

# Redes Neurais Artificiais e Deep Learning

*Prof. Edson Pedro Ferlin*

## O que são Redes Neurais Artificiais?

Modelos computacionais inspirados no cérebro humano.

Aprendem e realizam tarefas complexas.

Deep Learning: redes com múltiplas camadas.

## Arquitetura Básica de uma Rede Neural



1

Perceptron

Modelo mais simples, uma camada.

2

Camada de Entrada

Recebe os dados.

3

Camada Oculta

Processamento dos dados.

4

Camada de Saída

Gera a previsão final.

## Componentes de uma Rede Neural

1

Neurônios

Processam os dados de entrada.

2

Pesos (Weights)

Força das conexões.

3

Função de Ativação

Determina a ativação (ReLU, Sigmoid, Tanh).

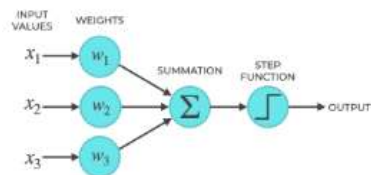
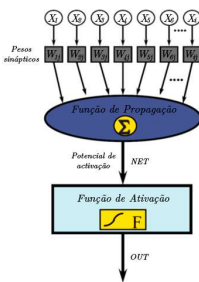
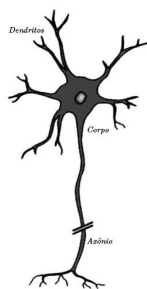
4

Bias

Valor adicional que ajuda a rede a aprender.

# Perceptron

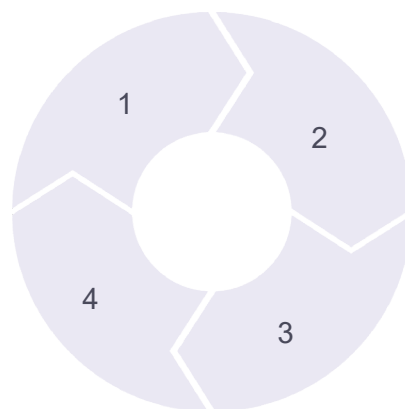
Simulam o funcionamento do Neurônio



# Estrutura de uma Rede Neural simples

**Estrutura**  
Uma única camada de neurônios.

**Ajuste**  
Treinamento ajusta pesos com base no erro.



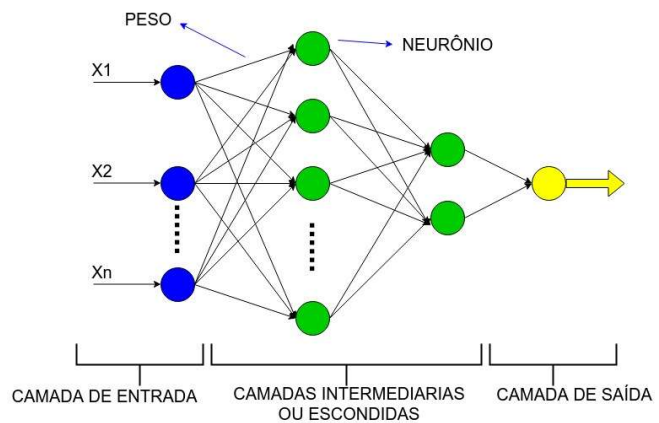
**Processamento**  
Entradas multiplicadas por pesos e somadas.

**Ativação**  
Resultado passa por função de ativação.

# Rede Neural

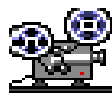
## Aprendizagem Supervisionada

Simulam o funcionamento do cérebro



# Rede Neural Artificial

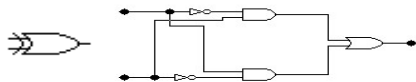
Assista o vídeo sobre a técnica Rede Neural  
(link: <https://youtu.be/Ut394JRyX9w>).



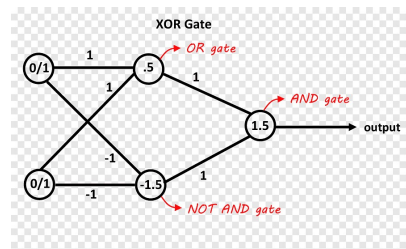
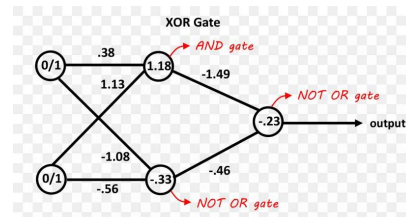
# Exemplo de uma Rede Neural: Função Lógica XOR



# Exemplo: Função Lógica XOR



A	B	X = (A ⊕ B)
0	0	0
0	1	1
1	0	1
1	1	0





## Como as Redes Neurais Aprendem?

### Treinamento

Ajusta os pesos com base nos dados fornecidos.

