

Universidade Federal do Pará – UFPA

Instituto de Tecnologia – ITEC

Faculdade de Engenharia Elétrica e Biomédica

Redes Neurais Artificiais

Relatório BackPropagation

Arthur Gutemberg Hora de Miranda – 202007140043

Arthur Felipe dos Santos Fernandes – 201707140065

Lucas Portilho Nunes – 201707140033

Belém

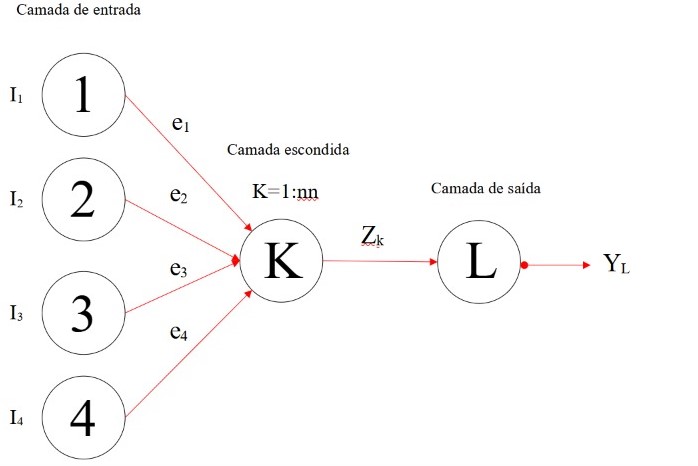
2021

Orientações

O código deve possuir:

1. Gráfico de evolução do erro médio quadrático
2. Gráfico saída desejada versus saída obtida pela rede
3. Quantidade genérica de neurônios na camada escondida
4. Verificar o valor dos pesos sinápticos após o treinamento

Figura 1 – Modelo da rede neural elaborada

Legenda:

In – Entrada dos neurônios ‘n’ na camada de entrada.  
en – Saída dos neurônios ‘n’ da camada de entrada.  
ZK – Saída dos neurônios ‘K’ da camada escondida.  
YL – Saída do neurônio ‘L’ da camada de saída.  
K – Iterador que vai de 1 até nn (número de neurônios na camada oculta).

L – Iterador da saída, análogo ao ‘K’.

Código

clear;

load Iris-Targets.mat

load Iris-Inputs.mat

% Entrada

input = IrisInput';

numlin = length (input(:,1));

numcol = length (input(1,:));

%Saída

output = IrisTargets';

tam1 = length (output(:,1));

tam2 = length (output(1,:));

%Número de neurônios da camada oculta – critério 3 cumprido

nn = 1;

maxn = nn + numcol + tam2;

delta = ones(maxn,1);

%Bias

bias = ones(1,maxn);

%Coeficiente de Aprendizado

na = 0.02;

%Número de iterações (épocas)

epochs = 10^3;

%Pesos

valor = max(numcol+1,nn+1);

pesos = rand(maxn,valor);

%pesos = [ 0 0 0; 0.1 0.1 0.1; 0.1 0.1 0.1; 0.1 0.1 0.1];

%Função Custo

custo = zeros(epochs,1);

for i = 1:epochs

y = zeros (tam1,tam2);

for j = 1:numlin

%Etapa feedfoward - Camada de Entrada

e = zeros(numcol,1);

for k = 1:numcol

e(k) = input(j,k);

end

%Etapa feedfoward - Camada Oculta

z = zeros(nn,1);

for k = 1:nn

temp = 0;

for l = 2:numcol+1

temp = temp + e(l-1)\*pesos(k+numcol,l);

end

temp = temp + bias(1,2+k)\*pesos(k+numcol,1);

z(k) = sigma(temp);

end

%Etapa feedfoward - Camada de Saída

for k = 1:tam2

soma = 0;

for l = 1:nn

soma = soma + z(l)\*pesos(nn+numcol+k,l+1);

end

yy = bias(1,nn+numcol+k)\*pesos(nn+numcol+k,1) + soma;

