**Redes Neurais Artificiais**

**Projeto 5**

O preço de uma determinada mercadoria disposta para ser comercializada no mercado financeiro de ações possui um histórico de variação de valor conforme mostrado na tabela apresentada no Anexo.

Um pool de pesquisadores tentará aplicar redes neurais para tentar prever o comportamento futuro deste processo. Assim, pretende-se utilizar uma arquitetura Perceptron multicamadas, com topologia “*Time Delay*” (TDNN), conforme mostrada na figura abaixo:



As topologias candidatas para serem aplicadas no mapeamento do problema acima são especificadas como se segue:

**Rede 1** 🡪 05 entradas (*p* = 05) com N1 = 10

**Rede 2** 🡪 10 entradas (*p* = 10) com N1 = 15

**Rede 3** 🡪 15 entradas (*p* = 15) com N1 = 25

Utilizando o algoritmo de aprendizagem *backpropagation sem momentum* e os dados de treinamento apresentados no Anexo, realize as seguintes atividades:

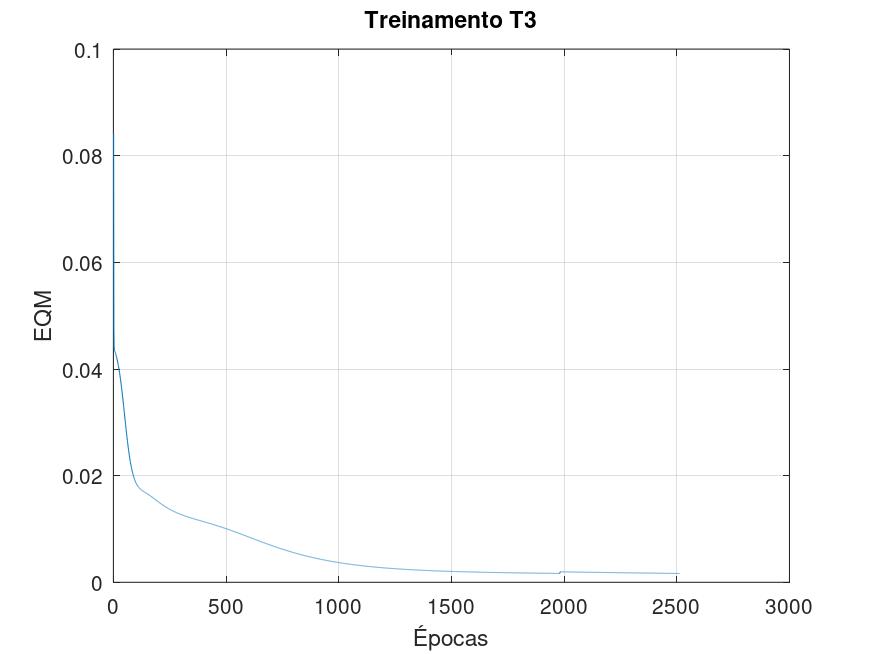
1. Execute 3 treinamentos para cada rede perceptron acima, inicializando-se as matrizes de pesos em cada treinamento com valores aleatórios entre 0 e 1. Se for o caso, reinicie o gerador de números aleatórios em cada treinamento de tal forma que os elementos das matrizes de pesos iniciais não sejam os mesmos. Utilize a função de ativação logística (*sigmoid*) para todos os neurônios, taxa de aprendizado η = 0.1 e precisão ε = 0.5x10-6.
2. Registre os resultados desses 3 treinamentos, considerando-se cada uma dessas três topologias de rede, na tabela a seguir:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Treinamento | **Rede 1** | | **Rede 2** | | **Rede 3** | |
| EQM | Épocas | EQM | Épocas | EQM | Épocas |
| 1o (T1) | **2.6395e-03** | **2320** | **5.2310e-03** | **2125** | **0.0038** | **5017** |
| 2o (T2) | **1.6552e-03** | **2513** | **4.4144e-03** | **2426** | **0.0028** | **4865** |
| 3o (T3) | **1.6592e-03** | **1982** | **5.0585e-03** | **1968** | **0.0034** | **4880** |

