

# Redes Neurais Artificiais

Perceptron

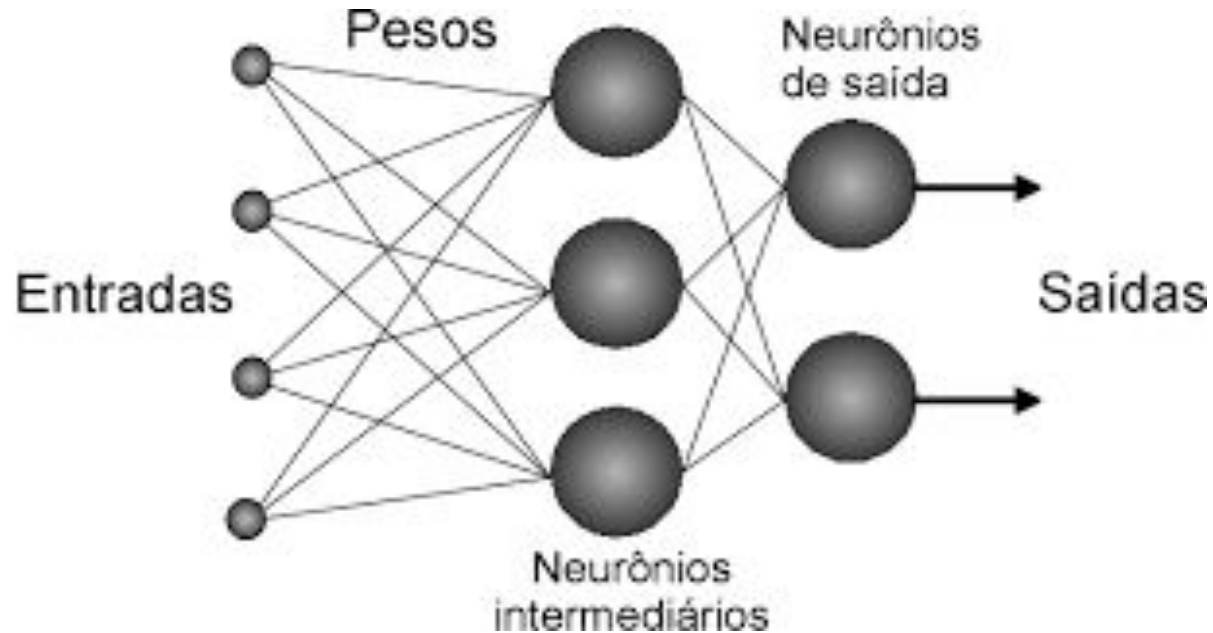
# Roteiro

- Introdução ao modelo Perceptron;
- Lógica Perceptron;
- Exemplo abordado;

# 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 esta rede de perceptron. Um perceptron é uma rede com os neurônios dispostos em camadas. Este pode ser considerado o primeiro modelo de redes neurais.
- Perceptron aprende conceitos, ele pode aprender a responder com verdadeiro (1) ou falso (0) pelas entradas que nós apresentamos a ele, “estudando” repetidamente os exemplos que lhe são apresentados.
- O perceptron é uma rede neural cujos os pesos e inclinações podem ser treinados para produzir um vetor alvo que quando apresentamos tem que corresponder ao vetor de entrada.

# Modelo Esquemático de Rede Neural



# Lógica do Modelo Perceptron

- 1º Passo: inicializar pesos e limites com zero;
- 2º Passo: apresentar nova entrada que vai se somar a saída desejada;
- 3º Passo: calcular saída atual;
- 4º Passo: atualizar o peso. Esta atualização é feita através da fórmula:

$$w_i(t+1) = w_i(t) + n[(d(t) - Y(t))] * x_i$$

$$0 \leq i \leq n-1$$

$d(t) = 1$  , se a entrada for classe A

$d(t) = -1$  , se a entrada for classe B

# Algoritmo Perceptron

## Início {Algoritmo Perceptron – Fase de Treinamento}

- <1> Obter o conjunto de amostras de treinamento  $\{x^{(k)}\}$ ;
  - <2> Associar a saída desejada  $\{d^{(k)}\}$  para cada amostra obtida;
  - <3> Iniciar o vetor  $w$  com valores aleatórios pequenos;
  - <4> Especificar a taxa de aprendizagem  $\{\eta\}$ ;
  - <5> Iniciar o contador de número de épocas  $\{\text{época} \leftarrow 0\}$ ;
  - <6> Repetir as instruções:
    - <6.1> erro  $\leftarrow$  "inexiste";
    - <6.2> Para todas as amostras de treinamento  $\{x^{(k)}, d^{(k)}\}$ , fazer:
      - <6.2.1>  $u \leftarrow w^T \cdot x^{(k)}$ ;
      - <6.2.2>  $y \leftarrow \text{sinal}(u)$ ;
      - <6.2.3> Se  $y \neq d^{(k)}$ 
        - <6.2.3.1> Então  $\begin{cases} w \leftarrow w + \eta \cdot (d^{(k)} - y) \cdot x^{(k)} \\ \text{erro} \leftarrow \text{"existe"} \end{cases}$
    - <6.3>  $\text{época} \leftarrow \text{época} + 1$ ;
- Até que: erro  $\leftarrow$  "inexiste"

