Categorização (“clustering”) de dados usando SOM - NNtool

O problema consiste em determinar as classes em que dados experimentais de entrada estão distribuídos sem conhecer de antemão as suas características estatísticas. São selecionadas duas classes distintas cujos vetores representativos são amostrados de duas distribuições gaussianas bidimensionais com médias $\underline{\mathbf{x}}^1$ = [4 4] e $\underline{\mathbf{x}}^2$ = [12 12] e variâncias unitárias. O conjunto de treinamento consiste de 100 pontos com igual probabilidade de amostragem de ambas as distribuições (Ex: gerar 50 pontos de cada distribuição e montar os 100 pontos de treinamento de maneira intercalada). Notar que os dados são apresentados à rede sem identificação de qual classe corresponde cada par de entradas. Cada amostra é dada pelo vetor $\underline{\mathbf{x}}^i = [x^i_1 \;\; x^i_2]$ com x^i_1 e x^i_2 representando grandezas hipotéticas.

Usando o Matlab e a ferramenta NNTOOL pede-se:

1- Construir um mapa SOM de topologia bidimensional de 16 neurônios (4x4) para resolver o problema de categorização. Realize o treinamento e verifique a forma do mapa resultante (plotar os pesos dos neurônios no mapa). Verifique por inspeção visual se o mapa fornece informação de como os dados estão distribuídos.
2- Após a aprendizagem, apresente os dados de treinamento à rede e verifique o número de pares de treinamento que dispara cada neurônio (neurônio vencedor para cada padrão). Gere um conjunto de teste de 50 pontos de cada distribuição e verifique também como estes novos dados são categorizados.
3- Repetir os item 1 para uma rede de dois neurônios (uma vez que temos duas classes distintas). Os dados de entrada serão classificados então nas classes C1 e C2, referentes aos neurônios 1 e 2 da rede. Apresente os pares de entrada do conjunto de treinamento e verifique se estes são corretamente classificados nas duas classes. Gere também um conjunto de teste de 50 pontos de cada distribuição e verifique se estes novos dados são categorizados corretamente.

“Dicas” de funções do Matlab: randn, plotsom.