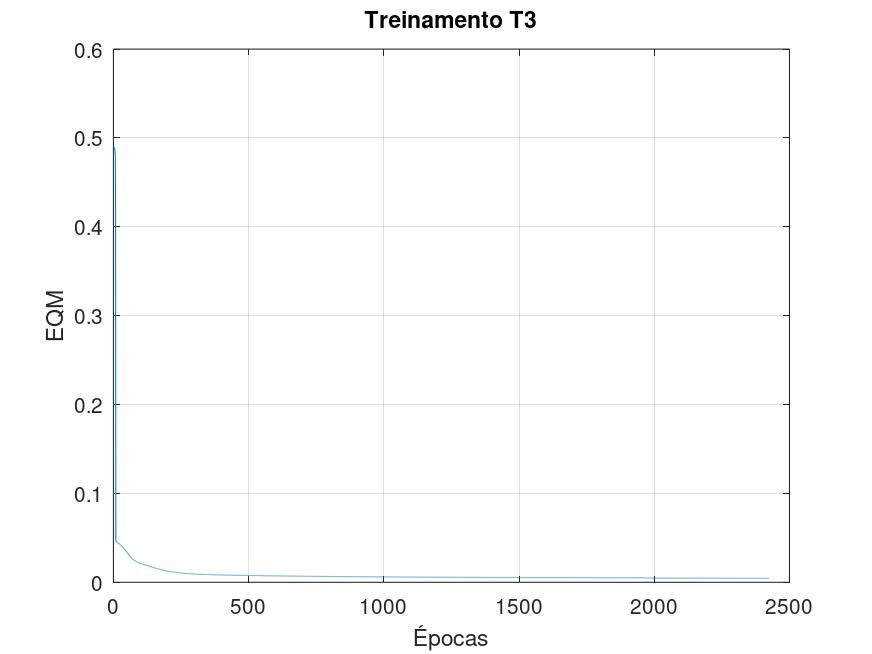
1. Para todos os treinamentos efetuados no item 2, faça então a validação da rede em relação aos valores desejados apresentados na tabela abaixo. Forneça para cada treinamento o erro relativo médio entre os valores desejados e os valores fornecidos pela rede em relação a todos os padrões de teste. Obtenha também a respectiva variância.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Rede 1** | | | **Rede 2** | | | **Rede 3** | | |
| Amostra | *f*(*t*) | (T1) | (T2) | (T3) | (T1) | (T2) | (T3) | (T1) | (T2) | (T3) |
| *t* = 101 | 0.4173 | **0,4531** | **0.459791** | **0.467237** | **0.443323** | **0.437598** | **0.440071** | **0.4345** | **0.4199** | **0.4130** |
| *t* = 102 | 0.0062 | **0,0247** | **0.029720** | **0.028995** | **0.012887** | **0.014525** | **0.011594** | **0.016** | **0.0067** | **0.0106** |
| *t* = 103 | 0.3387 | **0,3912** | **0.387445** | **0.384787** | **0.368168** | **0.368046** | **0.366513** | **0.3868** | **0.3634** | **0.3755** |
| *t* = 104 | 0.1886 | **0,1936** | **0.216309** | **0.216084** | **0.129688** | **0.133698** | **0.120054** | **0.1487** | **0.1524** | **0.1854** |
| *t* = 105 | 0.7418 | **0,7012** | **0.706665** | **0.705290** | **0.737744** | **0.733116** | **0.736879** | **0.7858** | **0.7324** | **0.7443** |
| *t* = 106 | 0.3138 | **0,2031** | **0.186609** | **0.184098** | **0.226085** | **0.226493** | **0.231688** | **0.2415** | **0.2848** | **0.2586** |
| *t* = 107 | 0.4466 | **0,4306** | **0.414881** | **0.411250** | **0.466230** | **0.457838** | **0.468726** | **0.4589** | **0.4300** | **0.4899** |
| *t* = 108 | 0.0835 | **0.0894** | **0.102070** | **0.096845** | **0.117916** | **0.107401** | **0.117813** | **0.0944** | **0.0870** | **0.1084** |
| *t* = 109 | 0.1930 | **0.2247** | **0.224152** | **0.224933** | **0.214499** | **0.219226** | **0.217790** | **0.1856** | **0.1933** | **0.2048** |
