Universidade Federal do Rio Grande do Norte

Disciplina: DCA0133 -APRENDIZAGEM DE MÁQUINA E MINERAÇÃO DE DADOS - T01

Terceira Lista de Exercícios

Alunos:
Felipe Ferreira Barbosa - 20170008733
Vanessa Dantas de Souto Costa - 20170010188
Mariana Beatriz Fonseca Alvesz - 20160154335

19 de Junho de 2018

Questão 1
Utilize uma rede NARX para fazer a predição de um passo, até predição de três passos da série temporal $x(n) = ln(1 = cos(n = sen^2(n)))$. Avalie o desempenho mostrando para cada caso os erros de predição. Solucione o problema considerando a NARX uma rede uma Perpcetron de múltiplas camadas com realimentação.

RESOLUÇÃO

Questão 2
Desenvolva um sistema para reconhecer vogais escritas a mão fazendo uso de uma:
a-) Rede neural competitiva.
b-) Rede neural SOM.
Compare o desempenho da duas redes.
Obs. Utilize um banco de dados existente ou gere seu banco de dados usando para cada vogal a escrita de 10 pessoas diferentes.

RESOLUÇÃO

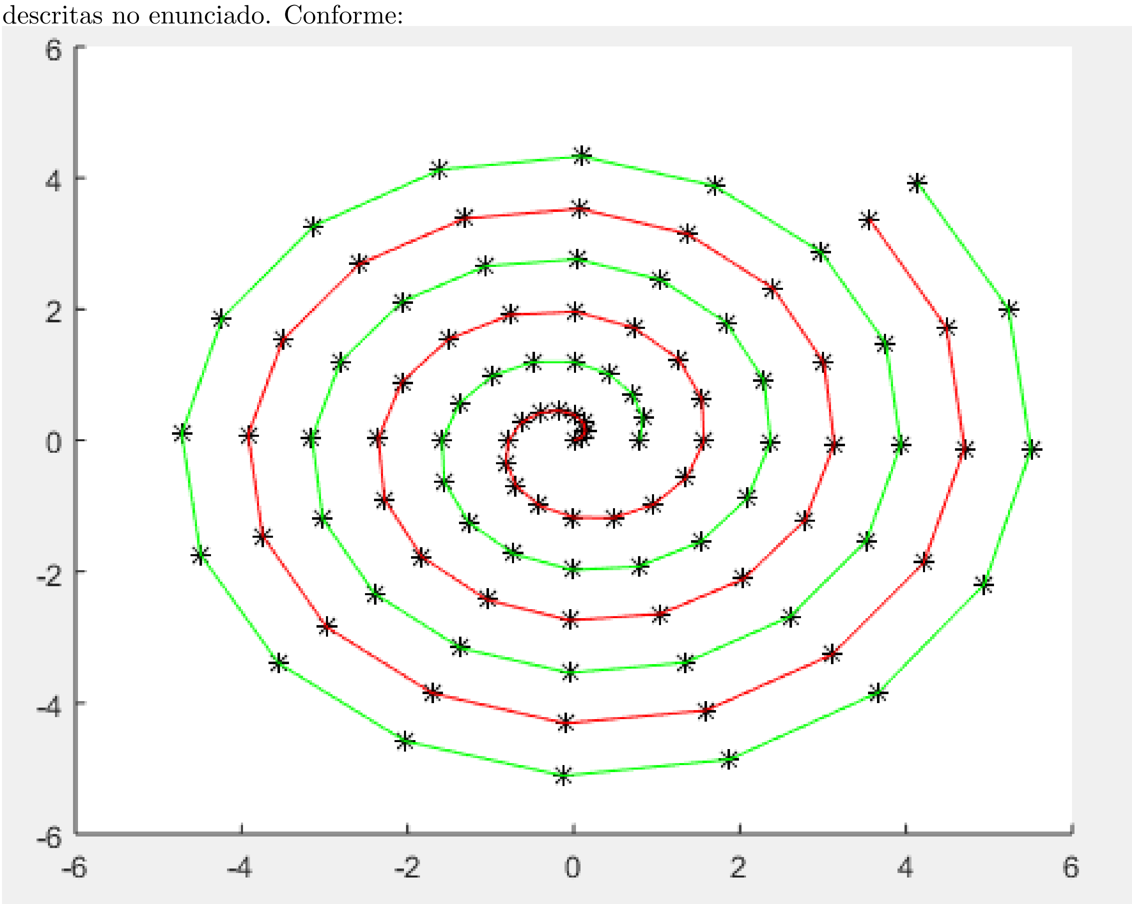
Questão 3
Pesquise e apresente um trabalho sobre a reconstrução de imagens bidimensional e tridimensional usando a rede SOM e a rede Neuro-GAS.

RESOLUÇÃO

Questão 4
Um problema para testar a capacidade de uma rede neural atuar como classificador de padrões é o problema das duas espirais intercaladas. Gere os exemplos de treinamento usando as seguintes equações:
para espiral 1 $x = \frac{\theta}{4}cos\theta \qquad y = \frac{\theta}{4}sen\theta \qquad \theta \geq 0$
para espiral 2 $x = (\frac{\theta}{4} + 0.8)cos\theta \qquad y = (\frac{\theta}{4} + 0.8)sen\theta \qquad \theta \geq 0$
fazendo θ assumir 51 igualmente espaçados valores entre 0 e 20 radianos.
Utilize uma rede competitiva e em seguida uma rede SOM para atuar como classificador autosupervisionado, isto é, a espiral 1 sendo uma classe e espiral 2 sendo outra classe. Para comparar as regiões de decisões formadas pela rede , gere uma grade uniforme com $100x100$ exemplos de teste em um qadrado $[-5, 5]$. Esboce os pontos classificados pela rede.