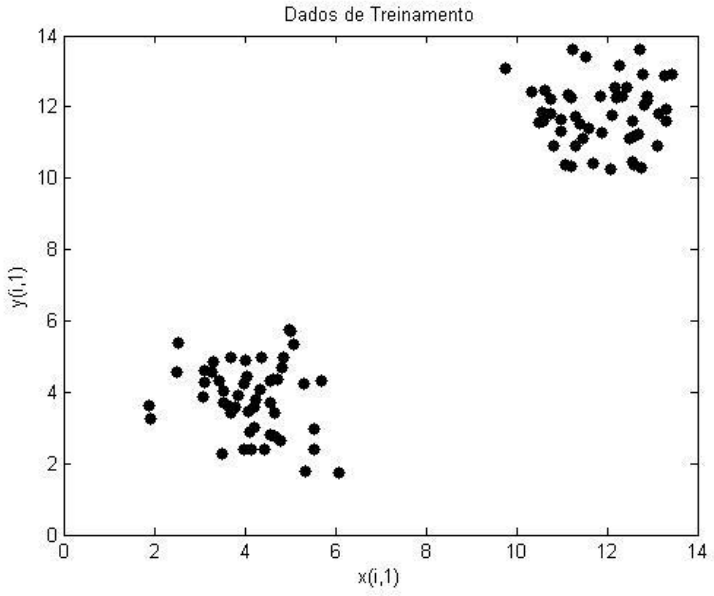
Mapa (Rede) com 16 neurônios

1- Geração dos dados de treinamento e de testes

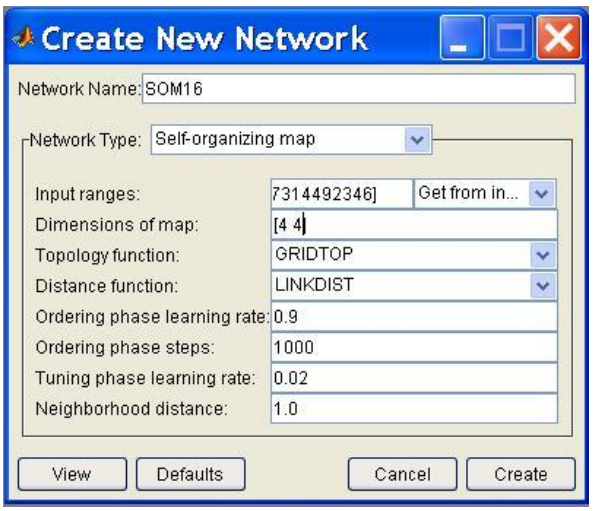
```
%conjunto de treinamento
% limpa dados
j=1;
x1treina=[];
x2treina=[];
x1teste=[];
xtreina=[];
x2teste=[];
x1teste=[];

for i=1:50
   x1treina(:,i)= 4 + randn(2,1);
   x2treina(:,i)= 12 + randn(2,1);
   xtreina(:,j)= x1treina(:,i);
   xtreina(:,j+1)=x2treina(:,i);
   j=j+2;
end

%conjunto de teste
for i=1:50
   x1teste(:,i)= 4 + randn(2,1);
   x2teste(:,i)= 12 + randn(2,1);
end
```

Verificando a distribuição dos dados de xtreina:

```
plot(xtreina(1,:),xtreina(2,:),'.k','markersize',20)
```