| *t* = 110 | 0.3807 | **0.281384** | **0.282201** | **0.274910** | **0.449103** | **0.449943** | **0.455491** | **0.4285** | **0.4362** | **0.4598** |
| *t* = 111 | 0.5438 | **0.534456** | **0.548827** | **0.547488** | **0.545018** | **0.546844** | **0.541098** | **0.5501** | **0.5456** | **0.5744** |
| *t* = 112 | 0.5897 | **0.635471** | **0.650255** | **0.648449** | **0.611296** | **0.617342** | **0.605360** | **0.6023** | **0.5890** | **0.6132** |
| *t* = 113 | 0.3536 | **0.349596** | **0.348106** | **0.351307** | **0.291254** | **0.303036** | **0.285275** | **0.3645** | **0.3559** | **0.3586** |
| *t* = 114 | 0.2210 | **0.189155** | **0.203281** | **0.211400** | **0.237934** | **0.239261** | **0.237789** | **0.2203** | **0.2144** | **0.2282** |
| *t* = 115 | 0.0631 | **0.166079** | **0.160668** | **0.167201** | **0.049904** | **0.051917** | **0.050015** | **0.0569** | **0.0685** | **0.0579** |
| *t* = 116 | 0.4499 | **0.411344** | **0.403543** | **0.402294** | **0.404710** | **0.404743** | **0.402266** | **0.4274** | **0.4825** | **0.4233** |
| *t* = 117 | 0.2564 | **0.274491** | **0.265492** | **0.270202** | **0.275026** | **0.267258** | **0.278829** | **0.2315** | **0.2368** | **0.2427** |
| *t* = 118 | 0.7642 | **0.793394** | **0.794934** | **0.793917** | **0.763431** | **0.768510** | **0.765124** | **0.7596** | **0.7689** | **0.7556** |
| *t* = 119 | 0.1411 | **0.143704** | **0.132553** | **0.129269** | **0.143963** | **0.143072** | **0.144580** | **0.1285** | **0.1289** | **0.1343** |
| *t* = 120 | 0.3626 | **0.313484** | **0.28215** | **0.323184** | **0.345093** | **0.346572** | **0.342041** | **0.357** | **0.3667** | **0.3470** |
| Erro Relativo Médio: | | **-21,155** | **-25.815** | **-25.422** | **-5.1358** | **-3.9153** | **-4.4118** | **2,7** | **0.98** | **0.41** |  |
| Variância: | | **5801.3** | **8347.2** | **8082.4** | **858.67** | **557.99** | **716.37** | **257** | **51** | **119** |  |

