



## SISTEMAS INTELIGENTES

### Prática 3 – Implementação do Perceptron

Ivan Nunes da Silva

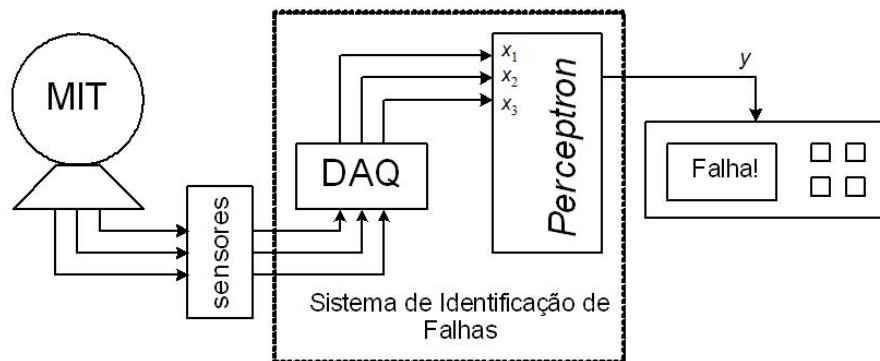


### Problema de Aplicação Prática

- Uma equipe de engenheiros determinou que um tipo de falha, comumente encontrado em motores de indução trifásicos de uma indústria, pode ser pré-identificada (antes que a mesma ocorra) mediante análises de três grandezas físicas  $\{x_1, x_2, x_3\}$ .
- Desta forma, a partir de tais grandezas, a equipe pretende aplicar um **Perceptron** para classificar a operação do motor em duas classes, ou seja, “Operação Normal (Classe  $C_1$ )” ou “Iminência de Falha (Classe  $C_2$ )”, tendo o intuito de se efetuar manutenção preventiva e minimizar o custo operacional da indústria.
- O esquema ilustrativo do processo é mostrado a seguir.



## Esquema Ilustrativo do Processo

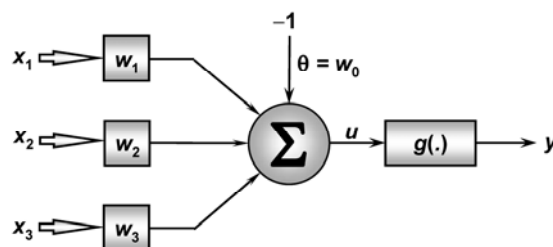


3



## Configuração do Perceptron

- Como existe três grandezas físicas que estão sendo medidas, o neurônio constituinte do **Perceptron** terá então três entradas  $\{x_1, x_2, x_3\}$ .
- Conseqüentemente, a saída  $\{y\}$  do **Perceptron** estará então classificando, baseado em suas três entradas, o status da operação do motor em duas classes, ou sejam:
  - ♦ Classe  $C_1 \rightarrow$  "Operação Normal"  $\rightarrow \{y = 1\}$
  - ♦ Classe  $C_2 \rightarrow$  "Iminência de Falha"  $\rightarrow \{y = -1\}$
- A figura seguinte ilustra o **Perceptron** a ser implementado.



4



## Configuração da Base de Treinamento

- A base de dados de treinamento do **Perceptron**, disponibilizada no arquivo {treinamento.txt}, foi levantada por meio de sucessivos ensaios experimentais e contém o seguinte formato.

	$x_1$	$x_2$	$x_3$	$d$
01	1.4715	0.8996	1.1509	-1
02	4.6315	0.3409	2.4209	-1
03	2.9055	3.1858	3.1770	1
04	3.2563	4.3231	3.8009	1
05	0.2798	4.0843	2.2521	1
06	2.6446	3.4718	3.2715	1
07	1.0620	2.7164	1.7854	-1
08	3.5126	4.7822	4.1947	1
09	2.2227	0.427	1.2697	-1
10	0.2867	3.1473	1.8032	1
...	(...)	(...)	(...)	(...)

5

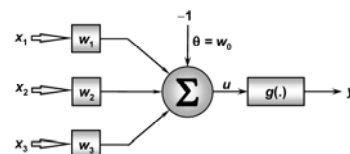


## Implementação do Treinamento (I)

- a) Carregar a matriz de treinamento  $M$  usando a seguinte instrução:

$M = \text{load}(\text{'nome\_do\_arquivo'}); \{ \text{Mostre } M \text{ para conferência} \}$

$$M = \begin{bmatrix} \overset{x_1}{1.4715} & \overset{x_2}{0.8996} & \overset{x_3}{1.1509} & \overset{d}{-1} \\ 4.6315 & 0.3409 & 2.4209 & -1 \\ 2.9055 & 3.1858 & 3.1770 & 1 \\ (...) & (...) & (...) & (...) \end{bmatrix}$$



- b) Implementar as seguintes instruções a partir da matriz  $M$ :

- ❖ Definir a matriz  $T$ , referentes aos sinais de entrada do **Perceptron**, que seja composta pelas três primeiras colunas de  $M$ , inserindo ainda o elemento -1 (relativo ao termo  $\theta$ ) em sua primeira coluna.
- ❖ Definir o vetor  $d$ , referente aos sinais de saída do **Perceptron**, que seja composto pela última coluna de  $M$ . {Mostre  $T$  e  $d$  para conferência}

$$T = \begin{bmatrix} -1 & 1.4715 & 0.8996 & 1.1509 \\ -1 & 4.6315 & 0.3409 & 2.4209 \\ -1 & 2.9055 & 3.1858 & 3.1770 \\ (...) & (...) & (...) & (...) \end{bmatrix} \quad d = \begin{bmatrix} -1 \\ -1 \\ 1 \\ (...) \end{bmatrix}$$

