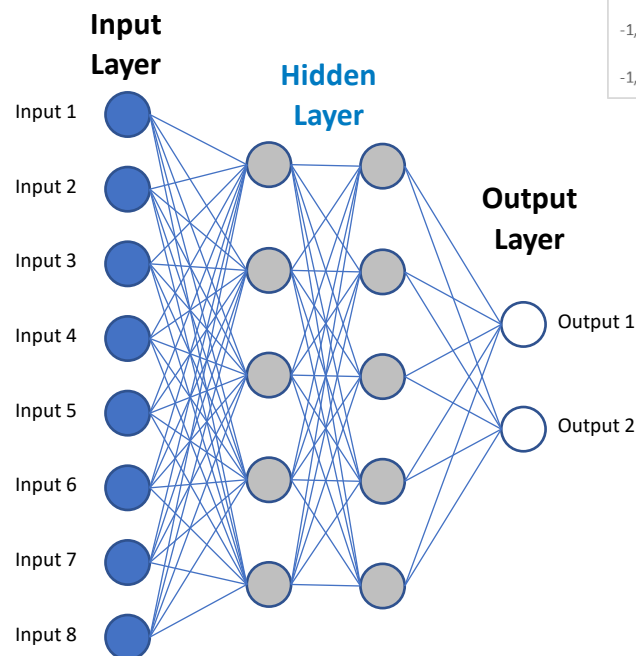


As RNA MLP possuem diversos hiperparâmetros que afetam seu desempenho e podemos organizá-los em 3 principais grupos:

### 1. Topologia

Define a estrutura da RNA MLP e ajusta sua capacidade de reconhecimento de padrões, mais ou menos complexos.

- Nº camadas
- Nº neurônios por camada
- Função de ativação



# Redes Neurais Artificiais

## Hiperparâmetros

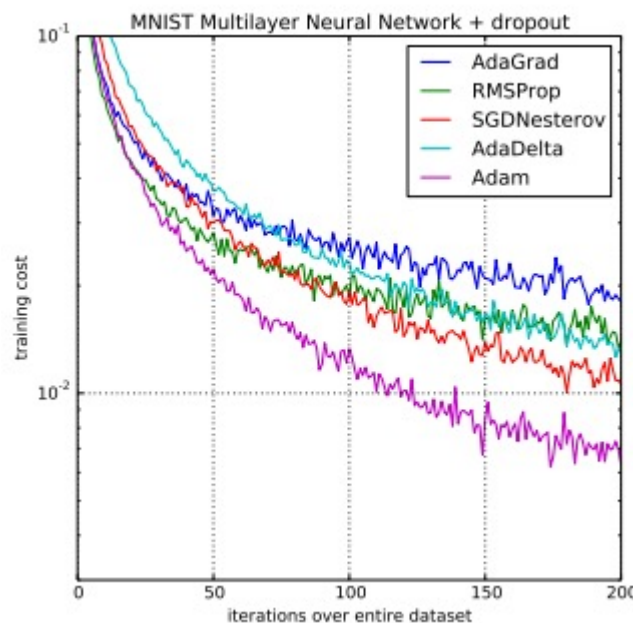


As RNA MLP possuem diversos hiperparâmetros que afetam seu desempenho e podemos organizá-los em 3 principais grupos:

### 2. Aprendizagem

Define como a RNA MLP reconhecerá os padrões utilizando diferentes métodos de otimização dos pesos sinápticos.