RESOLUÇÃO

Os dados de entrada foram 51 pontos amostrados em cada uma das espirais descritas no enunciado. Conforme:



Para a resolução com Redes Competitivas, utilizamos o script no Matlab:

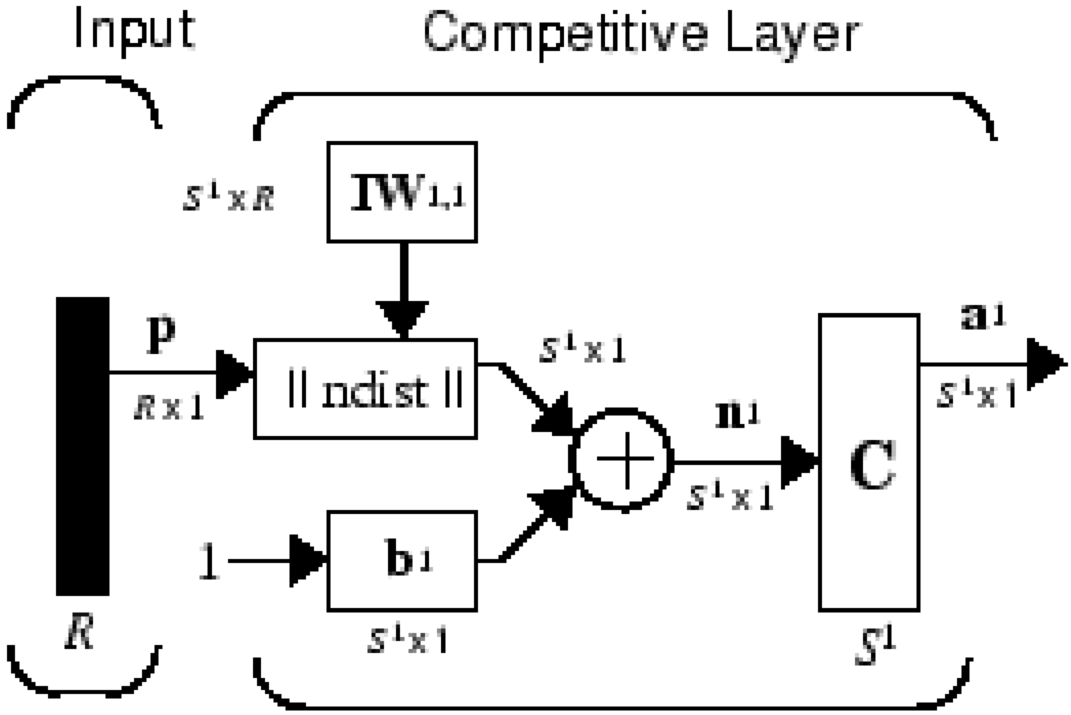
```
%Cria das Espirais e pontos %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%criando elementos de cada classe
nAmostras=51;

%para espiral 1: x=(teta/4)cos(teta) y=(teta/4)sen(teta) teta>=0
%para espiral 2: x= ((teta/4)+0.8)cos(teta)    y= ((teta/4)+0.8)sen(teta)    teta>=0
%fazendo teta assumir 51 igualmente espacados valores entre 0 e 20 radianos.

MAX=20;
MIN=0;
passo=(MAX-MIN)/nAmostras;
C=[];
espiral1=[]; %x1=(teta/4)*cos(teta) y1=(teta/4)*sin(teta) teta>=0
espiral2=[]; %x2= ((teta/4)+0.8)*cos(teta)    y2= ((teta/4)+0.8)*sin(teta)    teta>=0
p1=[];
p2=[];
for teta=MIN:passo:MAX
    x1=(teta/4)*cos(teta);
    y1=(teta/4)*sin(teta);
    x2=((teta/4)+0.8)*cos(teta);
    y2=((teta/4)+0.8)*sin(teta);
    espiral1=[espiral1;x1 y1];
    espiral2=[espiral2;x2 y2];

    p1=[p1;x1 y1 0];
    p2=[p2;x2 y2 100];

end
teta=[MIN:passo:MAX]';
espiral1=espiral1(1:(end-1),:);
espiral2=espiral2(1:(end-1),:);
p1=p1(1:(end-1),:,:);
p2=p2(1:(end-1),:,:);

C=[espiral1;espiral2];
p=[p1;p2];

%plotando em rela a teta
teta=teta(1:(end-1));

hold on
plot(teta,espiral1,'r');
plot(teta,espiral2,'g');
plot(teta,espiral1,'+r');
```

```
plot(teta,espiral2,'*g');

hold off

%plotando em rela  a ao plano x y cartesiano
figure();
hold on

plot(C(:,1),C(:,2),'*black');
plot(espiral1(:,1),espiral1(:,2),'r');
plot(espiral2(:,1),espiral2(:,2),'g');

hold off


%embaralhar ordem das colunas
aux=randperm(2*nAmostras);
p=p';
C=C';
C=C(:,aux);%desse modo trocamos a ordem das colunas
p=p(:,aux);
Target=Target(:,aux);
C=C';

%Normalizar entradas e pesos
p=p/max(max(abs(p)));

%Rede Competitiva
%cria  rede competitiva
nNeuronios=2;
net = competlayer(nNeuronios);


%weights will initialized to the centers of the input
    ranges with the function midpoint
wts = midpoint(nNeuronios,p);

%The initial biases are computed by initcon
biases = initcon(nNeuronios);


%treinar rede
epocas=100;%n mero de epocas
net.trainParam.epochs = epocas;
net = train(net,p);
net.trainFcn

%calculo da resposta
a = sim(net,p);
outputs = vec2ind(a);
```