y(j,k) = sigma(yy);

end

%Etapa backpropagation

%Cálculo do custo por época – critério 1 cumprido

custo(i) = mse(output(j,:),y(j,:));

for k = 1:tam2

delta(nn+k) = y(j,k)\*(1 - y(j,k))\*(output(j,k) - y(j,k));

end

for k = 1:nn

temp = 0;

for l = 1:tam2

temp = temp + pesos(numcol+nn+l,k+1)\*delta(nn+l);

end

delta(k) = z(k)\*(1 - z(k))\*temp/tam2;

end

for k = 1:nn

for l = 1:numcol

pesos(numcol+k,l+1) = pesos(numcol+k,l+1) + na\*e(l)\*delta(k);

end

pesos(numcol+k,1) = pesos(numcol+k,1) + na\*bias(1,numcol+k)\*delta(k);

end

for k = 1:tam2

for l = 1:nn

pesos(numcol+nn+k,l+1) = pesos(numcol+nn+k,l+1) + na\*z(l)\*delta(nn+k);

end

pesos(numcol+nn+k,1) = pesos(numcol+nn+k,1) + na\*bias(1,numcol+nn+k)\*delta(nn+k);

end

end

end

%Plotagem do gráfico saída obtida versus desejada  
%Critério 2 cumprido

classes = vec2ind (y');

saida = vec2ind (IrisTargets);

plot(saida,'or')

hold

plot(classes,'xg')

acerto = sum((saida-classes)==0)\*100/150;

disp (acerto)

immse(output,y)

%Verificação dos pesos obtidos – critério 4 cumprido

disp (pesos)