### Backpropagation

Ajusta os pesos e reduz o erro.

### Gradiente Descendente

Otimização para minimizar o erro.

## TensorFlow e Keras

### TensorFlow

Framework de código aberto.

### Keras

API de alto nível do TensorFlow.



## Por que usar TensorFlow e Keras?

E

Facilidade de uso

Interface simplificada.



Desempenho

Otimizado para  
grandes volumes de  
dados.



Flexibilidade

Redes neurais  
personalizadas.

## Construção de uma Rede Neural Simples usando TensorFlow e Keras

### 1. Importar Bibliotecas

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
```

### 2. Criar o Modelo

```
model = Sequential()
```

### 3. Adicionar Camadas

```
model.add(Dense(units=10, activation='relu', input_dim=2))
model.add(Dense(units=1, activation='sigmoid'))
```

### 4. Compilar o Modelo

```
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

## Exemplo de uma Rede Neural Função XOR

**1. Gerar dados:** Criamos um problema simples de **XOR** com 2 entradas e 1 saída.

**2. Modelo Sequencial:** Usamos o **Sequential** do Keras para adicionar as camadas.

1. Primeira camada oculta com **10 neurônios** e função de ativação **ReLU**.
2. Camada de saída com **1 neurônio** e função de ativação **Sigmoid** (para problema binário). **Compilação:** Utilizamos o **Adam** como otimizador e **binary\_crossentropy** como função de perda para classificação binária.

**3. Treinamento:** A rede é treinada por **100 épocas**.

**4. Avaliação:** O modelo é avaliado nos dados de teste.

```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split
import numpy as np

# Gerando dados de exemplo (2 features e 1 target)
X = np.array([[0, 0], [0, 1], [1, 0], [1, 1]]) # Dados de entrada
y = np.array([0, 1, 1, 0]) # Rótulos (XOR problem)

# Dividir os dados em treino e teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

# Criar o modelo
model = Sequential()
model.add(Dense(units=10, activation='relu', input_dim=2))
model.add(Dense(units=1, activation='sigmoid'))

# Compilar o modelo
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

# Treinar o modelo
model.fit(X_train, y_train, epochs=100)

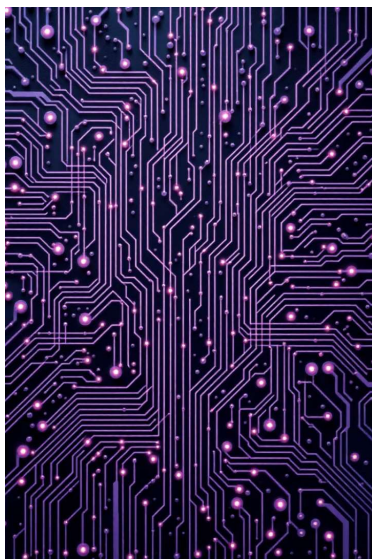
# Avaliar o modelo
loss, accuracy = model.evaluate(X_test, y_test)
print(f'Acurácia: {accuracy}')
```

## Treinamento de uma Rede Neural

1	Epochs Número de passagens.
2	Batch Size Exemplos por iteração.
3	Overfitting Evitar a especificidade.

## Exemplo de Classificação

Na publicação (**Treine sua primeira rede neural: classificação básica**)  
(link: <https://www.tensorflow.org/tutorials/keras/classification?hl=pt-br>)  
temos exemplo utilizando rede neural para classificação.



## Desafios no Treinamento

- 1 — Overfitting  
Memoriza os dados.
- 2 — Escalonamento  
Normalização melhora o desempenho.
- 3 — Hiperparâmetros  
Ajustar neurônios, camadas, taxa.

## Contato



[eferlin@live.com](mailto:eferlin@live.com)



(BLOG) [professorferlin.blogspot.com](http://professorferlin.blogspot.com)

(SITE) [professorferlin.webnode.com.br](http://professorferlin.webnode.com.br)

(YOUTUBE) [ProfEdsonPedroFerlin](https://www.youtube.com/ProfEdsonPedroFerlin)