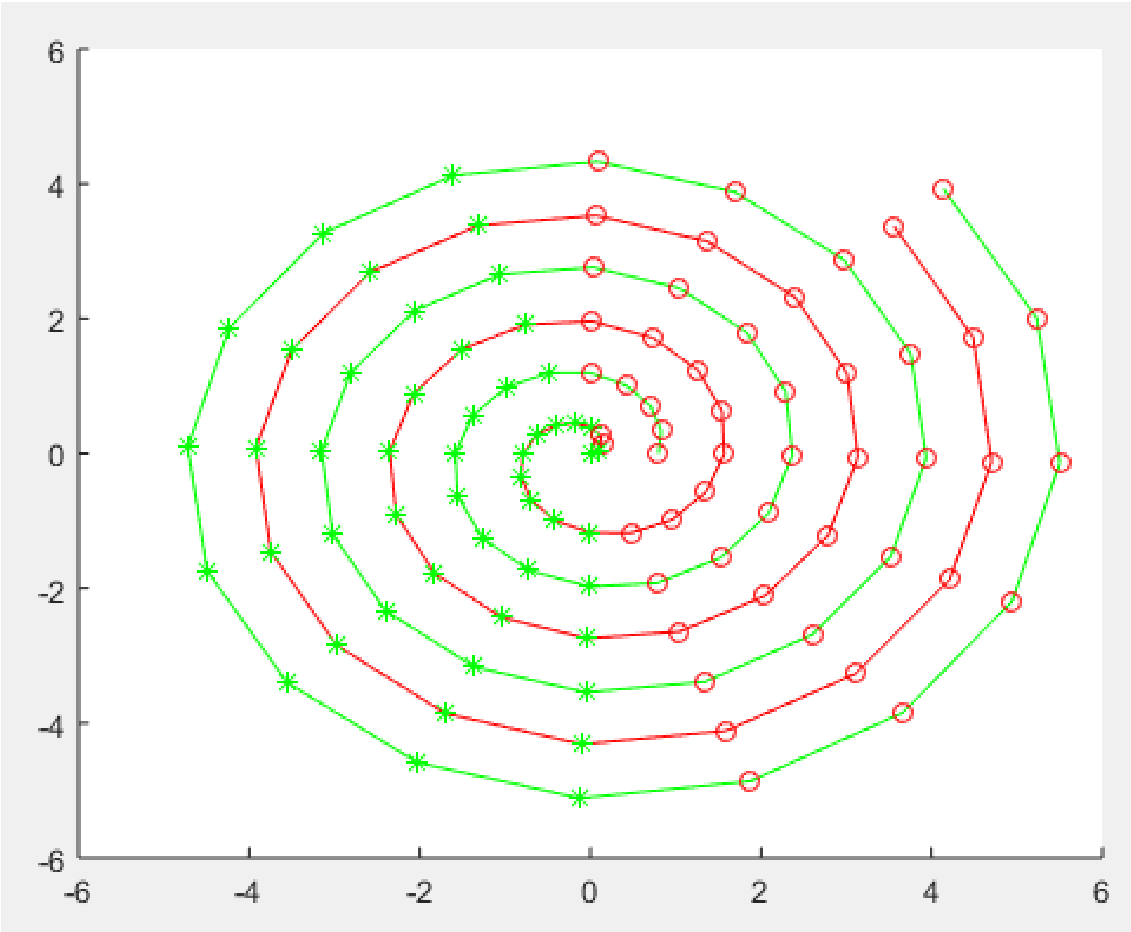
```matlab

%look at the final weights and biases
%net.IW{1,1}
%net.b{1}

%plot
figure();

hold on
plot(espiral1(:,1),espiral1(:,2),'r');
plot(espiral2(:,1),espiral2(:,2),'g');

comp=p';
if(outputs(1)==1 && comp(1,3)==0)
    %classe 1 outputs=1
    %classe 2 outputs=2
    classe1=1;
    classe2=2;
else %outputs(1)==1 && comp(1,3)==100
    %classe 2 outputs=1
    %classe 1 outputs=2
    classe1=2;
    classe2=1;
end

for i=1:2*nAmostras
     x=C(i,1);
     y=C(i,2);
     if(outputs(i)==1 && classe1==1)
         plot(x,y,'ro'); %classe 1
     elseif(outputs(i)==1 && classe1==2)
         plot(x,y,'*g'); %classe 2
     elseif(outputs(i)==2 && classe2==1)
         plot(x,y,'ro'); %classe 1
     else %outputs(i)==2 && classe2==2
         plot(x,y,'*g'); %classe 2
     end
 end

hold off

%erro e matriz de confuso
%plot e calculo do erro
figure();

erro=0;
hold on
plot(espiral1(:,1),espiral1(:,2),'r');
plot(espiral2(:,1),espiral2(:,2),'g');

```

```
%classe1==1 e classe2==2
if(classe1==1)
    for i=1:2*nAmostras

        x=C(i,1);
        y=C(i,2);
        if((outputs(i)==1 && classe1==1)||(outputs(i)==2 && classe2==2))
            plot(x,y,'bo'); %acerto
        else
            erro=erro+1;
            plot(x,y,'*m'); %erro
        end
    end
else %classe1==2 e classe2==1
    for i=1:2*nAmostras
        x=C(i,1);
        y=C(i,2);

        if((outputs(i)==1 && classe1==2)||(outputs(i)==2 && classe2==1))
            plot(x,y,'bo'); %acerto
        else
            erro=erro+1;
            plot(x,y,'*m'); %erro
        end
    end
end




disp(['Nmero de erros = ' num2str(erro)]);
disp(['Percentual de erro= ' num2str(erro/(2*nAmostras)) '%']);

hold off



return;
```

Observe a arquitetura de uma rede competitiva:

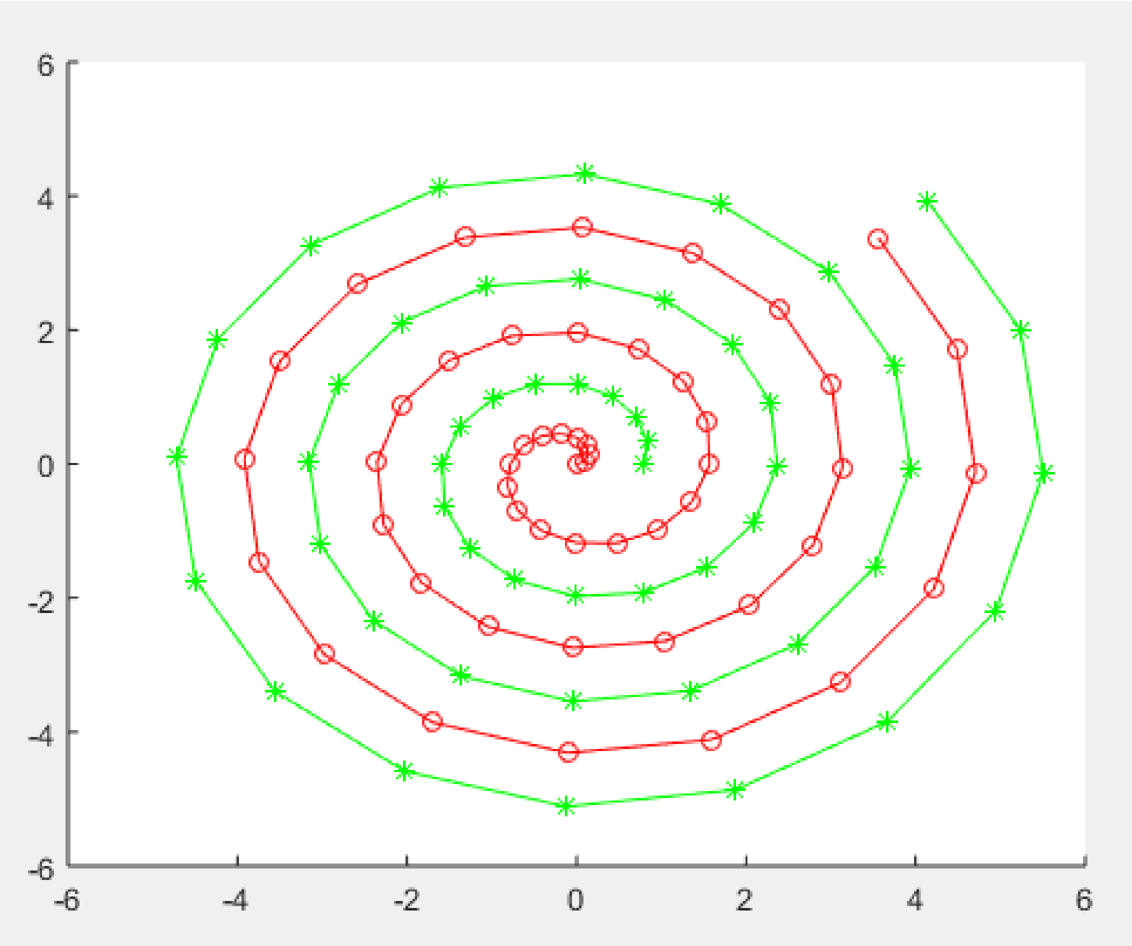


O resultado obtido foi:
Resultado com o grupo de amostragem X e Y puro:

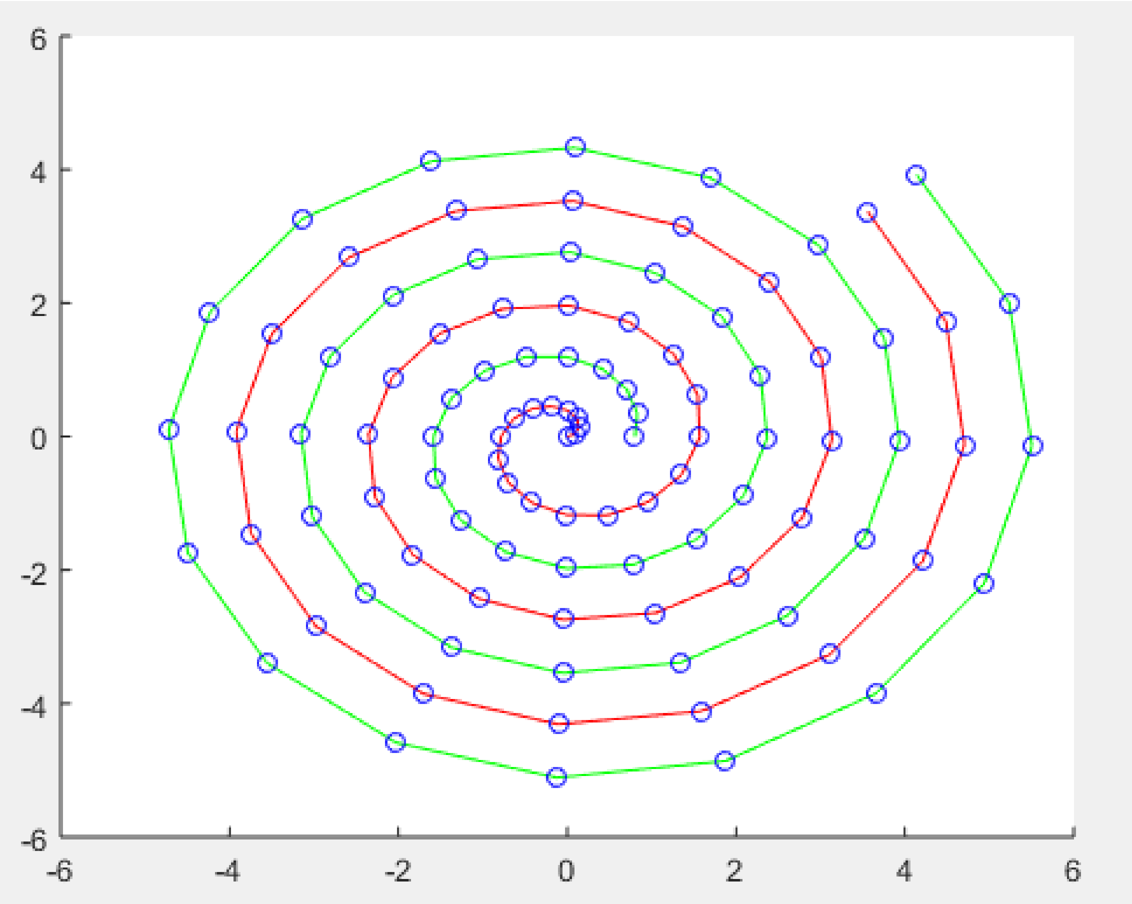


6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

No entanto, para melhorar a performance do programa, fizemos com que essas espirais se distanciassem mais, utilizando um terceiro parâmetro de entrada, 0 para classe 1 e 100 para classe 2. Feito isso, obtivemos:



Com matriz de confusão:



Erro:

```
Número de erros = 0
Percentual de erro= 0%
```

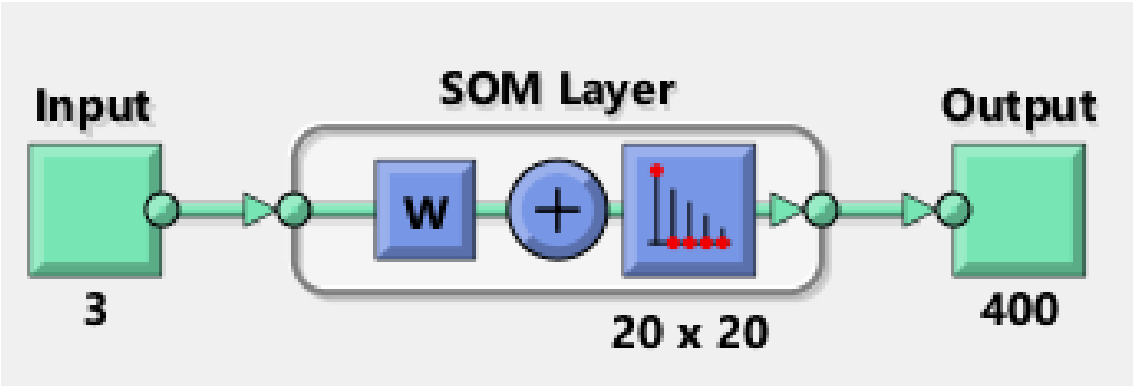
Para a resolução com SOM, utilizamos o script no Matlab:

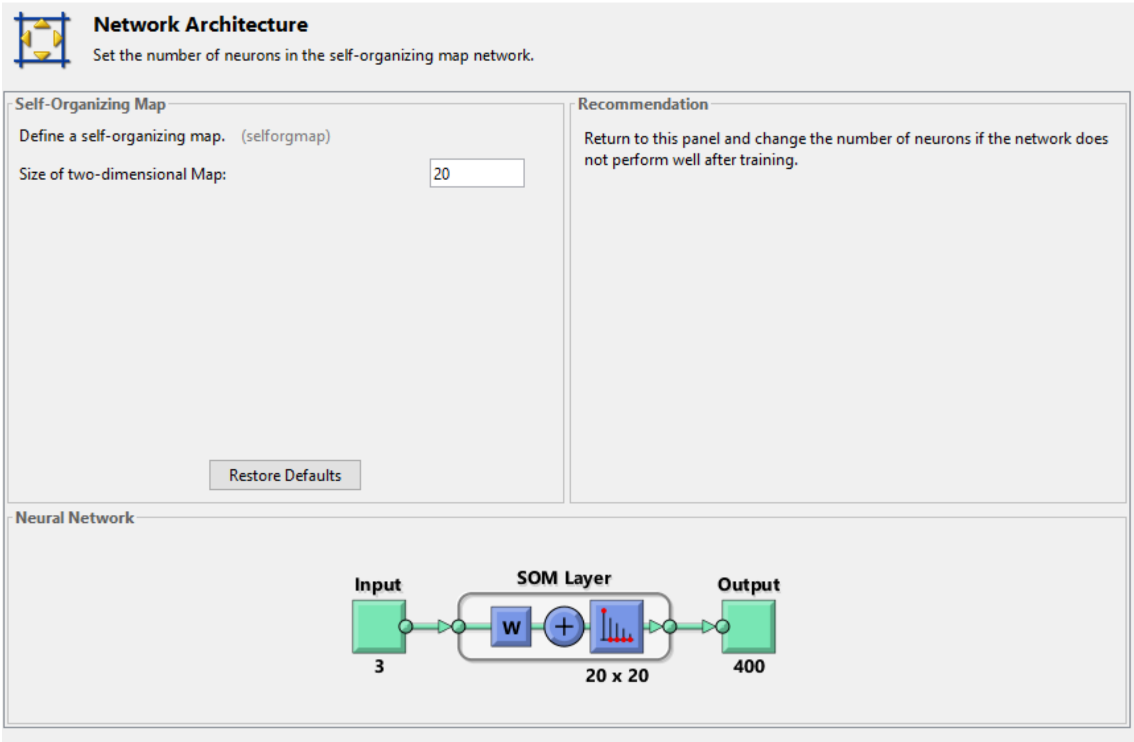
```
%Cria das Espirais e pontos %%%%%%%%%%%%%%%%%%%%%%%%%%%%
%criando elementos de cada classe
nAmostras=51;

%para espiral 1: x=(teta/4)cos(teta) y=(teta/4)sen(teta) teta>=0
%para espiral 2: x= ((teta/4)+0.8)cos(teta) y= ((teta/4)+0.8)sen(teta) teta>=0
%fazendo teta assumir 51 igualmente espacados valores entre 0 e 20 radianos.

MAX=20;
MIN=0;
passo=(MAX-MIN)/nAmostras;
C=[];
espiral1=[]; %x1=(teta/4)*cos(teta) y1=(teta/4)*sin(teta) teta>=0
espiral2=[]; %x2= ((teta/4)+0.8)*cos(teta) y2= ((teta/4)+0.8)*sin(teta) teta>=0
p1=[];
p2=[];
for teta=MIN:passo:MAX
    x1=(teta/4)*cos(teta);
    y1=(teta/4)*sin(teta);
    x2=((teta/4)+0.8)*cos(teta);
    y2=((teta/4)+0.8)*sin(teta);
    espiral1=[espiral1;x1 y1];
    espiral2=[espiral2;x2 y2];

    p1=[p1;x1 y1 0];
    p2=[p2;x2 y2 100];

end
teta=[MIN:passo:MAX]';
espiral1=espiral1(1:(end-1),:);
espiral2=espiral2(1:(end-1),:);
p1=p1(1:(end-1),:,:);
p2=p2(1:(end-1),:,:);

C=[espiral1;espiral2];
p=[p1;p2];

%plotando em rela a teta
teta=teta(1:(end-1));

hold on
```

```
plot(teta,espiral1,'r');
plot(teta,espiral2,'g');
plot(teta,espiral1,'+r');
plot(teta,espiral2,'*g');

hold off

%plotando em rela  a ao plano x y cartesiano
figure();
hold on

plot(C(:,1),C(:,2),'*black');
plot(espiral1(:,1),espiral1(:,2),'r');
plot(espiral2(:,1),espiral2(:,2),'g');

hold off

%dividir entre classe 1 (espiral1, saida 0) e classe 2(
    espiral2,saida 1)
%gerar a resposta desejada t
Target=zeros(1,2*nAmostras);
Target(nAmostras+1:end)=1;


p=p';
%Normalizar entradas e pesos
p=p/max(max(abs(p)));

%Rede SOM
outputs=myNeuralNetworkFunction(p);

%malha -5 a 5 igualmente espaados
inicio=-5;
fim=5;
espaco=(fim-inicio);
tamRede=20*20;

passoMalhaX=espaco/tamRede;
passoMalhaY=espaco/(2*nAmostras);

%plot dos clusters
figure();
hold on;

x=inicio;
for lin=1:length(outputs)
    y=inicio;
    for col=1:2*nAmostras
        if outputs(lin,col)==1 %--> foi ativado
            if(Target(col)==0)%classe 1
```