- Otimizador
- Learning Rate
- Batch Size



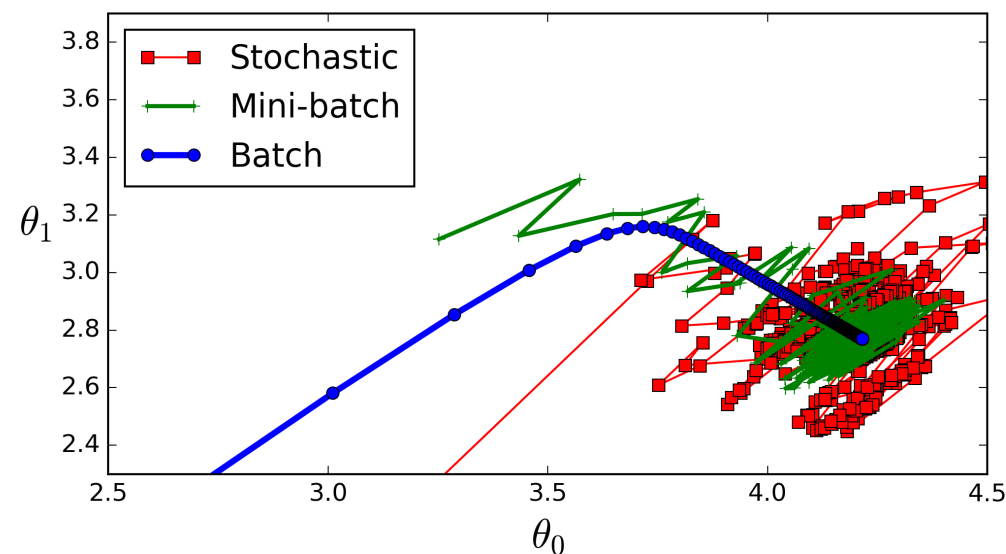
Fonte: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>

### Mini-Batch / Stochastic

Mais iterações por época  
Treinamento lento  
Gradiente mais instável

### Batch

Menos iterações por época  
Treinamento rápido  
Gradiente mais estável



Fonte: <https://stats.stackexchange.com/questions/153531/what-is-batch-size-in-neural-network>

# Redes Neurais Artificiais

## Hiperparâmetros



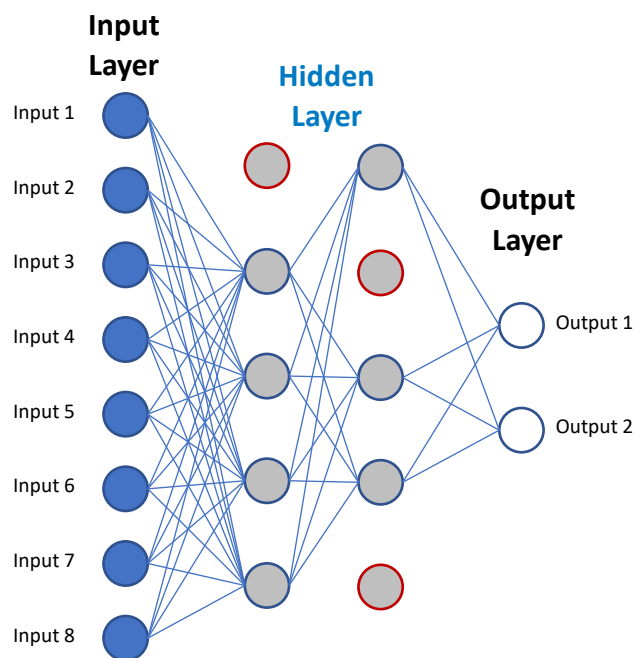
As RNA MLP possuem diversos hiperparâmetros que afetam seu desempenho e podemos organizá-los em 3 principais grupos:

### 3. Regularização

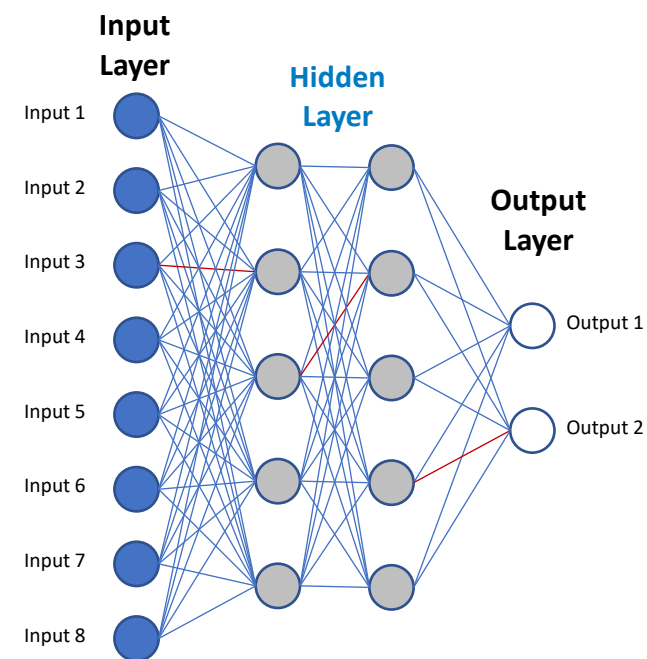
Aplica restrições no treinamento e nos valores dos pesos sinápticos visando melhor capacidade de generalização.