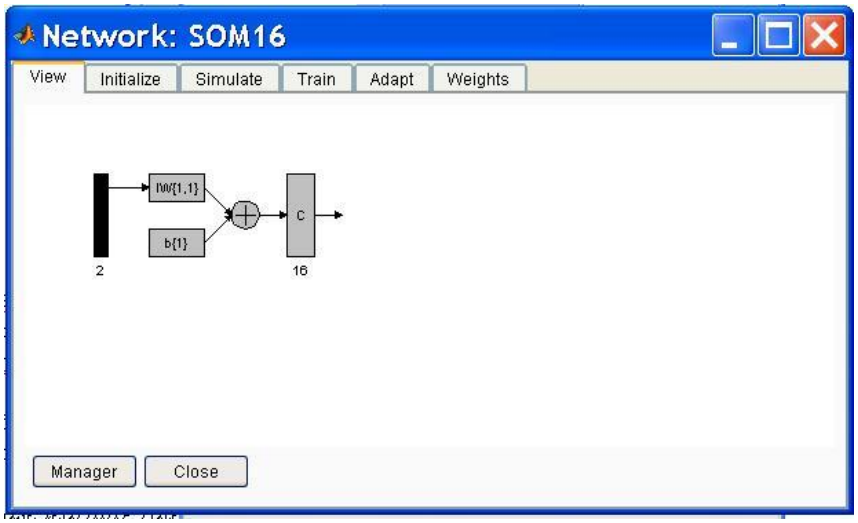


2- Construindo a rede



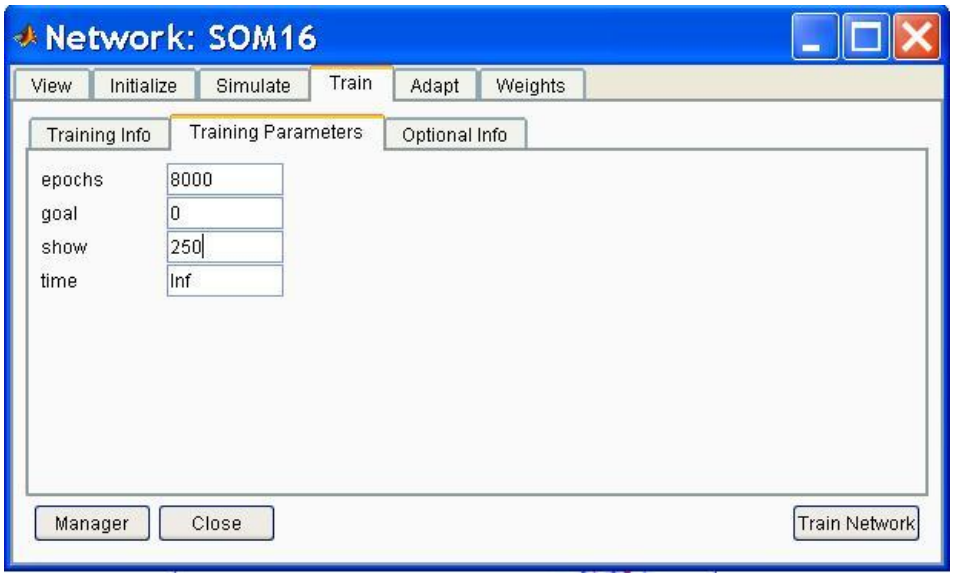
Network type : Tipo de rede = Self-organizing map.
Input Ranges: as faixas dos dados de entrada devem ser obtidas dos dados de treinamento (xtreina).
Dimensions of map: Dimensões do mapa: [4 4] →4x4=16 neurônios.
Topology function: Topologia da rede: gridtop = retangular, hextop = hexagonal, randtop = aleatória.
Função distância para determinar vizinho mais próximo: linkdist = número de passos ou conexões para ir de um neurônio a outro, dist = distância euclideana, mandist = distância "manhattan" (D = sum(abs(x-y))).
Ordering phase learning rate: taxa de aprendizagem no início da fase de ordenação.
Ordering phase steps: número de passos da fase de ordenação. Durante estes passos iniciais a taxa de amostragem irá diminuir do valor definido pela "Ordering phase learning rate" até o valor definido pela "tuning phase learning rate", e a distância considerado para a vizinhança do neurônio vencedor variará do raio da rede até a "Neighborhood distance" definida na configuração.
Tuning phase learning rate: taxa de aprendizagem da fase de convergência.
Neighborhood distance: vizinhança do nó vencedor durante a fase de convergência.

Rede criada:



3. Treinando a rede

Usando como dados de treinamento xtreina para o nntool e os parâmetros a seguir[1]:

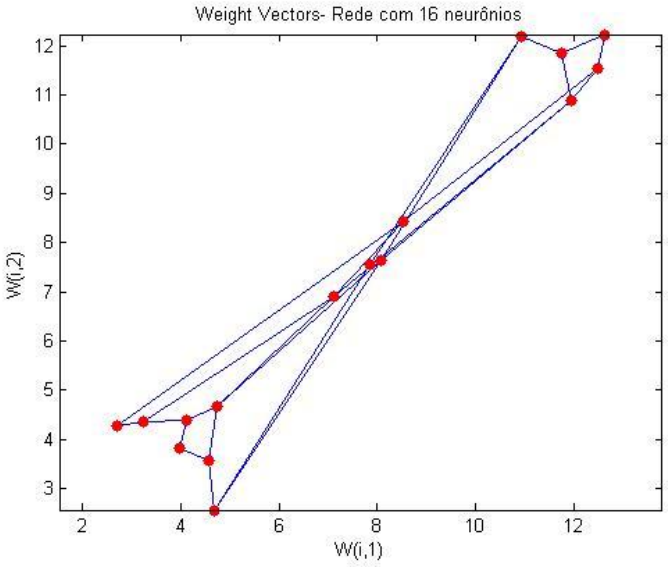


Após treinamento podemos obter os pesos de cada neurônio da rede, assim como qual neurônio foi disparado por cada padrão de treinamento. Para tal, exportamos a rede "SOM16" e as saídas "SOM16_outputs".

Para "plotarmos" o gráfico dos pesos utilizamos:

```
>> plotsom(SOM16.iw{1,1},SOM16.layers{1}.distances)
```

Obtendo o resultado a seguir, onde podemos notar que os pesos concentram-se em 2 a 3 aglomerados.