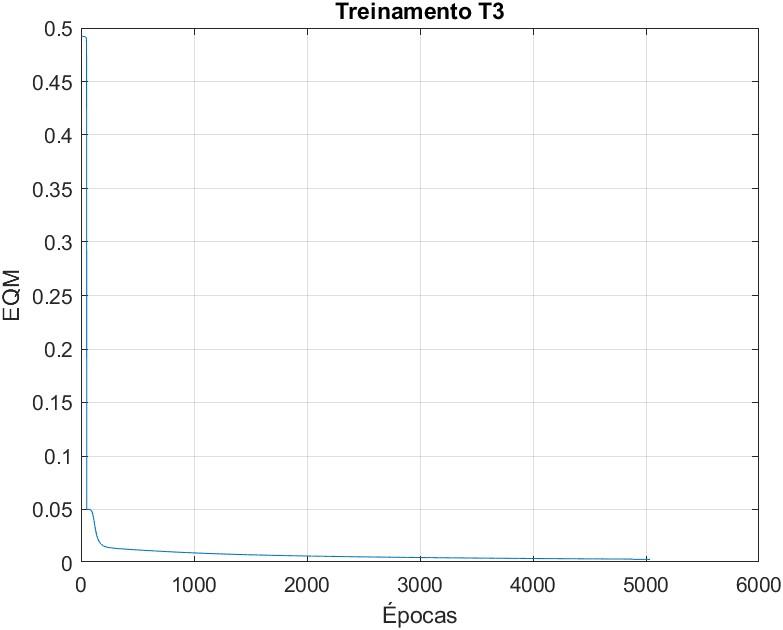
1. Para cada uma das topologias apresentadas na tabela acima, considerando-se ainda o melhor treinamento {T1, T2 ou T3} realizado em cada uma delas, trace então o gráfico dos valores de erro quadrático médio (EQM) em função de cada época de treinamento. Imprima os três gráficos numa mesma folha de modo não superpostos.