- Dropout
- L1
- L2

### Dropout



### L1: Encolhimento da RNA



### L2: Penalização maiores pesos

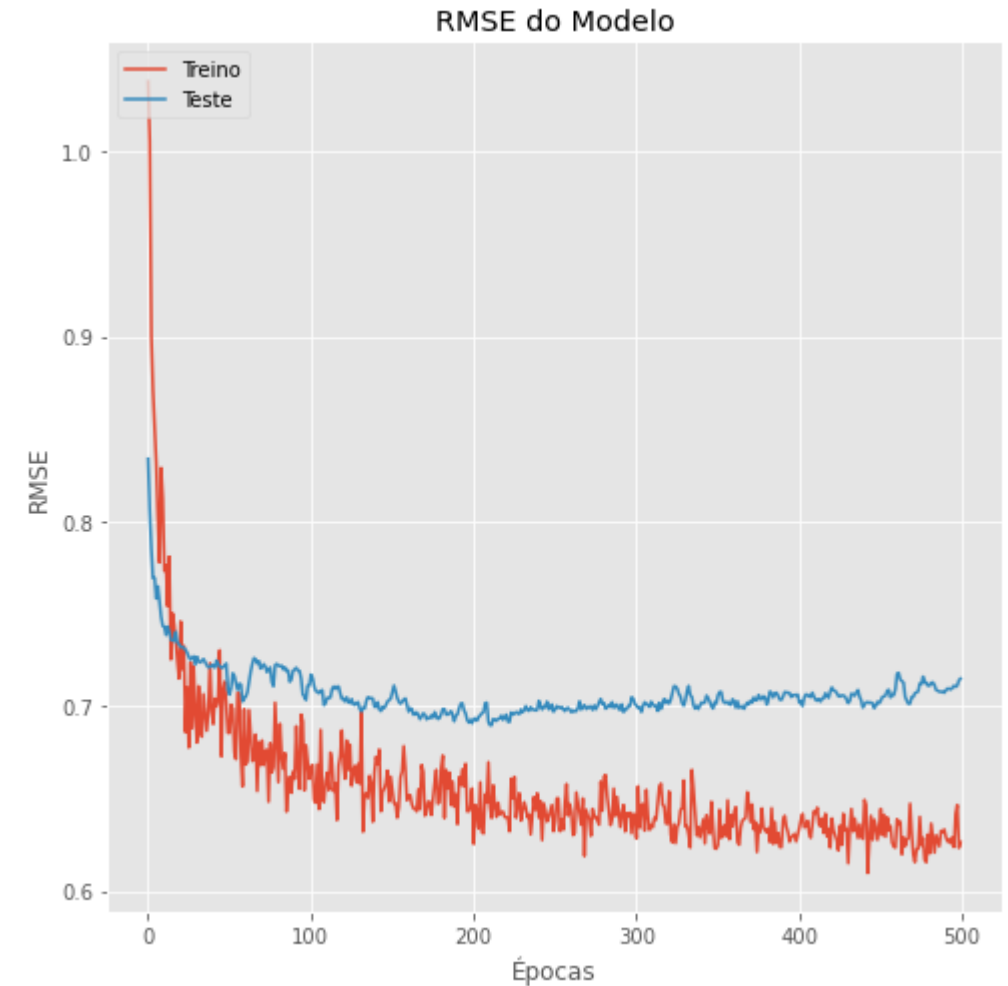
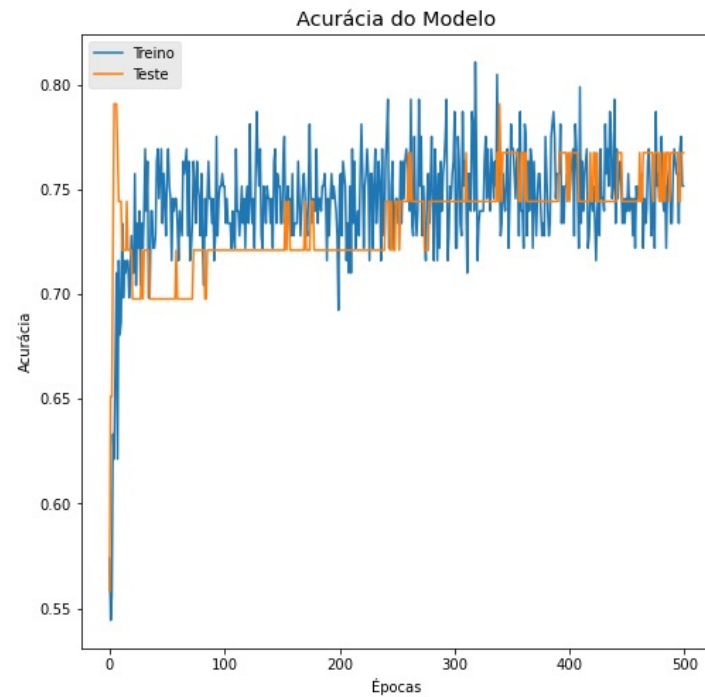


