Universidade de São Paulo
Escola de Engenharia de São Carlos
Departamento de Engenharia Elétrica

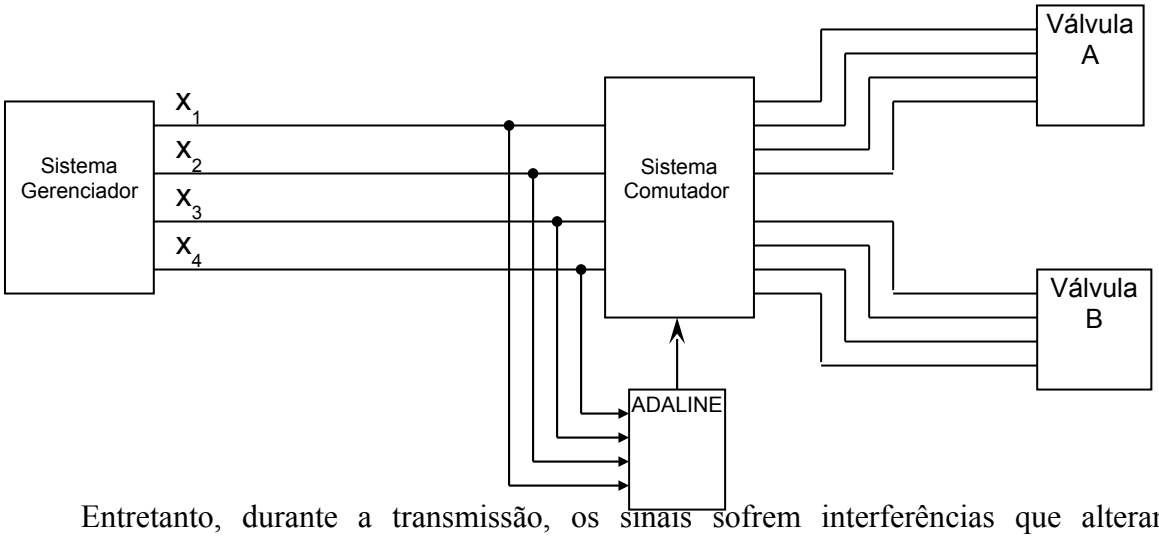
Alunos: Luisa Helena Bartocci Liboni
Rodrigo de Toledo Caropreso

Redes Neurais Artificiais

(Prof. Ivan Nunes da Silva)

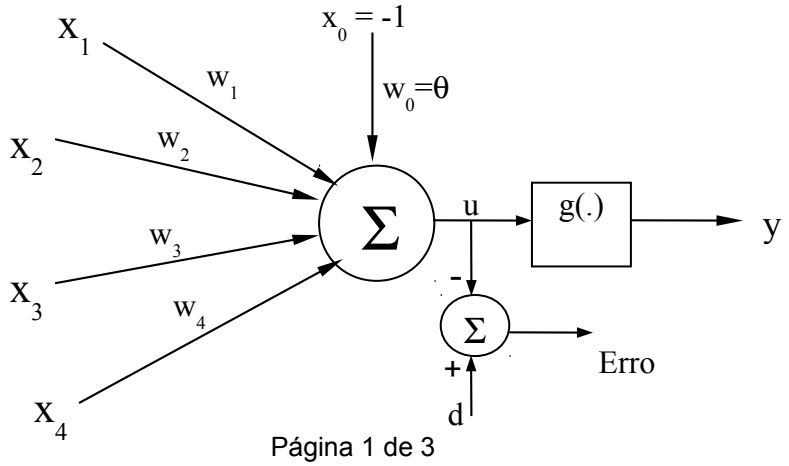
EPC-2

Um sistema de gerenciamento automático de controle de duas válvulas, situado a 500 metros de um processo industrial, envia um sinal codificado constituído de quatro grandezas {x_1, x_2, x_3 e x_4} que são necessárias para o ajuste de cada uma das válvulas. Conforme mostra a figura abaixo, a mesma via de comunicação é utilizada para acionamento de ambas as válvulas, sendo que o comutador localizado próximo das válvulas deve decidir se o sinal é para a válvula A ou B.