[1] Utilzou-se o seguinte critério para o número de épocas 1000+ 500*número de neurônios na rede (ref. Haykin)

Pesos obtidos:

```
>> SOM16.iw{1,1}

ans =

    3.9731    3.8179
    4.5689    3.5740
    4.6811    2.5641
    7.8520    7.5555 (neurônio 4)
    4.1129    4.3824
    4.7383    4.6826
    8.0710    7.6262  (neurônio 7)
   10.9224   12.1830
    3.2432    4.3664
    7.1143    6.9026  (neurônio 10)
   11.9365   10.8998
   11.7498   11.8585
    2.7061    4.2831
    8.5354    8.4348 (neurônio 14)
   12.4779   11.5494
   12.6266   12.2318
```

4. Verificando a alocação de cada padrão de treinamento na rede (neurônio vencedor para cada padrão).

A saída SOM16_outputs mostra em que neurônio foi alocado cada padrão. A notação (j,i) indica que o padrão i (no exemplo i = 1 a 100) foi alocado no neurônio j.

```
SOM16_outputs =

(6,1)     1    (3,21)    1    (3,41)    1    (5,61)    1    (6,81)    1
(11,2)    1    (12,22)   1    (8,42)    1    (8,62)    1    (16,82)   1
(3,3)     1    (6,23)    1    (9,43)    1    (3,63)    1    (13,83)   1
(15,4)    1    (16,24)   1    (8,44)    1    (8,64)    1    (16,84)   1
(6,5)     1    (1,25)    1    (6,45)    1    (3,65)    1    (9,85)    1
(16,6)    1    (16,26)   1    (8,46)    1    (15,66)   1    (8,86)    1
(9,7)     1    (3,27)    1    (3,47)    1    (1,67)    1    (5,87)    1
(11,8)    1    (12,28)   1    (11,48)   1    (16,68)   1    (11,88)   1
(9,9)     1    (2,29)    1    (6,49)    1    (13,69)   1    (9,89)    1
(15,10)   1    (16,30)   1    (15,50)   1    (11,70)   1    (8,90)    1
(5,11)    1    (13,31)   1    (1,51)    1    (1,71)    1    (6,91)    1
(16,12)   1    (16,32)   1    (8,52)    1    (15,72)   1    (12,92)   1
(3,13)    1    (9,33)    1    (3,53)    1    (1,73)    1    (3,93)    1
(16,14)   1    (8,34)    1    (12,54)   1    (8,74)    1    (15,94)   1
(2,15)    1    (3,35)    1    (3,55)    1    (1,75)    1    (5,95)    1
(16,16)   1    (8,36)    1    (12,56)   1    (11,76)   1    (11,96)   1
(6,17)    1    (3,37)    1    (6,57)    1    (6,77)    1    (13,97)   1
(16,18)   1    (8,38)    1    (16,58)   1    (11,78)   1    (11,98)   1
(6,19)    1    (9,39)    1    (5,59)    1    (1,79)    1    (1,99)    1
(8,20)    1    (16,40)   1    (8,60)    1    (11,80)   1    (11,100)  1
```

Devemos notar que xtreina é formado por padrões alternados das duas gaussianas (padrão x1 corresponde a índice ímpar e x2 a índice par). Então, da tabela anterior obtemos:

Disposição espacial dos neurônios (nntool):

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Ocorrência dos padrões de treinamento: (A=x1, B=x2, N =nenhum padrão)

1-(8A)	2-(2A)	3-(13A)	4-(N)
5-(5A)	6-(10A)	7-(N)	8-(15B)
9- (7A)	10-(N)	11-(12B)	12-(4B)
13- (5A)	14-(N)	15- (6B)	16- (12B)

É fácil verificar que o mapa mostra que os dados da mesma distribuição são alocados em neurônios próximos. Também, analisando a tabela anterior e o gráfico dos pesos, pode-se verificar que os neurônios 4, 7, 10 e 14 , localizados próximo ao centro do gráfico, não tiveram nenhum dado de treinamento alocado aos mesmos. Retirando tais neurônios do gráfico, sobressaem-se dois aglomerados (“clusters”). Isto nos leva à conclusão que os dados estão divididos em duas classes estatísticas.

5. Verificando a alocação de cada padrão de teste na rede (neurônio vencedor para cada padrão).

Usamos a opção SIMULATE do nntool e obtemos exportamos as saídas para o workspace.