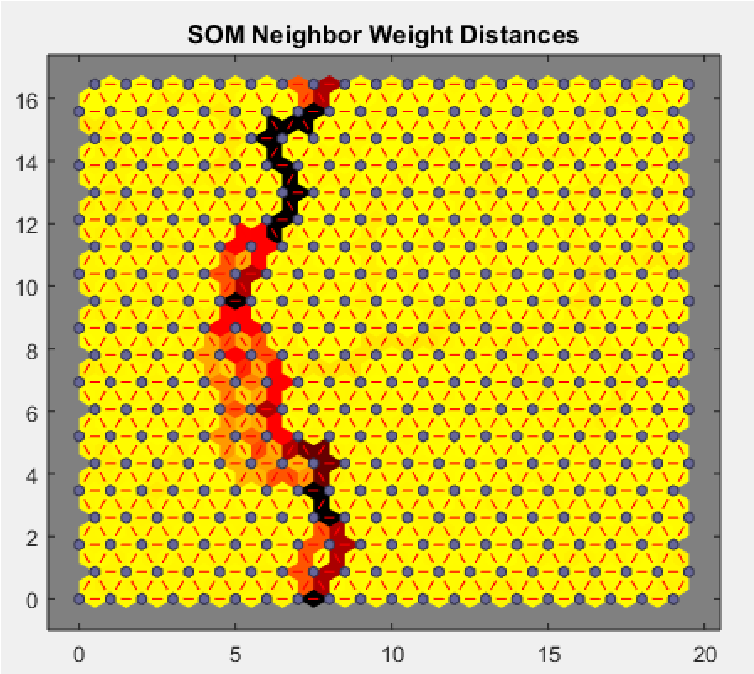
```
                    plot(x,y,'ro');
                else %classe 2
                    plot(x,y,'*g');
                end
            else % outputs(lin,col)==0 --> no foi ativado
                %plot(x,y,'.b');
            end
            y=y+passoMalhaY;
        end
        x=x+passoMalhaX;
end

hold off

return;
```

Para o treinamento, utilizamos a ferrementa nnstart, com as seguintes configurações:

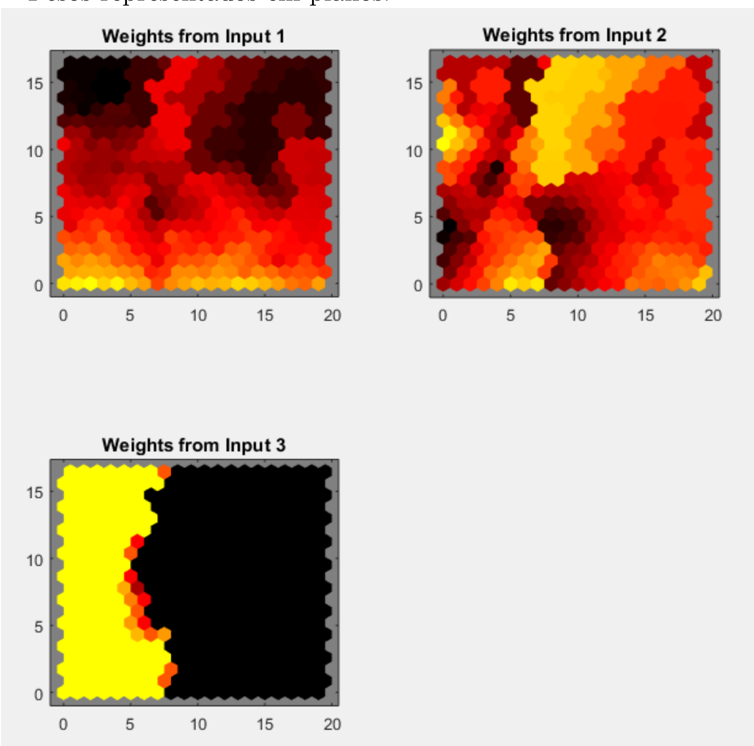



Network Architecture
Set the number of neurons in the self-organizing map network.
Self-Organizing Map
Define a self-organizing map. (selforgmap)
Size of two-dimensional Map:
20
Restore Defaults
Recommendation
Return to this panel and change the number of neurons if the network does not perform well after training.
Neural Network
Input
3
SOM Layer
W
20 x 20
Output
400

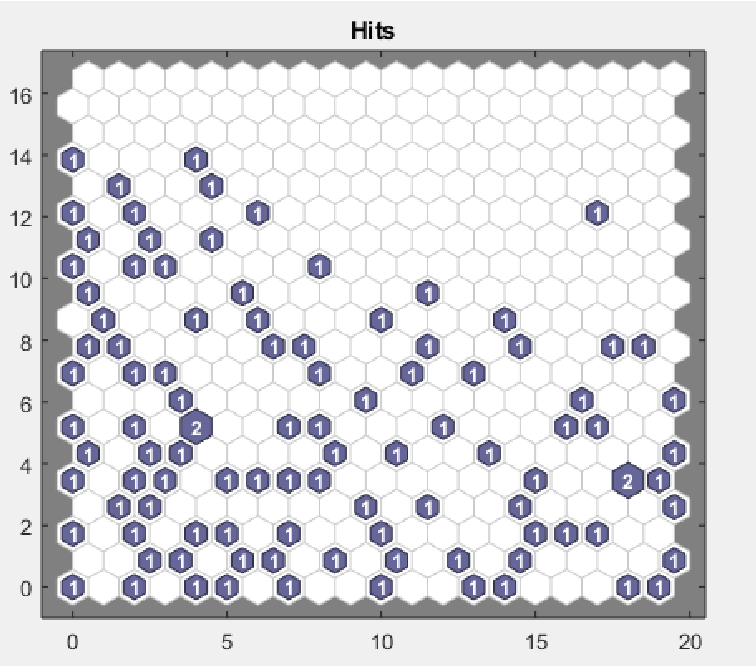
Estas são características da rede treinada:
Distância da vizinhança