Rede 1

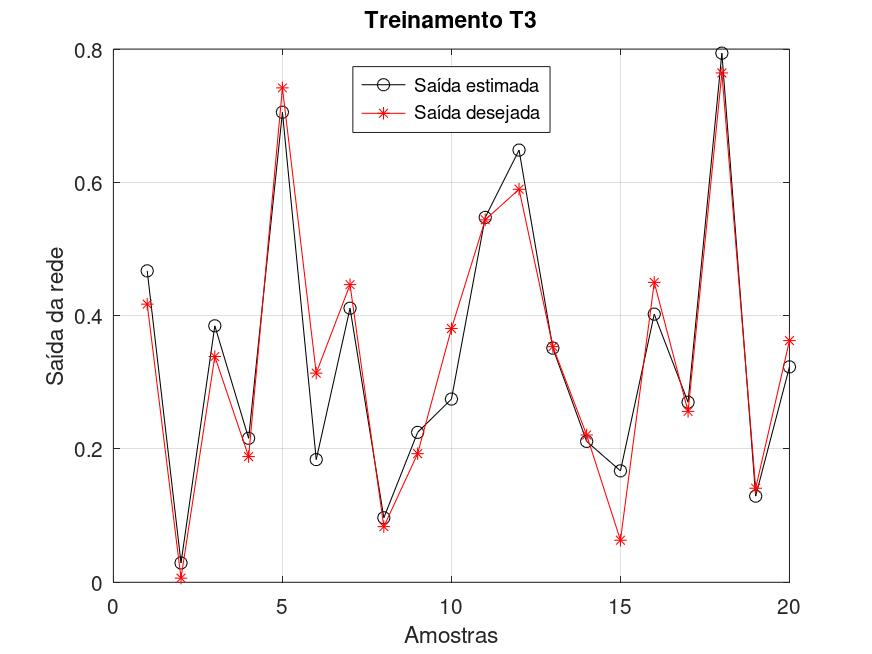


Rede 2

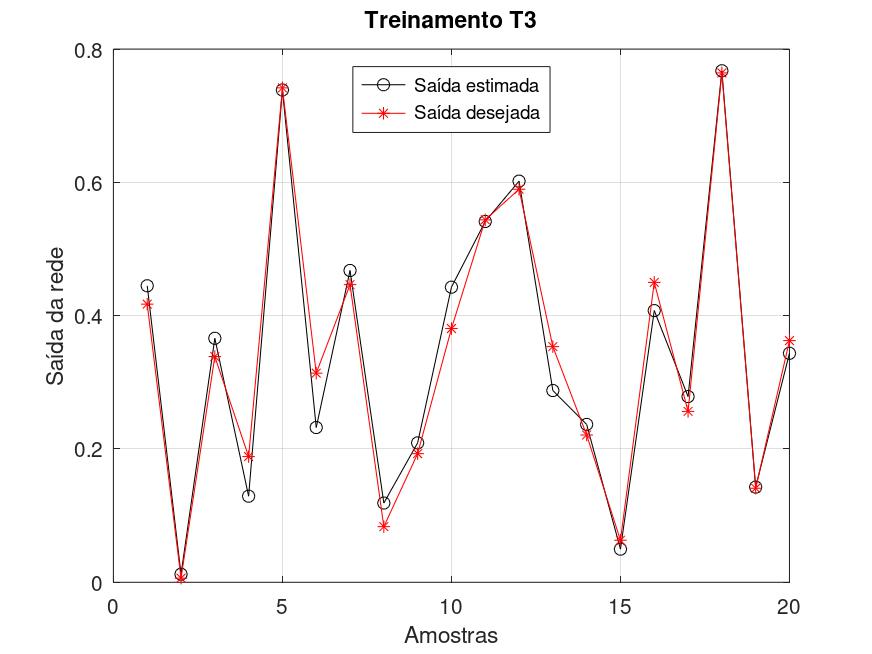


Rede 3

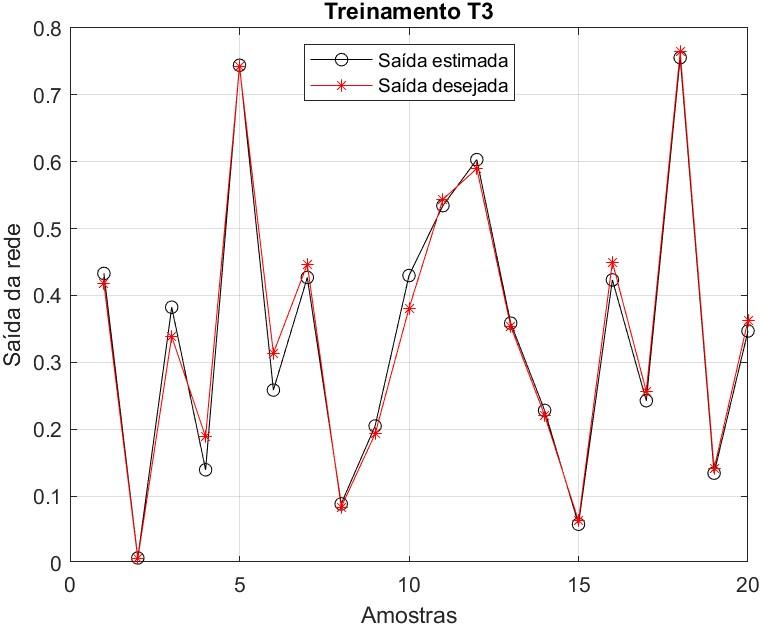
1. Para cada uma das topologias apresentadas na tabela acima, considerando-se ainda o melhor treinamento {T1, T2 ou T3} realizado em cada uma delas, trace então o gráfico dos valores desejados e dos valores estimados pela respectiva rede em função do domínio de operação assumido (*t*=101..120). Imprima os três gráficos numa mesma folha de modo não superpostos.



Rede 1



Rede 2



Rede 3

1. Baseado nas análises dos itens acima, indique então qual das topologias candidatas {Rede 1, Rede 2 ou Rede 3}, e que com qual configuração final de treinamento {T1, T2 ou T3}, seria a mais adequada para realização de previsões neste processo.