Entretanto, durante a transmissão, os sinais sofrem interferências que alteram o conteúdo das informações transmitidas. Para resolver este problema, a equipe de engenheiros e cientistas pretende treinar uma rede ADALINE para classificar os sinais ruidosos, confirmando ao sistema comutador se os dados devem ser encaminhados para o comando de ajuste da válvula A ou B.

Assim, baseado nas medições dos sinais já com ruídos, formou-se o conjunto de treinamento em anexo, tomando por convenção o valor –1 para os sinais que devem ser encaminhados para o ajuste da válvula A e o valor +1 se os mesmos devem ser enviados para a válvula B. Assim, a estrutura do ADALINE é mostrada na figura seguinte.



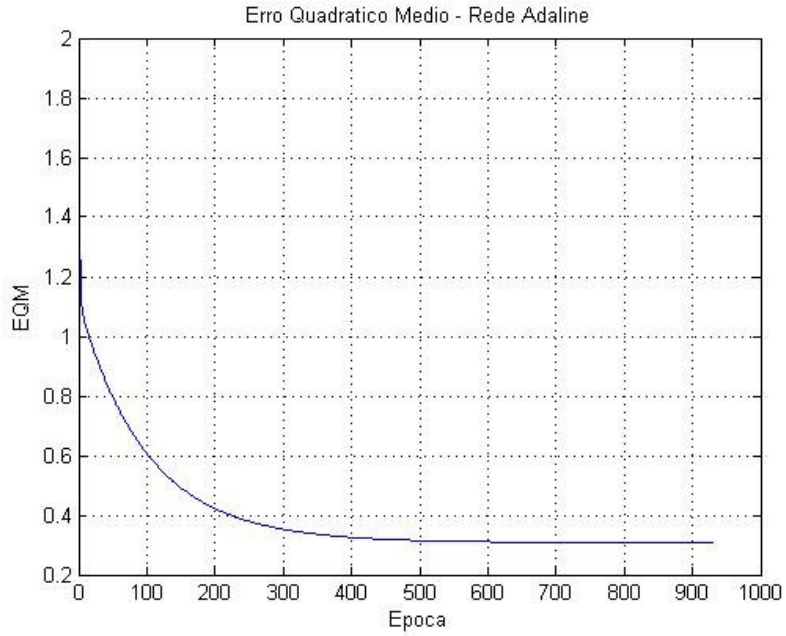
Utilizando o algoritmo de treinamento da Regra Delta para classificação de padrões no ADALINE, realize as seguintes atividades:

1. Execute 5 treinamentos para a rede ADALINE inicializando o vetor de pesos em cada treinamento com valores aleatórios entre zero e um. Se for o caso, reinicie o gerador de números aleatórios em cada treinamento, de tal forma que os elementos do vetor de pesos iniciais não sejam os mesmos. Utilize taxa de aprendizado $\eta = 0.0025$ e precisão $\varepsilon = 10^{-6}$.

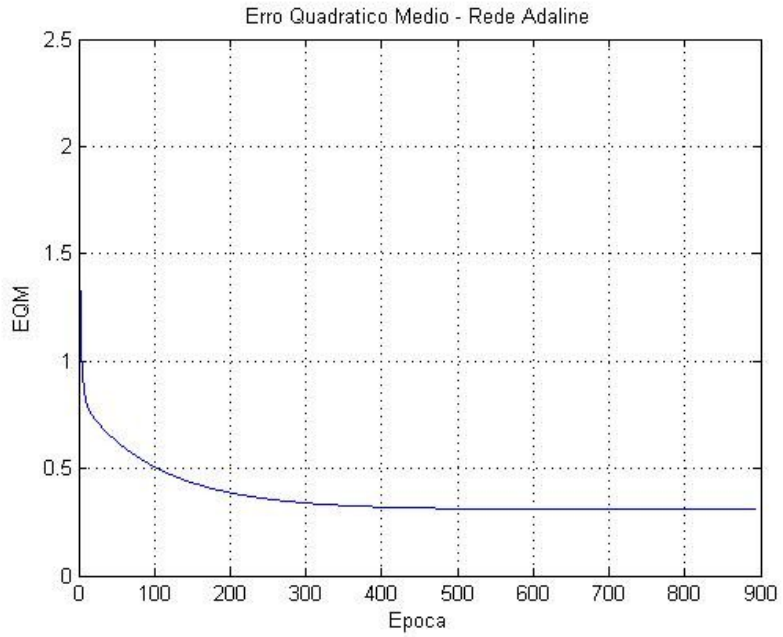
2. Registre os resultados dos 5 treinamentos acima na tabela abaixo:

Trein.	**Vetor de Pesos Inicial**					**Vetor de Pesos Final**					**Epc**
	w_0	w_1	w_2	w_3	w_4	w_0	w_1	w_2	w_3	w_4	
1º (T1)	0.7621	0.4565	0.0185	0.8214	0.4447	-1.8113	1.3126	1.6414	-0.4263	-1.1771	930
2º (T2)	0.6154	0.7919	0.9218	0.7382	0.1763	-1.8113	1.3126	1.6415	-0.4262	-1.1772	892
3º (T3)0.4057	0.4057	0.9355	0.9169	0.4103	0.8936	-1.8113	1.3126	1.6414	-0.4264	-1.1772	882
4º (T4)	0.0579	0.3529	0.8132	0.0099	0.1389	-1.8114	1.3125	1.6413	-0.4267	-1.1771	837
5º (T5)	0.2028	0.1987	0.6038	0.2722	0.1988	-1.8113	1.3125	1.6413	-0.4265	-1.1771	866

3. Para os dois primeiros treinamentos realizados acima, trace então os respectivos gráficos dos valores de erro quadrático médio (EQM) em função de cada época de treinamento. Imprima os dois gráficos numa mesma folha.



1º Treinamento



2º Treinamento

4. Para todos os treinamentos realizados, aplique então a rede ADALINE para classificar e indicar ao comutador se os sinais seguintes devem ser encaminhados para a válvula A ou B.

Amostra	x_1	x_2	x_3	x_4	y (T1)	y (T2)	y (T3)	y (T4)	y (T5)
1	0.9694	0.6909	0.4334	3.4965	A	A	A	A	A
2	0.5427	1.3832	0.6390	4.0352	A	A	A	A	A
3	0.6081	-0.9196	0.5925	0.1016	B	B	B	B	B
4	-0.1618	0.4694	0.2030	3.0117	A	A	A	A	A
5	0.1870	-0.2578	0.6124	1.7749	A	A	A	A	A
6	0.4891	-0.5276	0.4378	0.6439	B	B	B	B	B
7	0.3777	2.0149	0.7423	3.3932	B	B	B	B	B
8	1.1498	-0.4067	0.2469	1.5866	B	B	B	B	B
9	0.9325	1.0950	1.0359	3.3591	B	B	B	B	B
10	0.5060	1.3317	0.9222	3.7174	A	A	A	A	A
11	0.0497	-2.0656	0.6124	-0.6585	A	A	A	A	A
12	0.4004	3.5369	0.9766	5.3532	B	B	B	B	B
13	-0.1874	1.3343	0.5374	3.2189	A	A	A	A	A
14	0.5060	1.3317	0.9222	3.7174	A	A	A	A	A
15	1.6375	-0.7911	0.7537	0.5515	B	B	B	B	B

5. Embora o número de épocas de cada treinamento realizado no item 2 seja diferente, explique por que então os valores dos pesos continuam praticamente inalterados.

Os pesos praticamente não se alteram porque a Regra Delta leva a rede ADALINE para uma região de separação das classes onde o erro quadrático médio é mínimo, diferentemente do Perceptron, que terminava o treinamento assim que fosse possível separar os conjuntos de treino.

Como a região de separação com EQM mínimo é a mesma para qualquer treinamento, os pesos determinados pelo algoritmo são praticamente os mesmos.

OBSERVAÇÕES:

1. O EPC pode ser realizado em grupo de três pessoas. Se for o caso, entregar somente um EPC com o nome de todos integrantes.
2. As folhas contendo os resultados do EPC devem ser entregue em seqüência e grampeadas (não use clips).
3. Em se tratando de EPC que contenha implementação computacional, anexe (de forma impressa) o programa fonte referente ao mesmo.

ANEXO – Conjunto de Treinamento.