6

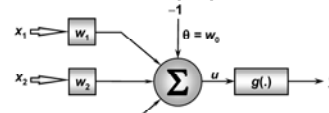


## Implementação do Treinamento (II)

c) Inicializar as seguintes variáveis:

- Taxa de aprendizagem em 0.01;  $\{\eta \leftarrow 0.01\}$
- Contador de épocas em 0;  $\{\text{época} \leftarrow 0\}$
- Vetor de pesos  $\{\mathbf{w}\}$  com valores aleatórios uniformemente distribuídos entre 0 e 1, sendo que cada um de seus elementos estará representando os seguintes parâmetros:

$$\mathbf{w} = [\theta \ w_1 \ w_2 \ w_3]^T; \quad \{\text{Mostre } \mathbf{w} \text{ para conferência}\}$$



d) Implementar a instrução que, dada uma linha  $k$  da matriz  $\mathbf{T}$ , obtenha o potencial de ativação do neurônio, ou seja:

$\mathbf{x} \leftarrow \mathbf{T}(k, :)^T$  {onde  $\mathbf{x}$  conterá a  $k$ -ésima linha da matriz  $\mathbf{T}$ }

$u \leftarrow \mathbf{w}^T * \mathbf{x}$  {realize eventuais transposições que sejam necessárias}

Teste a sua instrução para  $k = 2$ , verificando se o valor de retorno está correto.

$\mathbf{T} =$	-1	1.4715	0.8996	1.1509	
	-1	4.6315	0.3409	2.4209	$\leftarrow k = 2$
	-1	2.9055	3.1858	3.1770	
	(...)	(...)	(...)	(...)	

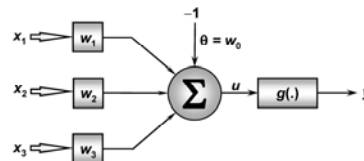
7



## Implementação do Treinamento (III)

e) Implementar o processo de treinamento do **Perceptron** inspirado no algoritmo apresentado na aula teórica, também conhecida como regra de **Hebb**, ou seja:

```
(...)
erro ← "existe"
época ← 0
Enquanto (erro = "existe"), faça:
    erro ← "não existe";
    época ← época + 1;
    Para k variando de 1 até a quantidade total de amostras em  $\mathbf{T}$ , faça:
         $\mathbf{x} \leftarrow \mathbf{T}(k, :)^T$ ; {atribua amostra k de  $\mathbf{T}$  ao vetor  $\mathbf{x}$  // Use instruções do item "d"}
         $u \leftarrow \mathbf{w}^T * \mathbf{x}$ ; {realize as eventuais transposições que sejam necessárias}
         $y \leftarrow \text{sin}(u)$ ;
        Se  $y \neq d(k)$  então
             $\mathbf{w} \leftarrow \mathbf{w} + \eta * (d(k) - y) * \mathbf{x}$ ;
            erro ← "existe";
        Fim_se
    Fim_Para
Fim_Enquanto
```



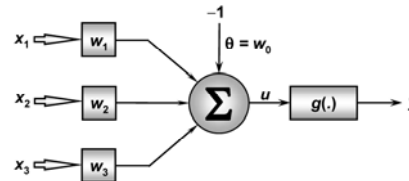
{Execute a rede pelo menos três vezes e analise os números de épocas e os valores finais para o vetor  $\mathbf{w}$  imprimir o número de épocas e o valor final do vetor  $\mathbf{w}$ }

8



## Implementação do Treinamento (IV)

- f) Após o treinamento do **Perceptron**, aplique-o para efetuar a identificação de falhas para algumas situações coletadas pelos sensores situados na planta industrial. Carregue numa matriz **V** o arquivo {teste.txt} que contém a relação completa destas situações representadas por medições de  $x_1$ ,  $x_2$  e  $x_3$ .

$$V = \begin{bmatrix} \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} \begin{matrix} 4.0736 \\ 0.6349 \\ 3.1618 \\ 1.3925 \\ 4.7875 \\ (...) \end{matrix} \begin{matrix} 4.5290 \\ 4.5669 \\ 0.4877 \\ 2.7344 \\ 4.8244 \\ (...) \end{matrix} \begin{matrix} 4.2580 \\ 2.4343 \\ 1.8373 \\ 2.0922 \\ 4.6913 \\ (...) \end{matrix} \end{bmatrix};$$


- g) Prepare esta matriz **V**, adicionando os elementos -1 em sua primeira coluna, a fim de ser inserida nas entradas do **Perceptron** já treinado.

9



## Implementação do Treinamento (V)

- h) Implemente as instruções que permita a classificação, após a realização do processo de treinamento, usando o algoritmo seguinte. Forneça também os resultados da classificação.

(...)

Para  $k$  variando de 1 até a quantidade total de amostras em **V**, faça:

$$\begin{cases} \mathbf{x} \leftarrow \mathbf{V}(k, :)^T; & \{\text{atribua a amostra } k \text{ de } \mathbf{V} \text{ ao vetor } \mathbf{x}\} \\ u \leftarrow \mathbf{w}^T * \mathbf{x}; & \{\text{realize as transposições que sejam necessárias}\} \\ y \leftarrow \text{sinal}(u); \\ \text{Imprima}(y); \end{cases}$$

Fim\_Para

(...)

- ♦ Se  $\{y = 1\} \rightarrow$  "Operação Normal"  $\rightarrow$  Classe  $C_1$
- ♦ Se  $\{y = -1\} \rightarrow$  "Iminência de Falha"  $\rightarrow$  Classe  $C_2$

10