**Levando em consideração ao erro quadrático médio, a Rede 3, com o T3, onde apresentaram os menores erros.**

**ANEXO**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Amostra** | ***f*(*t*)** | **Amostra** | ***f*(*t*)** | **Amostra** | ***f*(*t*)** | **Amostra** | ***f*(*t*)** |
| *t* = 1 | 0.1701 | *t* = 26 | 0.2398 | *t* = 51 | 0.3087 | *t* = 76 | 0.3701 |
| *t* = 2 | 0.1023 | *t* = 27 | 0.0508 | *t* = 52 | 0.0159 | *t* = 77 | 0.0006 |
| *t* = 3 | 0.4405 | *t* = 28 | 0.4497 | *t* = 53 | 0.4330 | *t* = 78 | 0.3943 |
| *t* = 4 | 0.3609 | *t* = 29 | 0.2178 | *t* = 54 | 0.0733 | *t* = 79 | 0.0646 |
| *t* = 5 | 0.7192 | *t* = 30 | 0.7762 | *t* = 55 | 0.7995 | *t* = 80 | 0.7878 |
| *t* = 6 | 0.2258 | *t* = 31 | 0.1078 | *t* = 56 | 0.0262 | *t* = 81 | 0.1694 |
| *t* = 7 | 0.3175 | *t* = 32 | 0.3773 | *t* = 57 | 0.4223 | *t* = 82 | 0.4468 |
| *t* = 8 | 0.0127 | *t* = 33 | 0.0001 | *t* = 58 | 0.0085 | *t* = 83 | 0.0372 |
| *t* = 9 | 0.4290 | *t* = 34 | 0.3877 | *t* = 59 | 0.3303 | *t* = 84 | 0.2632 |
| *t* = 10 | 0.0544 | *t* = 35 | 0.0821 | *t* = 60 | 0.2037 | *t* = 85 | 0.3048 |
| *t* = 11 | 0.8000 | *t* = 36 | 0.7836 | *t* = 61 | 0.7332 | *t* = 86 | 0.6516 |
| *t* = 12 | 0.0450 | *t* = 37 | 0.1887 | *t* = 62 | 0.3328 | *t* = 87 | 0.4690 |
| *t* = 13 | 0.4268 | *t* = 38 | 0.4483 | *t* = 63 | 0.4445 | *t* = 88 | 0.4132 |
| *t* = 14 | 0.0112 | *t* = 39 | 0.0424 | *t* = 64 | 0.0909 | *t* = 89 | 0.1523 |
| *t* = 15 | 0.3218 | *t* = 40 | 0.2539 | *t* = 65 | 0.1838 | *t* = 90 | 0.1182 |
| *t* = 16 | 0.2185 | *t* = 41 | 0.3164 | *t* = 66 | 0.3888 | *t* = 91 | 0.4334 |
| *t* = 17 | 0.7240 | *t* = 42 | 0.6386 | *t* = 67 | 0.5277 | *t* = 92 | 0.3978 |
| *t* = 18 | 0.3516 | *t* = 43 | 0.4862 | *t* = 68 | 0.6042 | *t* = 93 | 0.6987 |
| *t* = 19 | 0.4420 | *t* = 44 | 0.4068 | *t* = 69 | 0.3435 | *t* = 94 | 0.2538 |
| *t* = 20 | 0.0984 | *t* = 45 | 0.1611 | *t* = 70 | 0.2304 | *t* = 95 | 0.2998 |
| *t* = 21 | 0.1747 | *t* = 46 | 0.1101 | *t* = 71 | 0.0568 | *t* = 96 | 0.0195 |
| *t* = 22 | 0.3964 | *t* = 47 | 0.4372 | *t* = 72 | 0.4500 | *t* = 97 | 0.4366 |
| *t* = 23 | 0.5114 | *t* = 48 | 0.3795 | *t* = 73 | 0.2371 | *t* = 98 | 0.0924 |
| *t* = 24 | 0.6183 | *t* = 49 | 0.7092 | *t* = 74 | 0.7705 | *t* = 99 | 0.7984 |
| *t* = 25 | 0.3330 | *t* = 50 | 0.2400 | *t* = 75 | 0.1246 | *t* = 100 | 0.0077 |

Código em OCTAVE

%Paulo Régis P. Lima%

clear,clc

ordem = input('Digite a ordem do preditor: ');

N = input('Digite a quantidade de neurônios na camada escondida: ');

for T = 1:3 % Serão realizados 3 treinamentos

p = ordem;

% Obter o conjunto de amostras de treinamento

dados = load('dados\_de\_treinamento.dat');

qtde\_dados\_treinamento = size(dados,1);

% Inicializar as matrizes pesos W1 (Neuronios Intermediário x Entradas) e

% W2(Neurônios saida x Neuronios Intermediários+1) aleatoriamente com

% valores aleatórios pequenos

neuronios\_intermediarios = N; %N

neuronios\_saida = 1;

entradas = p+1;

w1 = random('Uniform',0,1,neuronios\_intermediarios,entradas);

w2 = random('Uniform',0,1,neuronios\_saida,neuronios\_intermediarios+1);

% Taxa de aprendizagem (ta) e precisão

ta = 0.1;

precisao = 0.5\*10^-6;

% Iniciar o contador de épocas

ep = 1;

% Iniciar o Erro Quadrático Médio atual

EQM = 0;

% Laço principal

while true

for k = 1:qtde\_dados\_treinamento - p

x = fliplr(dados(k:p+k-1)');

x = [-1 x];

d = dados(p+k);

% Fase Foward

% Nx1 Nx(p+1) (p+1)x1 Nx1 Nx1 (N+1)x1

I1 = w1 \* x'; Y1 = 1./(1 + exp(-I1)); Y1 = [-1; Y1];

% 1x1 1x(N+1)(N+1)x1 1x1 1x1

I2 = w2 \* Y1; Y2 = 1./(1 + exp(-I2));

% Fase backward

%Derivada da função sigmóide em I1

% Nx1 Nx1 Nx1

a = exp(-I1)./(1+exp(-I1)).^2;

%Derivada da função sigmóide em I2

% 1x1 1x1 1x1

b = exp(-I2)./(1+exp(-I2)).^2;

% 1x1 1x1 1x1 1x1

delta2 = b .\* (d'-Y2);