# Redes Neurais Artificiais

## Processo de Treinamento



Uma das formas mais utilizadas para avaliar o desempenho de uma RNA MLP é utilizar o histórico da **função loss** (perda/erro) ao longo das épocas.



# Demonstração

## Redes Neurais Artificiais

Arquivo: [7\\_RNA\\_Classificacao\\_Demo.ipynb](#)



## Hands on

**Aumente o desempenho da RNA MLP apresentada na demonstração, mantendo a capacidade de generalização**

Roteiro:

1. Avalie a possibilidade de criar novas variáveis usando *feature engineering*.
2. Ajuste os hiperparâmetros aprendidos para controlar:
  1. **Generalização**: overfit vs. underfit
  2. **Desempenho** na aprendizagem
3. Estime a probabilidade de doença cardíaca em um paciente com as seguintes características:

age	sex	cp	trestbps	chol	fbs	restecg
35	0	1	123	280	0	1
thalach	exang	oldpeak	slope	ca	tal	
162	1	1.8	2	2	normal	

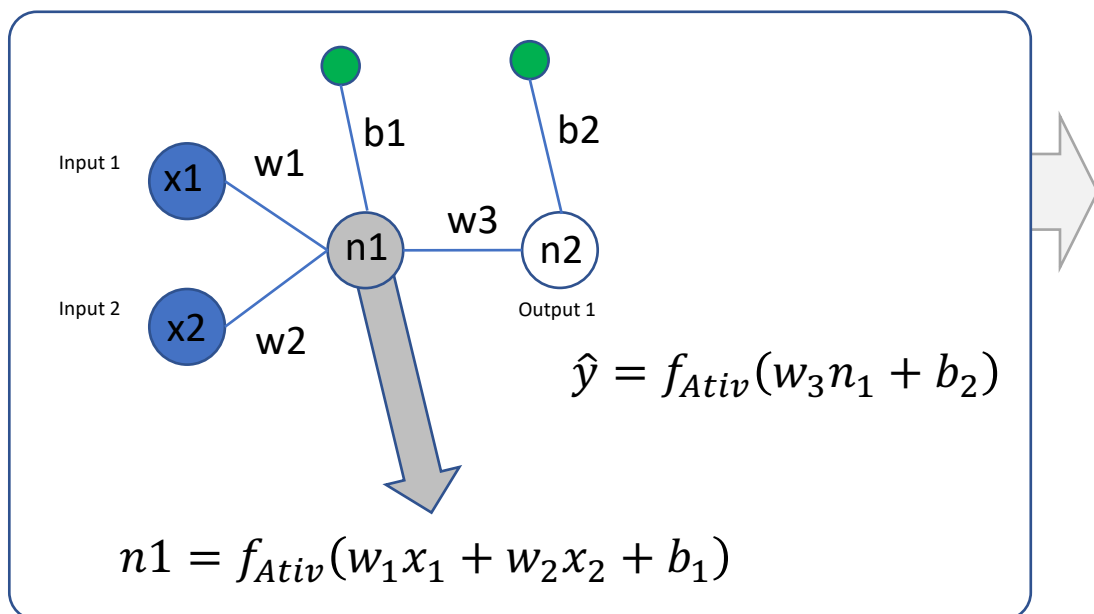
# Redes Neurais Artificiais

## Backpropagation



O método **Backpropagation (BP)** é a base dos diversos métodos de aprendizagem, ou otimização dos pesos sinápticos, utilizados.

A ideia principal por trás do **BP** é de calcular o erro entre o **Output Real** e o **Output Estimado** e utilizando os gradientes, **propagar o ajuste dos pesos sinápticos** desde a camada de Output até a camada Input.



$$\hat{y} = f_{Ativ}(\overbrace{w_3 f_{Ativ}(w_1x_1 + w_2x_2 + b_1)}^{n1} + b_2)$$

Existem diversas funções erro:

$$f_{Erro} = f(y, \hat{y}) = (\hat{y} - y)^2$$

# Redes Neurais Artificiais

## Backpropagation



Preditiva.ai

$$\hat{y} = f_{Ativ}(\mathbf{w}_3 f_{Ativ}(\mathbf{w}_1 x_1 + \mathbf{w}_2 x_2 + b_1) + b_2)$$

$$\frac{\partial \hat{y}}{\partial w_3} = f_{Ativ}(\mathbf{w}_3 n_1 + b_2)' \cdot n_1$$

Se  $f_{Ativ} = \text{sigmoid}$

$$\frac{\partial \hat{y}}{\partial w_3} = \text{sigm}(\mathbf{w}_3 n_1 + b_2) \cdot (1 - \text{sigm}(\mathbf{w}_3 n_1 + b_2)) \cdot n_1$$

$$\frac{\partial \hat{y}}{\partial w_3} = n_2 \cdot (1 - n_2) \cdot n_1$$

# Redes Neurais Artificiais

## Funções de Ativação, Erro e Métricas



As **Funções de Ativação** na camada de saída, as **Funções Erro** e as **Métricas de Desempenho** das RNA MLP devem ser escolhidas de acordo com o tipo de **Target**:

Target	Funções Ativação (Camada Saída)	Funções Loss	Métricas de Desempenho
Categórico Binário	Sigmoid	Binary Cross Entropy	AUC Accuracy Binary Accuracy Binary Cross Entropy
Categórico Multiclasse	Sigmoid	Categorical Cross Entropy	Categorical Cross Entropy
Numérico	Relu	Mean Absolute Error Mean Squared Error	Mean Absolute Error Mean Squared Error

# Demonstração

**Redes Neurais Artificiais**

Arquivo: [7\\_RNA\\_Regressao\\_Demo.ipynb](#)



Preditiva.ai