Amostra	x_1	x_2	x_3	x_4	d
01	0.4329	-1.3719	0.7022	-0.8535	1.0000
02	0.3024	0.2286	0.8630	2.7909	-1.0000
03	0.1349	-0.6445	1.0530	0.5687	-1.0000
04	0.3374	-1.7163	0.3670	-0.6283	-1.0000
05	1.1434	-0.0485	0.6637	1.2606	1.0000
06	1.3749	-0.5071	0.4464	1.3009	1.0000
07	0.7221	-0.7587	0.7681	-0.5592	1.0000
08	0.4403	-0.8072	0.5154	-0.3129	1.0000
09	-0.5231	0.3548	0.2538	1.5776	-1.0000
10	0.3255	-2.0000	0.7112	-1.1209	1.0000
11	0.5824	1.3915	-0.2291	4.1735	-1.0000
12	0.1340	0.6081	0.4450	3.2230	-1.0000
13	0.1480	-0.2988	0.4778	0.8649	1.0000
14	0.7359	0.1869	-0.0872	2.3584	1.0000
15	0.7115	-1.1469	0.3394	0.9573	-1.0000
16	0.8251	-1.2840	0.8452	1.2382	-1.0000
17	0.1569	0.3712	0.8825	1.7633	1.0000
18	0.0033	0.6835	0.5389	2.8249	-1.0000
19	0.4243	0.8313	0.2634	3.5855	-1.0000
20	1.0490	0.1326	0.9138	1.9792	1.0000
21	1.4276	0.5331	-0.0145	3.7286	1.0000
22	0.5971	1.4865	0.2904	4.6069	-1.0000
23	0.8475	2.1479	0.3179	5.8235	-1.0000
24	1.3967	-0.4171	0.6443	1.3927	1.0000
25	0.0044	1.5378	0.6099	4.7755	-1.0000
26	0.2201	-0.5668	0.0515	0.7829	1.0000
27	0.6300	-1.2480	0.8591	0.8093	-1.0000
28	-0.2479	0.8960	0.0547	1.7381	1.0000
29	-0.3088	-0.0929	0.8659	1.5483	-1.0000
30	-0.5180	1.4974	0.5453	2.3993	1.0000
31	0.6833	0.8266	0.0829	2.8864	1.0000
32	0.4353	-1.4066	0.4207	-0.4879	1.0000
33	-0.1069	-3.2329	0.1856	-2.4572	-1.0000
34	0.4662	0.6261	0.7304	3.4370	-1.0000
35	0.8298	-1.4089	0.3119	1.3235	-1.0000

CÓDIGO FONTE UTILIZADO

Carregamento dos Dados

```
[N_Amostra DB_X1 DB_X2 DB_X3 DB_X4 DB_D] = textread( 'Dados_Treino.dat', '%d %f %f %f %f %f', 'headerlines', 1);

[N_Amostra DB_X1 DB_X2 DB_X3 DB_X4] = textread( 'Dados_Operacao.dat', '%d %f %f %f %f', 'headerlines', 1);
```

Treinamento da Rede ADALINE

```
function [pesos, eqm] = Adaline_Treino( eta, epson, entradas, saidas, max_epocas )
%Adaline_Treino Treinamento de Adaline 1 camada
%   eta          -> coeficiente de treino
%   epson        -> margem de erro
%    entradas     -> matriz com entradas
%   saidas       -> vetor com saidas desejadas
%   max_epocas   -> limite de epocas de treinamento

sizeW = size(entradas);
N_entradas = sizeW(1);
N_amostras = sizeW(2);

pesos = rand(N_entradas, 1)*0.1;

disp('Inicialização da Rede Adaline - Pesos (Pressione uma tecla para continuar)');
pesos
pause

%inicio do treinamento
epoca = 1;
eqm(epoca) = 1 + epson;
stop = 0;

while( epoca <= max_epocas && ~stop )
    erro_parcial = 0;

    for k=1:N_amostras
        u = pesos' * entradas( :, k );

        erro_parcial = erro_parcial + ( saidas(k) - u )^2;
        pesos = pesos + eta * ( saidas(k) - u ) * entradas( :, k );
    end;

    %Calcula erro Quadratico Medio
    eqm( epoca ) = erro_parcial / N_amostras;

    if( epoca > 1 ) %precisamos de 2 epocas para poder comparar

%                   abs(eqm(epoca) - eqm(epoca-1))

            if( abs(eqm(epoca) - eqm(epoca-1)) < epson )
                stop = 1;
            end;
    end;
    epoca = epoca + 1;
end;

epoca = epoca - 1;

if( stop == 1 )
    disp( sprintf( 'Rede treinada. Numero de epocas: %d', epoca) );
else
    disp( sprintf( 'Limite de epocas atingido (%d), rede nao treinada.', epoca) );
end;
```