Resultados do treinamento

**Para 1 neurônio e 104 épocas:**

y =

0.9228 0.3606 0.0000

0.9178 0.3600 0.0000

0.9198 0.3597 0.0000

0.9158 0.3591 0.0000

0.9231 0.3591 0.0000

0.9210 0.3586 0.0000

0.9177 0.3581 0.0000

0.9210 0.3579 0.0000

0.9130 0.3571 0.0000

0.9202 0.3571 0.0000

0.9243 0.3569 0.0000

0.9192 0.3563 0.0000

0.9199 0.3559 0.0000

0.9204 0.3556 0.0000

0.9271 0.3555 0.0000

0.9258 0.3551 0.0000

0.9239 0.3546 0.0000

0.9212 0.3541 0.0000

0.9232 0.3539 0.0000

0.9226 0.3535 0.0000

0.9212 0.3530 0.0000

0.9199 0.3526 0.0000

0.9244 0.3524 0.0000

0.9077 0.3513 0.0000

0.9156 0.3513 0.0000

0.9161 0.3509 0.0000

0.9148 0.3505 0.0000

0.9227 0.3505 0.0000

0.9228 0.3502 0.0000

0.9166 0.3495 0.0000

0.9161 0.3491 0.0000

0.9188 0.3489 0.0000

0.9265 0.3489 0.0000

0.9268 0.3486 0.0000

0.9205 0.3479 0.0000

0.9223 0.3476 0.0000

0.9248 0.3474 0.0000

0.9205 0.3468 0.0000

0.9163 0.3463 0.0000

0.9218 0.3462 0.0000

0.9219 0.3458 0.0000

0.8978 0.3444 0.0000

0.9186 0.3450 0.0000

0.9081 0.3441 0.0000

0.9166 0.3442 0.0000

0.9145 0.3437 0.0000

0.9237 0.3438 0.0000

0.9187 0.3432 0.0000

0.9244 0.3431 0.0000

0.9216 0.3426 0.0000

0.0015 0.3112 0.0182

0.0003 0.3069 0.0662

0.0001 0.3043 0.1528

0.0001 0.3047 0.1673

0.0001 0.3049 0.1944

0.0002 0.3089 0.0913

0.0001 0.3084 0.1287

0.0367 0.3275 0.0010

0.0026 0.3193 0.0114

0.0003 0.3132 0.0692

0.0045 0.3227 0.0069

0.0006 0.3168 0.0408

0.0122 0.3275 0.0028

0.0002 0.3148 0.1037

0.0248 0.3314 0.0015

0.0088 0.3287 0.0038

0.0001 0.3147 0.1873

0.0485 0.3358 0.0008

0.0000 0.3115 0.4815

0.0299 0.3357 0.0013

0.0000 0.3134 0.4680

0.0671 0.3400 0.0006

0.0001 0.3201 0.1692

0.0152 0.3366 0.0023

0.0758 0.3426 0.0005

0.0503 0.3419 0.0008

0.0081 0.3368 0.0041

0.0003 0.3270 0.0781

0.0022 0.3343 0.0129

0.4423 0.3534 0.0001

0.0461 0.3453 0.0008

0.1324 0.3496 0.0003

0.0688 0.3481 0.0006

0.0000 0.3229 0.5328