% 1x(N+1) 1x(N+1) 1x1 1x1 1x(N+1)

w2 = w2 + (ta \* delta2) \* Y1';

% Nx1 Nx1 [1x1 1xN]'

delta1 = a .\* (delta2'\*w2(:,2:neuronios\_intermediarios+1))';

% Nx(p+1) Nx(p+1) 1x1 Nx1 1x(p+1)

w1 = w1 + ta \* delta1 \* x;

end

% Obter saída da rede ajustada

for k = 1:qtde\_dados\_treinamento - p

% Nx1 Nx(p+1) (p+1)x1 Nx1 Nx1 (N+1)x1

I1 = w1 \* x'; Y1 = 1./(1 + exp(-I1)); Y1 = [-1; Y1];

% 1x1 1x(N+1) (N+1)x1 1x1 1x1

I2 = w2 \* Y1; Y2 = 1./(1 + exp(-I2));

% 1x1 1x1 1x1

EQ(:,k) = 0.5\*((d'-Y2).^2);

end

% Cálculo do EQM

% 1x1 1x1 1x1 1x1

EQM = EQM + sum(EQ)/k;

ep = ep + 1;

% 1x1 1x1

eqm(:,ep) = EQM;

EQM = 0;

% 1x1 1x1 1x1

if (abs(eqm(ep) - eqm(ep-1)) < precisao)

break

end

end

disp(strcat('Treinamento T',num2str(T),' finalizado!'))

%% Validação

% Obter o conjunto de dados de validação

validacao = load('dados\_de\_validacao.dat');

qtde\_dados\_validacao = size(validacao,1);

dados = [dados; validacao];

quantidade\_total\_dados = qtde\_dados\_treinamento+qtde\_dados\_validacao;

figure(1),plot(dados(1:qtde\_dados\_treinamento),'\*-'),hold on

eixo\_x = qtde\_dados\_treinamento+1:length(dados);

eixo\_y = dados(qtde\_dados\_treinamento+1:length(dados));

plot(eixo\_x,eixo\_y,'\*-r'),grid

legend('Treinamento','Validação')

xlabel('Amostras'),ylabel('Saída da rede'),title(strcat('Treinamento T',num2str(T)))

for k = qtde\_dados\_treinamento+1:quantidade\_total\_dados

xv = fliplr(dados(k-p:k-1)');

xv = [-1 xv];

yv = dados(k);

% Fase Foward

% Nx1 Nx(p+1) (p+1)x1 Nx1 Nx1 (N+1)x1

I1 = w1 \* xv'; Y1 = 1./(1 + exp(-I1)); Y1 = [-1; Y1];

% 1x20 1x11 11x20 1x20 1x20

I2 = w2 \* Y1; Y2(k-qtde\_dados\_treinamento) = 1./(1 + exp(-I2));

end

% Gráfico da comparação entre a saída calculada pela RNA e a desejada

figure(2),plot(Y2,'o-k'),hold on,plot(validacao,'\*-r'),grid

legend('Saída estimada','Saída desejada','Location','north')

xlabel('Amostras'),ylabel('Saída da rede'),title(strcat('Treinamento T',num2str(T)))

% Salvar o gráfico

grafico = gca;

saveas(grafico, strcat('T', num2str(T), ' - Saídas.jpg'));

% Gráficos do EQM

figure(3),plot(eqm(:,2:size(eqm,2))),grid

xlabel('Épocas'),ylabel('EQM'),title(strcat('Treinamento T',num2str(T)))

% Salvar o gráfico do Épocas x EQM

grafico = gca;

saveas(figure(3), strcat('T', num2str(T), ' - EQM.jpg'), 'jpeg');

% Dados de saída

EQM\_final = eqm(ep)

Epocas = ep

Saida\_da\_rede = Y2'

% Erro Relativo Médio percentual ((Calculado-Real)/Real)/quantidade

ER = 100\*((validacao - Y2')./validacao); % em percentual

ERM = sum(ER)/qtde\_dados\_validacao % em percentual

% Variância do Erro Relativo (Somatório[(Xi-Xméd)^2])/N

Variancia = var(ER) % em percentual

% Salvar os dados finais

save(strcat('T',num2str(T),' - Dados de saída'),...

'EQM\_final','Epocas','Saida\_da\_rede','ERM','Variancia')

close all

end