Considerando que o padrão pertença à classe A (ou C1) se ativar os neurônios 1,2,3,5,6, 9 e 13 e à classe B (ou C2) se ativar os neurônios 8,11,12, 15 e 16, temos:

Para o conjunto X1teste:

SOM16_outputsX1teste =

(3,1) 1	(13,21) 1	(1,41) 1
(1,2) 1	(6,22) 1	(9,42) 1
(6,3) 1	(3,23) 1	(2,43) 1
(6,4) 1	(3,24) 1	(9,44) 1
(2,5) 1	(2,25) 1	(1,45) 1
(5,6) 1	(1,26) 1	(5,46) 1
(13,7) 1	(5,27) 1	(5,47) 1
(3,8) 1	(1,28) 1	(3,48) 1
(9,9) 1	(3,29) 1	(9,49) 1
(3,10) 1	(13,30) 1	(1,50) 1
(2,11) 1	(5,31) 1	
(13,12) 1	(9,32) 1	
(2,13) 1	(9,33) 1	
(3,14) 1	(6,34) 1	
(2,15) 1	(1,35) 1	
(5,16) 1	(2,36) 1	
(6,17) 1	(3,37) 1	
(2,18) 1	(2,38) 1	
(1,19) 1	(13,39) 1	
(5,20) 1	(6,40) 1	

Ocorrência dos padrões de treinamento: (A=x1, N =nenhum padrão)

1-(8A)	2-(9A)	3-(9A)	4-(N)
5-(7A)	6-(6A)	7-(N)	8-(N)
9- (7A)	10-(N)	11-(N)	12-(N)
13- (5A)	14-(N)	15- (N)	16- (N)

É fácil notar que todos os dados foram classificados corretamente, ativando os neurônios associados à classe do padrão x1.

Para o conjunto X2teste:

SOM16_outputsX2teste =

(16,1) 1	(11,21) 1	(11,41) 1
(11,2) 1	(11,22) 1	(15,42) 1
(16,3) 1	(8,23) 1	(11,43) 1
(16,4) 1	(15,24) 1	(16,44) 1
(8,5) 1	(16,25) 1	(11,45) 1
(11,6) 1	(15,26) 1	(12,46) 1
(8,7) 1	(11,27) 1	(16,47) 1
(16,8) 1	(8,28) 1	(15,48) 1
(8,9) 1	(12,29) 1	(15,49) 1
(16,10) 1	(8,30) 1	(11,50) 1
(11,11) 1	(16,31) 1	
(16,12) 1	(8,32) 1	
(8,13) 1	(8,33) 1	
(16,14) 1	(15,34) 1	
(12,15) 1	(16,35) 1	
(16,16) 1	(8,36) 1	
(12,17) 1	(8,37) 1	
(14,18) 1	(8,38) 1	
(11,19) 1	(15,39) 1	
(8,20) 1	(16,40) 1	

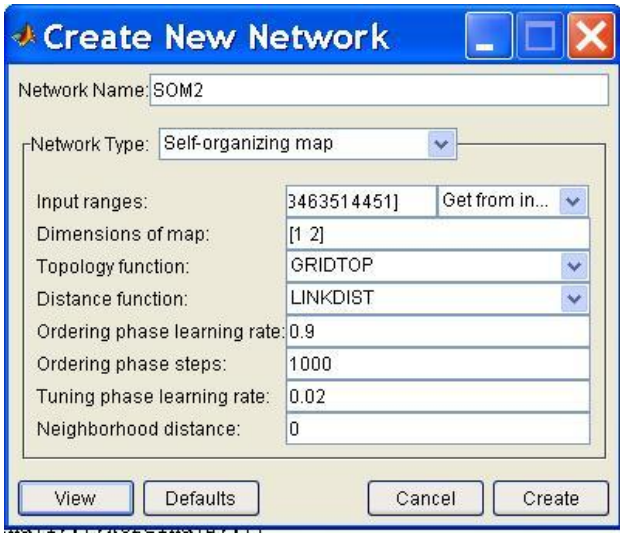
Ocorrência dos padrões de treinamento: (B = x2, N =nenhum padrão)

