

**INE5633 – SISTEMAS INTELIGENTES – TURMA B - 2011/2**  
**PROF. LUIS OTAVIO ALVARES**

**TRABALHO PRÁTICO 4 (individual)**

Fazer um programa que implemente um neurônio simples para discriminar entre dois padrões de entrada. Cada padrão de entrada corresponde a uma matriz  $7 \times 5$ , totalizando 35 entradas. O nodo deverá ser treinado para responder com 1 para um padrão e -1 para o outro padrão.

Utilizar uma função de ativação do tipo sigmoide : função tangente hiperbólica  $y(v)=\tanh(a.v)$  , onde  $a$  é o parametro de inclinação. Usar  $a = 10$ .

Utilizar a seguinte regra de treinamento:  $W(t+1) = W(t) + \eta(d(t)-y(t)).(1-y(t)^2).X(t)$ , onde

$W$  = vetor de pesos:  $w_0, w_1, w_2, w_3, \dots$

$X$  = vetor de entrada:  $x_0, x_1, x_2, x_3, \dots$

$\eta$  = constante de correção do erro

$d$  = saída desejada (última posição do registro de entrada)

$y$  = saída fornecida

Treinar a rede com o arquivo treino1.txt e avaliá-la com o arquivo avali.txt

Inicializar todos os pesos com **zero**.

No treinamento, deve-se passar todos os registros de entrada. E repetir isso por 10.000 ou 100.000 vezes. (alternando sempre os dados de entrada)

**Responder:**

1- Preencher a tabela:

Dado de teste	Constante de correção do erro	épocas	resultado
1	0,01	1.000	
1	0,01	10.000	
1	0,01	100.000	
1	0,001	1.000	
1	0,001	10.000	
1	0,001	100.000	
1	0,0001	1.000	
1	0,0001	10.000	
1	0,0001	100.000	
1	0,00001	10.000	
1	0,00001	100.000	
1	0,00001	1.000.000	
1	0,00001	10.000.000	
2	0,01	1.000	

2	0,01	10.000	
2	0,01	100.000	
2	0,001	1.000	
2	0,001	10.000	
2	0,001	100.000	
2	0,0001	1.000	
2	0,0001	10.000	
2	0,0001	100.000	
2	0,00001	10.000	
2	0,00001	100.000	
2	0,00001	1.000.000	
2	0,00001	10.000.000	

2- Comentar sobre o número de épocas de treinamento e a constante de correção do erro utilizadas;

3- Considerando os valores 0,0001 para a constante de correção de erro e 1.000.000 para o número de épocas, imprimir o vetor de pesos do neurônio, após o treino.

4- Usando estes valores de pesos, para cada um dos dados de teste, mostrar a saída do neurônio.

5- Comentar sobre a capacidade de generalização da rede.

OBS:  $\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$