Operação da Rede ADALINE

```
function y = Adaline_Executa( pesos, entrada )
%Perceptron_Executa Operacao de Perceptron 1 camada
%    entradas     -> vetor com uma entrada
%   pesos        -> matriz de pesos do treinamento
%   max_epocas   -> limite de epocas de treinamento

u = pesos' * entrada;
y = sign( u );

if( y == -1 )
    disp( sprintf( 'Amostra pertence a Valvula A (-1)') );
else
    disp( sprintf( 'Amostra pertence a Valvula B (+1)') );
end;
```

Execução do EPC2

```
clear;
clc;

%Carrega os dados
Carrega_Tabela_Treino;

%Monta vetores de amostras
N_entradas = 4; %entradas da adaline
eta = 0.0025; %coeficiente de treinamento
epson = 1e-06; % margem do erro

%Normaliza dados (pre-processamento)
% DB_X1_Norm = Normaliza( 1, -1, DB_X1 );
% DB_X2_Norm = Normaliza( 1, -1, DB_X2 );
% DB_X3_Norm = Normaliza( 1, -1, DB_X3 );
% DB_X4_Norm = Normaliza( 1, -1, DB_X4 );

DB_X1_Norm = DB_X1;
DB_X2_Norm = DB_X2;
DB_X3_Norm = DB_X3;
DB_X4_Norm = DB_X4;


x = [];
%monta matriz de entradas
for k=1: length(DB_X1_Norm)
     x(:, k) = [ -1 DB_X1_Norm(k) DB_X2_Norm(k) DB_X3_Norm(k) DB_X4_Norm(k)]';
end;

d = [DB_D];
max_epocas = 20000;

[pesos, erro] = Adaline_Treino(eta, epson, x, d, max_epocas);

disp('Pesos da Rede Adaline treinada');
pesos
% pause
% erro
pause

plot( 1: length(erro), erro );
grid;
title( 'Erro Quadratico Medio - Rede Adaline');
xlabel( 'Epoca' );
ylabel( 'EQM' );

%TESTE - Roda o conjunto de treinamento para avaliar a precisao da rede
% x = [];
% %monta matriz de entradas
% for k=1: length(DB_X1_Norm)
%     y(k) = Adaline_Executa(pesos, [-1 DB_X1_Norm(k) DB_X2_Norm(k) DB_X3_Norm(k) DB_X4_Norm(k)]' );
%     if(y(k) == d(k))
%             disp( sprintf('ACERTO \n') );
%     else
%             disp( sprintf( 'ERRO \n' ));
%     end;
% end;
%
% acerto = sum((y' == d)) / length(d);
% disp( sprintf('Acerto: %3.4f %%', acerto*100));
% pause
%FIM - TESTE


%OPERACAO
%Carrega os dados
Carrega_Tabela_Operacao;

%Monta vetores de amostras
%Normaliza dados (pre-processamento)
% DB_X1_Norm = Normaliza( 1, -1, DB_X1 );
% DB_X2_Norm = Normaliza( 1, -1, DB_X2 );
% DB_X3_Norm = Normaliza( 1, -1, DB_X3 );
% DB_X4_Norm = Normaliza( 1, -1, DB_X4 );

DB_X1_Norm = DB_X1;
DB_X2_Norm = DB_X2;
DB_X3_Norm = DB_X3;
DB_X4_Norm = DB_X4;

x = [];
%monta matriz de entradas
for k=1: length(DB_X1_Norm)
    y = Adaline_Executa(pesos, [-1 DB_X1_Norm(k) DB_X2_Norm(k) DB_X3_Norm(k) DB_X4_Norm(k)]' );
end;
```