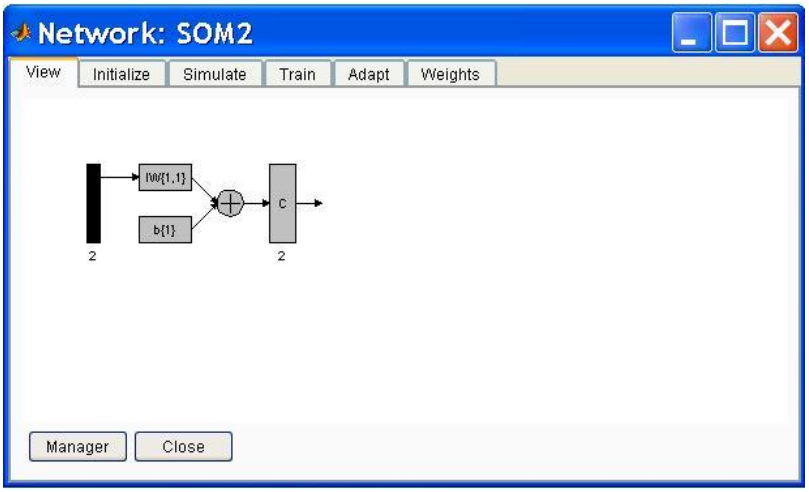
1-(N)	2-(N)	3-(N)	4-(N)
5-(N)	6-(N)	7-(N)	8-(13B)
9- (N)	10-(N)	11-(11B)	12-(4B)
13- (N)	14-(1B)	15- (7B)	16- (14B)

Os dados foram classificados corretamente, ativando os neurônios associados à classe do padrão x2., com exceção do padrão número 18 que ativou o neurônio 14 (1 erro em 50 →2% erro).

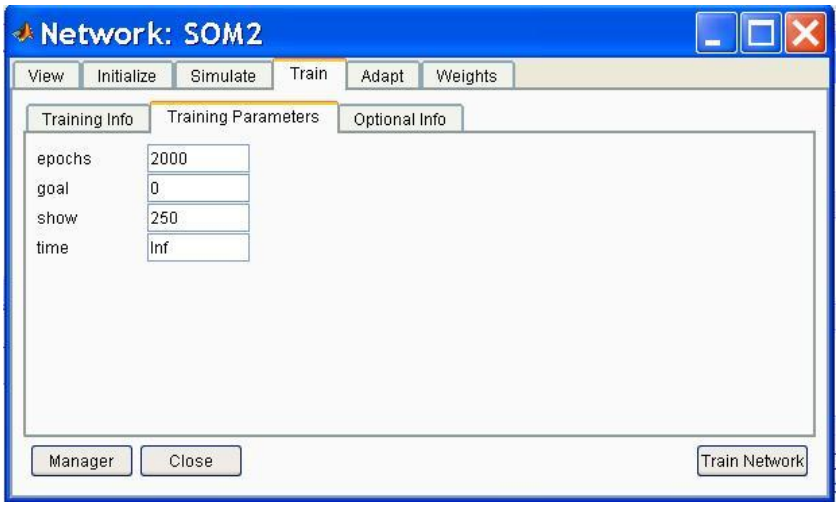
Rede com 2 (dois) neurônios:

Notar que devemos reduzir a vizinhança topológica para 0 na fase de convergência, pois só temos dois neurônios!