Fim {Algoritmo Perceptron – Fase de Treinamento}

Após o treinamento, temos a fase de operação:

## Início {Algoritmo Perceptron – Fase de Operação}

- <1> Obter uma amostra a ser classificada  $\{x\}$ ;
- <2> Utilizar o vetor  $w$  ajustado durante o treinamento;
- <3> Executar as seguintes instruções:
  - <3.1>  $u \leftarrow w^T \cdot x$ ;
  - <3.2>  $y \leftarrow \text{sinal}(u)$ ;
  - <3.3> Se  $y = -1$ 
    - <3.3.1> Então: amostra  $x \in \{\text{Classe A}\}$
  - <3.4> Se  $y = 1$ 
    - <3.4.1> Então: amostra  $x \in \{\text{Classe B}\}$

Fim {Algoritmo Perceptron – Fase de Operação}

# Exemplo Proposto

## 2.7 Exercício com dois neurônios na camada de saída

Esse exemplo tem como objetivo construir um sistema para uma livraria com base no *Perceptron* capaz de classificar automaticamente os clientes da livraria em quatro categorias de forma a estabelecer o desconto adequado na hora do pagamento no caixa. Os clientes são classificados em: Cliente Diamante (desconto de 15%), Cliente Ouro (desconto de 10%), Cliente Prata (desconto de 5%) e Cliente Bronze (desconto de 0%).

As informações disponíveis na hora do pagamento para realizar a classificação são:

O cliente possui cartão fidelidade da livraria?

O valor da compra é maior do que R\$ 50,00 (Cinquenta Reais)?

Pagamento em dinheiro?

A tabela 2.9 apresenta um conjunto de oito clientes classificados por categoria em função das informações disponíveis.

a) Qual a arquitetura mínima para o *Perceptron*.

a) Treinar a rede *Perceptron* adotando o valor 1 para a taxa de aprendizagem e inicializar todos os pesos com valor 1.

Cliente	Cartão Fidelidade	Compra > R\$50,00	Pagamento em dinheiro?	Categoria
1	sim	sim	sim	Diamante
2	sim	sim	não	Diamante
3	sim	não	sim	Ouro
4	sim	não	não	Ouro
5	não	sim	sim	Prata
6	não	sim	não	Prata
7	não	não	sim	Bronze
8	não	não	não	Bronze

Tabela 2.9: Exemplos de clientes da livraria

# Exemplo Proposto

Os clientes serão classificados em quatro categorias. Logo, precisamos de no **mínimo** dois neurônios na saída para termos quatro possibilidades. A codificação para os dois neurônios de saída fica conforme tabela 2.10.

Categoria	Saídas	
	$d_1$	$d_2$
Diamante	0	0
Ouro	0	1
Prata	1	0
Bronze	1	1

Tabela 2.10: Codificação dos neurônios de saída

Pode-se visualizar a arquitetura desta rede na figura 2.14.

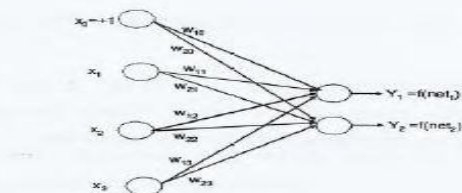


Figura 2.14: Arquitetura para o problema da livraria

A arquitetura da rede fica com três neurônios de entrada além da entrada constante  $x_0 = 1$  totalizando quatro neurônios de entrada e dois neurônios de saídas.

A tabela 2.11 mostra a codificação utilizada para as entradas deste exemplo: (sim = 1 e não = -1).

Cliente	Entradas			Saídas	
	$x_1$ Cartão Fidelidade	$x_2$ Compra > R\$50,00	$x_3$ Pagamento em dinheiro	$d_1$ Categoria	$d_2$ Categoria
1	1	1	1	0	0
2	1	1	-1	0	0
3	1	-1	1	0	1
4	1	-1	-1	0	1
5	-1	1	1	1	0
6	-1	1	-1	1	0
7	-1	-1	1	1	1
8	-1	-1	-1	1	1

Tabela 2.11: Codificação dos exemplos para a livraria

b) A taxa de aprendizagem é 1 (um) e todos os pesos serão inicializados com valor 1 (um). Neste exemplo apresenta-se a cada ciclo os exemplos de forma aleatória.

Para este primeiro ciclo os exemplos serão apresentados na seguinte ordem: 4 1 3 5 8 7 2 6