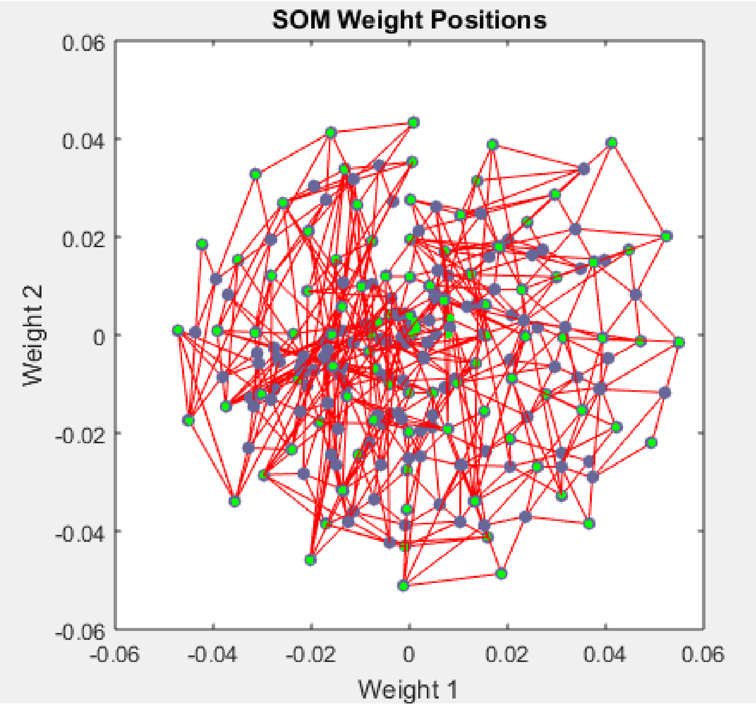
Pesos representados em planos:



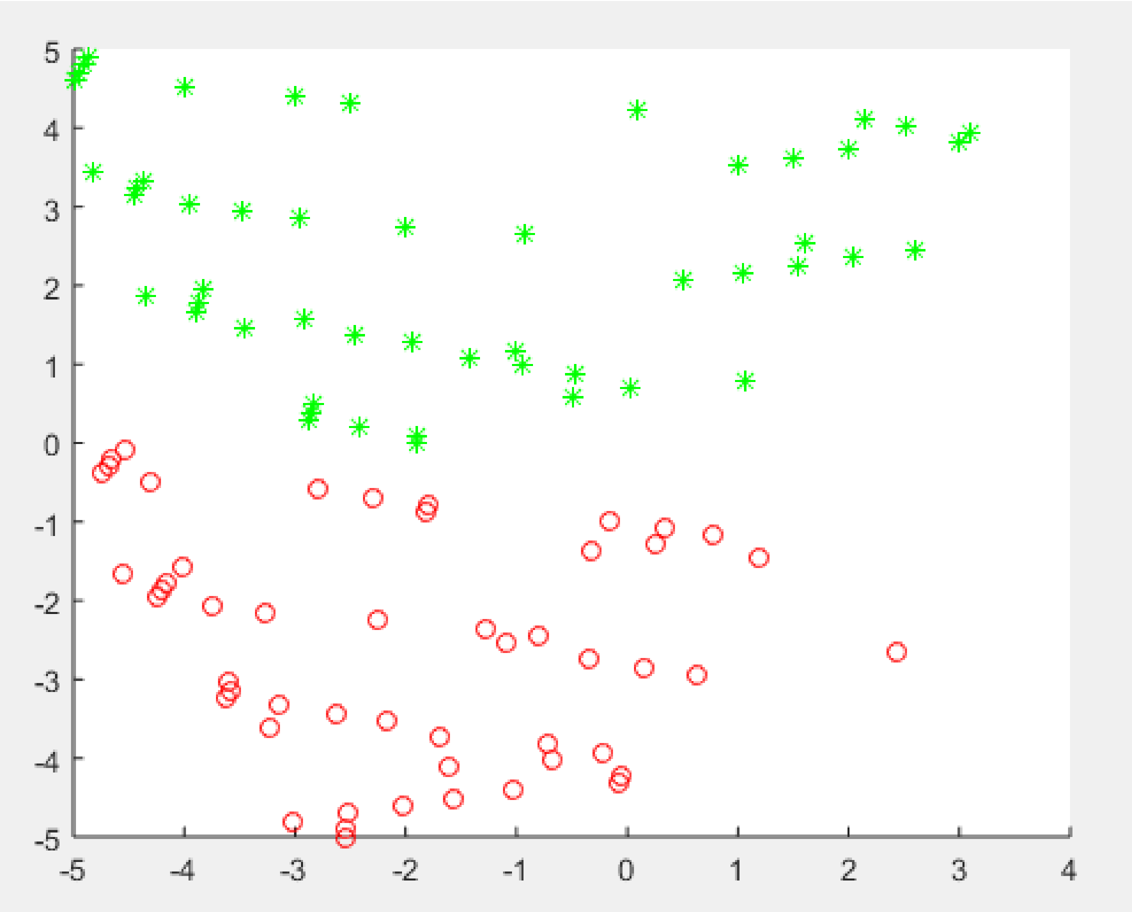
Hits de Amostras:



Rede com a posição dos pesos:



O resultado obtido foi:



Neste, vemos que a função foi capaz de aprender as características dos dados corretamento. Ou seja, cada classe ativou o mesmo grupo de neurônios. Caso não tivesse aprendido as características da entrada veriámos uma mistura entre os os neurônios ativados por cada classe.

Questão 5

A propriedade de ordenação topológica do algoritmo SOM pode ser usada para formar uma representação bidimensional abstrata de um espaço de entrada de alta dimensionalidade. Para investigar esta forma de representação, considere uma grade bidimensional consistindo de 10x10 neurônios que é treinada tendo como entrada os dados oriundos de quatro distribuições gaussianas ,C_1 ,C_2 ,C_3 e C_4, em um espaço de entrada de dimensionalidade igual a oito, isto é ,x= $(x_1, x_2, Kx_8)^t$. Todas as nuvens têm variâncias unitária, mas centros ou vetores média diferentes dados por: m_1= $(0, 0, 0, 0, 0, 0, 0, 0)$,m_2= $(4, 0, 0, 0, 0, 0, 0, 0)$,m_3= $(0, 0, 0, 4, 0, 0, 0, 0)$, ,m_4= $(0, 0, 0, 0, 0, 0, 0, 4)$. Calcule o mapa produzido pelo algoritmo SOM, com cada neurônio do mapa sendo rotulado com a classe particular mais representada pelos pontos de entrada em sua volta. O objetivo é visualizar os dados de dimensão 8 em um espaço de dimensão 2, constiuido pela grade de neurônios.