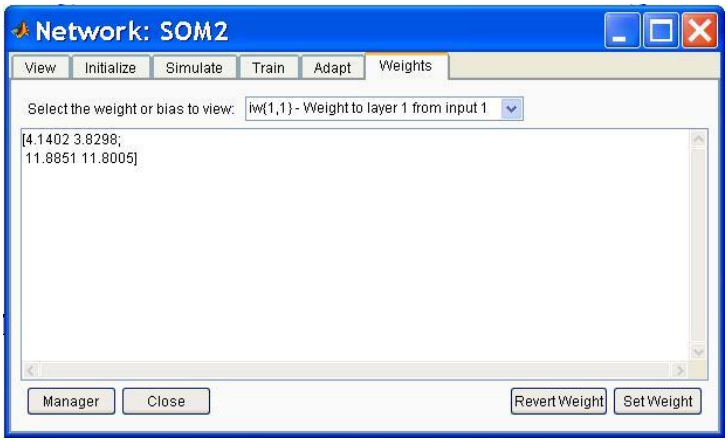




Treinamento



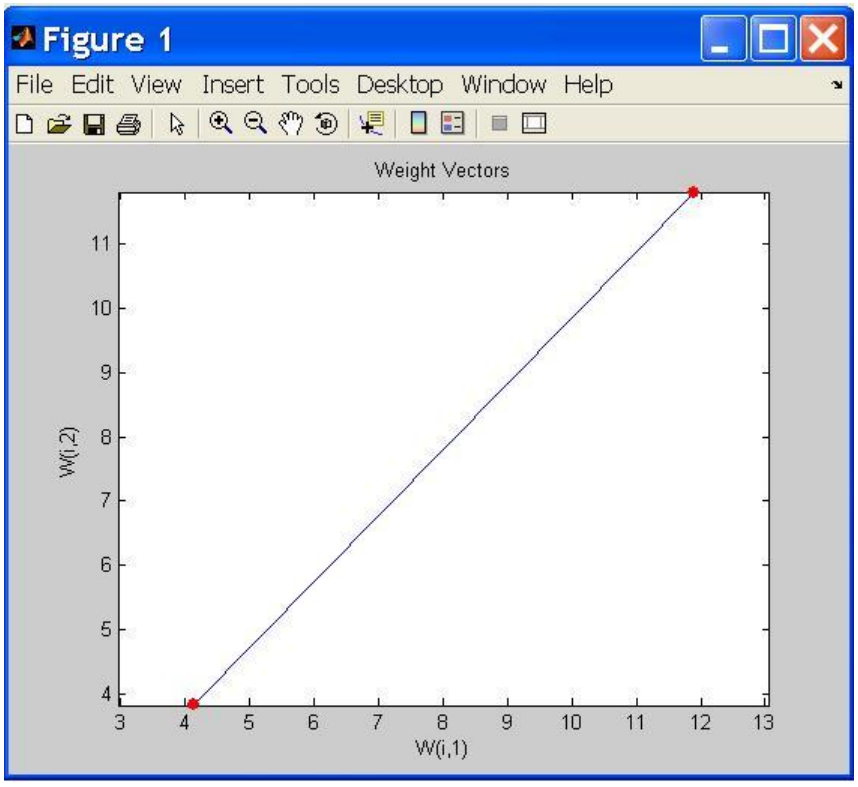
Pesos obtidos:



Pesos: w1 = [4.1402 3.8298]^T, w2 = [11.8851 11.8005]^T

Para plotar os pesos dos neurônios de saída (exportar as saídas e a rede para o workspace):

```
>> plotsom(SOM2.iw{1,1},SOM2.layers{1}.distances)
```



Vemos que a alocação espacial dos neurônios reflete a estatística dos dados de entrada.

Verificando a alocação dos dados de treinamento nas classes dadas pelos neurônios de saída.

A saída SOM2_outputs mostra em que neurônio foi alocado cada padrão. A notação (j,i) indica que o padrão i foi alocado no neurônio j. Notar que os padrões de treinamento foram obtidos alternadamente das distribuições normais, e portanto a alocação alternada obtida, que mostra a classificação correta dos dados de treinamento:

SOM2_outputs =

(1,1)	1	(1,21)	1	(1,41)	1	(1,61)	1	(1,81)	1
(2,2)	1	(2,22)	1	(2,42)	1	(2,62)	1	(2,82)	1
(1,3)	1	(1,23)	1	(1,43)	1	(1,63)	1	(1,83)	1
(2,4)	1	(2,24)	1	(2,44)	1	(2,64)	1	(2,84)	1
(1,5)	1	(1,25)	1	(1,45)	1	(1,65)	1	(1,85)	1
(2,6)	1	(2,26)	1	(2,46)	1	(2,66)	1	(2,86)	1
(1,7)	1	(1,27)	1	(1,47)	1	(1,67)	1	(1,87)	1
(2,8)	1	(2,28)	1	(2,48)	1	(2,68)	1	(2,88)	1
(1,9)	1	(1,29)	1	(1,49)	1	(1,69)	1	(1,89)	1
(2,10)	1	(2,30)	1	(2,50)	1	(2,70)	1	(2,90)	1
(1,11)	1	(1,31)	1	(1,51)	1	(1,71)	1	(1,91)	1
(2,12)	1	(2,32)	1	(2,52)	1	(2,72)	1	(2,92)	1
(1,13)	1	(1,33)	1	(1,53)	1	(1,73)	1	(1,93)	1
(2,14)	1	(2,34)	1	(2,54)	1	(2,74)	1	(2,94)	1
(1,15)	1	(1,35)	1	(1,55)	1	(1,75)	1	(1,95)	1
(2,16)	1	(2,36)	1	(2,56)	1	(2,76)	1	(2,96)	1
(1,17)	1	(1,37)	1	(1,57)	1	(1,77)	1	(1,97)	1
(2,18)	1	(2,38)	1	(2,58)	1	(2,78)	1	(2,98)	1
(1,19)	1	(1,39)	1	(1,59)	1	(1,79)	1	(1,99)	1
(2,20)	1	(2,40)	1	(2,60)	1	(2,80)	1	(2,100)	1

O neurônio 1 está associado à classe C1, referente à distribuição de x1 e o neurônio 2 está associado à classe C2, referente à distribuição de x2.

Simulando a rede para os dados de teste x1teste e x2teste

Usamos a opção SIMULATE do nntool e obtemos exportamos as saídas para o workspace.

Para o conjunto X1teste:

SOM2_outputsX1teste =

(1,1)	1	(1,21)	1	(1,41)	1
(1,2)	1	(1,22)	1	(1,42)	1
(1,3)	1	(1,23)	1	(1,43)	1
(1,4)	1	(1,24)	1	(1,44)	1
(1,5)	1	(1,25)	1	(1,45)	1
(1,6)	1	(1,26)	1	(1,46)	1
(1,7)	1	(1,27)	1	(1,47)	1
(1,8)	1	(1,28)	1	(1,48)	1
(1,9)	1	(1,29)	1	(1,49)	1
(1,10)	1	(1,30)	1	(1,50)	1
(1,11)	1	(1,31)	1		
(1,12)	1	(1,32)	1		
(1,13)	1	(1,33)	1		
(1,14)	1	(1,34)	1		
(1,15)	1	(1,35)	1		
(1,16)	1	(1,36)	1		
(1,17)	1	(1,37)	1		
(1,18)	1	(1,38)	1		
(1,19)	1	(1,39)	1		
(1,20)	1	(1,40)	1		

Vemos que todos os dados foram classificados corretamente para apenas uma única saída, neurônio 1, correspondendo à classe C1.

Para o conjunto de teste X2teste:

SOM2_outputsX2teste =

(2,1)	1	(2,21)	1	(2,41)	1
(2,2)	1	(2,22)	1	(2,42)	1
(2,3)	1	(2,23)	1	(2,43)	1
(2,4)	1	(2,24)	1	(2,44)	1
(2,5)	1	(2,25)	1	(2,45)	1
(2,6)	1	(2,26)	1	(2,46)	1
(2,7)	1	(2,27)	1	(2,47)	1
(2,8)	1	(2,28)	1	(2,48)	1
(2,9)	1	(2,29)	1	(2,49)	1
(2,10)	1	(2,30)	1	(2,50)	1
(2,11)	1	(2,31)	1		
(2,12)	1	(2,32)	1		
(2,13)	1	(2,33)	1		
(2,14)	1	(2,34)	1		
(2,15)	1	(2,35)	1		
(2,16)	1	(2,36)	1		
(2,17)	1	(2,37)	1		
(2,18)	1	(2,38)	1		
(2,19)	1	(2,39)	1		
(2,20)	1	(2,40)	1		

Vemos que todos os dados foram classificadas corretamente para o outro neurônio 2, correspondendo à classe C2.