0.0012 0.3369 0.0227

0.0118 0.3448 0.0029

0.0238 0.3477 0.0015

0.0080 0.3451 0.0041

0.0670 0.3525 0.0006

0.0120 0.3478 0.0029

0.0097 0.3479 0.0035

0.0147 0.3500 0.0024

0.0830 0.3561 0.0005

0.2175 0.3602 0.0002

0.0134 0.3519 0.0026

0.1109 0.3592 0.0004

0.0350 0.3563 0.0011

0.0718 0.3593 0.0005

0.3607 0.3659 0.0001

0.0255 0.3575 0.0014

0.0000 0.3230 0.9879

0.0000 0.3295 0.8964

0.0000 0.3275 0.9321

0.0000 0.3303 0.8304

0.0000 0.3232 0.9742

0.0000 0.3231 0.9700

0.0000 0.3291 0.8116

0.0000 0.3265 0.8914

0.0000 0.3239 0.9391

0.0000 0.3201 0.9773

0.0000 0.3297 0.6610

0.0000 0.3231 0.9240

0.0000 0.3215 0.9451

0.0000 0.3186 0.9718

0.0000 0.3153 0.9878

0.0000 0.3179 0.9686

0.0000 0.3236 0.8288

0.0000 0.3175 0.9627

0.0000 0.3114 0.9926

0.0000 0.3231 0.7845

0.0000 0.3141 0.9780

0.0000 0.3157 0.9594

0.0000 0.3120 0.9837

0.0000 0.3209 0.7952

0.0000 0.3141 0.9601

0.0000 0.3181 0.8679

0.0000 0.3200 0.7704

0.0000 0.3191 0.7890

0.0000 0.3085 0.9855

0.0000 0.3191 0.7406

0.0000 0.3091 0.9768

0.0000 0.3130 0.9249

0.0000 0.3053 0.9896

0.0000 0.3189 0.6394

0.0000 0.3112 0.9310

0.0000 0.3036 0.9902

0.0000 0.3029 0.9907

0.0000 0.3081 0.9555

0.0000 0.3109 0.8948

0.0000 0.3048 0.9756

0.0000 0.3006 0.9913

0.0000 0.3030 0.9805

0.0000 0.3025 0.9805

0.0000 0.2993 0.9909

0.0000 0.2982 0.9921

0.0000 0.2998 0.9863

0.0000 0.3014 0.9754

0.0000 0.3017 0.9701

0.0000 0.2979 0.9875

0.0000 0.3025 0.9528  
pesos =

0.3517 0.3804 0.5688 0.1656 0.2290

0.8308 0.5678 0.4694 0.6020 0.9133

0.5853 0.0759 0.0119 0.2630 0.1524

0.5497 0.0540 0.3371 0.6541 0.8258

-1.6015 -0.4769 -0.8704 0.9610 2.1891

2.5790 -19.6688 0.7943 0.7482 0.9961

-0.5711 -0.3102 0.3112 0.4505 0.0782

-12.0849 17.5552 0.5285 0.0838 0.4427

Figura 2 – Função custo x épocas

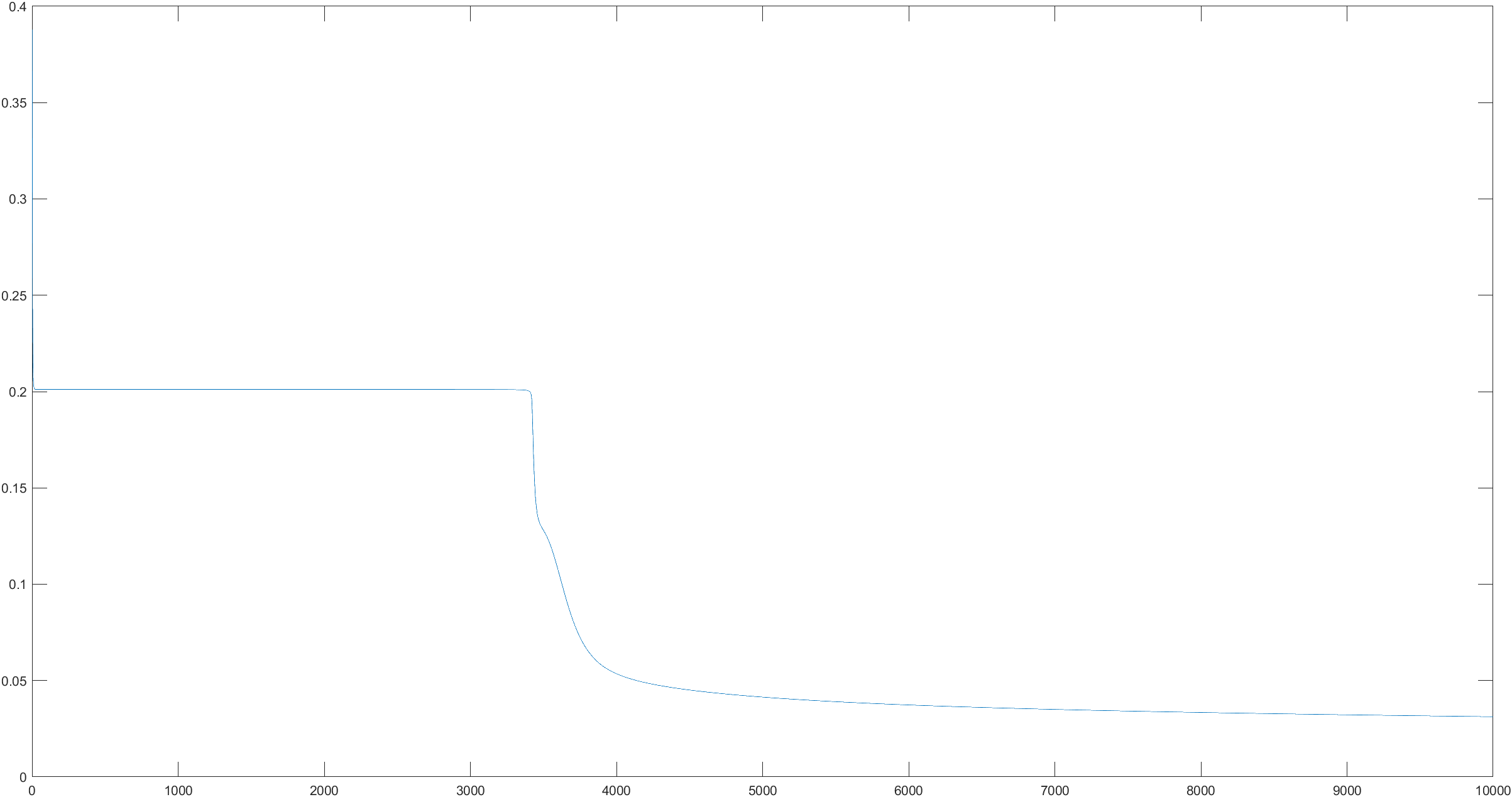


Figura 3 – Saída obtida (verde) x saída desejada (vermelho)