RESOLUÇÃO

Questão 6
Implemente o algoritmo K-means e considere o dados apresentados na tabela abaixo para serem usando no processo de clustering.

Amostra	x_1	x_2	x_3
1	-7.82	-4.58	-3.97
2	-6.68	3.16	2.71
3	4.36	-2.19	2.09
4	6.72	0.88	2.80
5	-8.64	3.06	3.50
6	-6.87	0.57	-5.45
7	4.47	-2.62	5.76
8	6.73	-2.01	4.18
9	-7.71	2.34	-6.33
10	-6.91	-0.49	-5.68
11	6.18	2.81	5.82
12	6.72	-0.93	-4.04
13	-6.25	-0.26	0.56
14	-6.94	-1.22	1.13
15	8.09	0.20	2.25
16	6.81	0.17	-4.15
17	-5.19	4.24	4.04
18	-6.38	-1.74	1.43
19	4.08	1.30	5.33
20	6.27	0.93	-2.78

a-) Considere que existam três clusters e a inicialização dos centros seja dada por $m_1 = (0, 0, 0)^t$,$m_2 = (1, 1, 1)^t$,$m_3 = (-1, 0, 2)^t$.
b-) Repita o item a considerando que os centros iniciais sejam
$m_1 = (-0.1, 0, 0.1)^t$, $m_2 = (0, -0.1, 0.1)^t$, $m_3 = (-0.1, -0.1, 0.1)^t$.
Compare obtido com o item (a) e explique a razão da diferenças, incluindo o número de interações para alcançar a convergência.

RESOLUÇÃO

Questão 7
Considere o processo de identificação de aglomerados ("clusters") com base em uma técnica hierárquica aglomerativa. Neste problema considere o método de Ward resumido abaixo. Considere também dois critérios para parada do processo aglomerativo no dendograma e identificação do número de aglomerados. O critério R^2 e o critérioo Pseudo T^2. Para o problema considere a tabela de índices de desenvolvimento de paíse (Fonte ONU- 2002, Livro – Análise de dados através de métodos de estatística multivariada– Sueli A.Mingoti) abaixo.
Método de Ward:
a-) Inicialmente, cada elemento é considerado como um único conglomerado
b-) Em cada passo do algoritmo de agrupamento (formação do dendograma) calcule a similaridade fazendo uso da distância Euclidiana ao quadrado entre os conglomerados formados, isto é

$d(C_l, C_i) = \frac{n_l n_i}{n_l + n_i} ||m_l - m_i||^2$ onde,
n_i é o número de elementos no conglomerado C_i;m_i é o centroide do conglomerado C_i dado por $m_i = \frac{1}{n_i}{}^{ni}_{(} j = 1)$
Junte os aglomerados com menor distância.
Cirtério de parada pelo coeficiente R^2
Calcule o coeficiente R^2 em função do número de passos e pare o processo quando for observado um salto elevado no valor do coeficiente. Este ponto determina o número de aglomerados.

RESOLUÇÃO

Questão 9

Considere o problema de análise de componentes principais (PCA), isto é, determinar em uma distribuição de dados as componentes que tenham associadas a elas a maior variância e representar as mesmas no espaço de dados formado pelos autovetores da matriz de correlação. Neste sentido considere o seguinte problema. A tabela abaixo apresenta os dados relativos a amostras de solo. Para cada amostra, tem-se as medidas das porcentagens de areia (X1), sedimentos (X2), argila (X3) e a quantidade de material orgânico (X4). Da referida tabela obtenha as estatísticas descritivas de cada variável, isto é, a média, a mediana, o desvio padrão, os valores máximo e mínimo. Sob estas condições :

a-) Obtenha desta tabela a matriz de covariância.
b-)Desta matriz determine os autovalores ordenados do máximo ao mínimo e os autovetores correspondentes.
c-) Apresente as equações da componentes principais, isto é, cada componente é dada por

$Y_i = \mathbf{e}^t{}_i X_{\mathbf{i}} = e_{1i}X_1 + e_{2i}X_2 + e_{3i}X_3 + e_{4i}X_4 \qquad i = 1,2,3,4$, onde e_{ji} é a componente i do autovetor j.

d-) Calcule os percentuais de variância para cada componente e ordene a classificação das variáveis segundo este critério.

Tabela 1: Tabela: Dados das amostras de solo (Livro – Análise de dados através de métodos de estatística multivariada – Sueli A. Mingoti)

Amostra	Areia (%):X1	Sedimentos(%):X2	Argila(%):X3	Mat. Orgân(%):X4
1	79.9	13.9	6.2	3.3
2	78.5	16.3	7.2	2.5
3	68.9	22.6	8.5	3.6
4	62.2	20.2	17.6	2.8
5	69.2	23.7	7.1	0.9
6	67.8	19.8	12.4	3.8
7	61.3	24.9	13.8	2.2
8	71.6	19.2	9.2	3.6
9	83.7	10.5	5.8	4.4
10	67.1	26.5	6.4	1.4
11	59.8	27.9	12.3	3.5
12	66.7	23.2	10.1	2.9
13	72.8	14.5	12.7	1.9
14	60.9	28.9	10.2	1.5
15	61.4	29.2	9.4	2.5
16	75.0	16.8	8.2	3.1
17	80.5	11.9	7.6	3.8
18	71.3	18.5	10.2	2.6
19	56.6	28.9	14.5	2.8
20	55.9	32.8	11.3	3.1
21	61.5	28.1	10.4	2.7
22	59.2	28.4	12.4	2.8
23	76.9	16.3	6.8	2.9
24	58.0	27.6	14.4	3.4

RESOLUÇÃO

Para resolução das letras a, b, c e d, utilizamos a ferramenta de trabalho Excel.

A) Matriz de Covariância

B) Autovalores ordenados do máximo ao mínimo e os autovetores correspondentes

C) Equações da componentes principais

D) Percentuais de variância para cada componente e ordene a classificação das variáveis segundo este critério.

Questão 10

Pesquise e apresente um estudo sobre BIG DATA.

RESOLUÇÃO

O trabalho sobre Big Data foi feito em relatório separado