

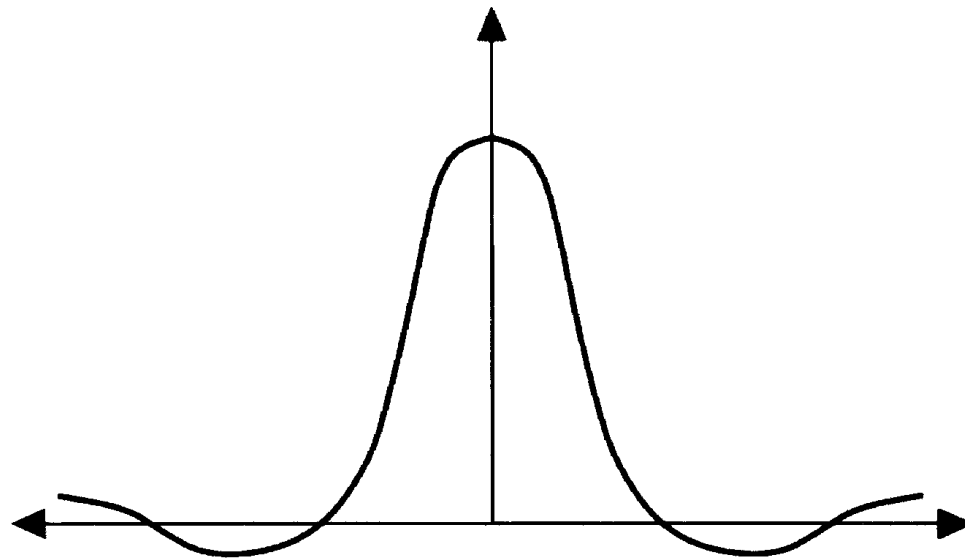
Self-organizing Maps

- ♦ La red neuronal SOM busca *respetar la topología* de los datos.
- ♦ Neuronas que representan grupos de patrones similares se encuentran próximas dentro de la arquitectura.
- ♦ Utiliza el concepto de inhibición lateral.

Inhibición lateral

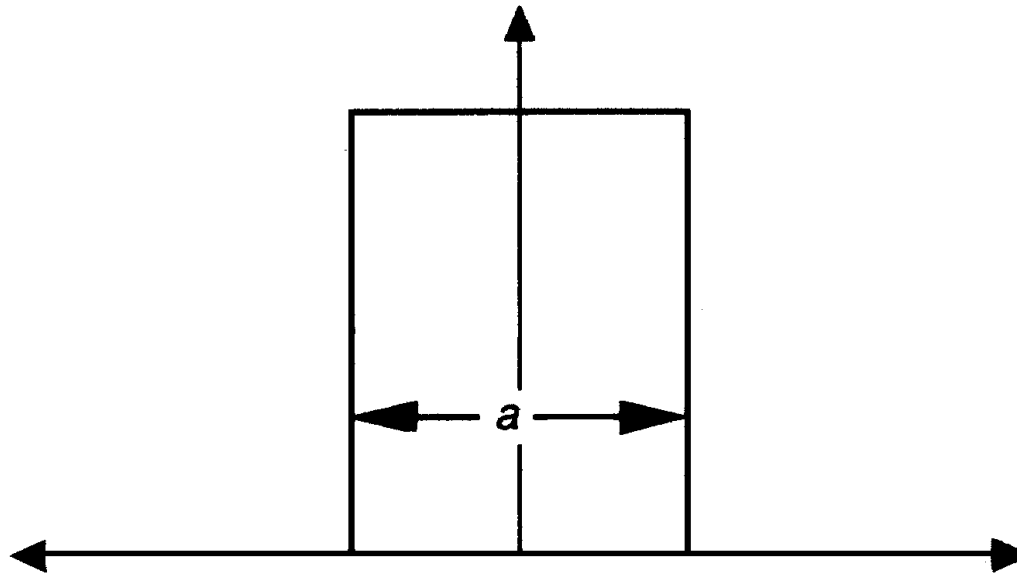
- ♦ La inhibición lateral es un proceso que ocurre en las redes neuronales biológicas.
- ♦ Se forman conexiones laterales entre las neuronas de un nivel dado.
- ♦ La fuerza de dicha conexión es inversamente proporcional a la distancia entre las neuronas (cada neurona está más relacionada con sus vecinas que con las que no lo son).
- ♦ Ej: sistema de visión humano

Función sombrero



- ♦ Cuando se excita a la neurona central, también se excita al conjunto de neuronas más próximas. A medida que la distancia aumenta, la excitación cae hasta llegar a ser una inhibición.

Función sombrero aproximada



- ♦ La distancia a define el tamaño del entorno de neuronas vecinas de la ganadora que van a participar del aprendizaje (junto con la unidad central).

Entrenamiento

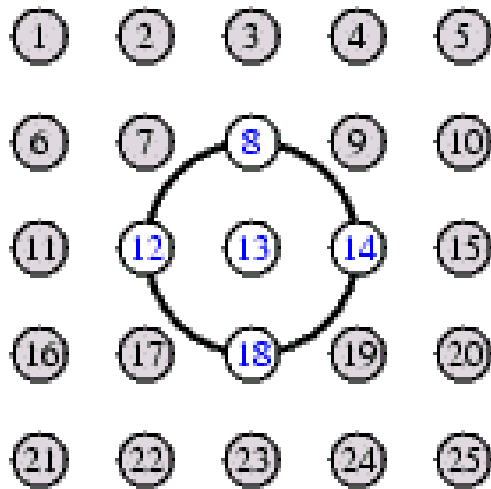
- ◆ Es similar a la red CPN

$$w_{\text{new}} = w_{\text{old}} + \alpha (\text{Input} - w_{\text{old}})$$

pero se aplica a la ganadora y su **vecindad**.

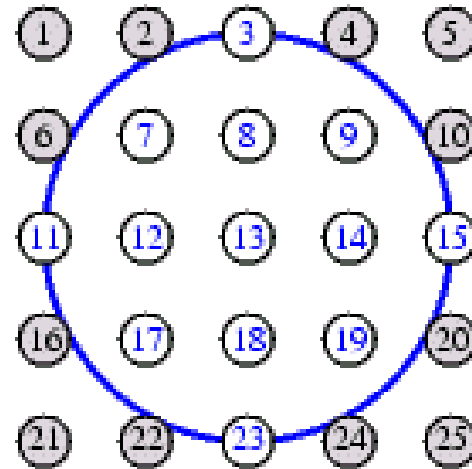
- ◆ El tamaño de la vecindad se va reduciendo a medida que pasan los ciclos.
- ◆ Es conveniente comenzar con una vecindad grande e ir reduciéndola hasta especializarse en cada neurona competitiva.

Vecindad



$N_{13}(1)$

$$N_{13}(1) = \{8, 12, 13, 14, 18\}$$



$N_{13}(2)$

$$N_{13}(2) = \{3, 7, 8, 9, 11, 12, 13, 14, 15, 17, 18, 19, 23\}$$

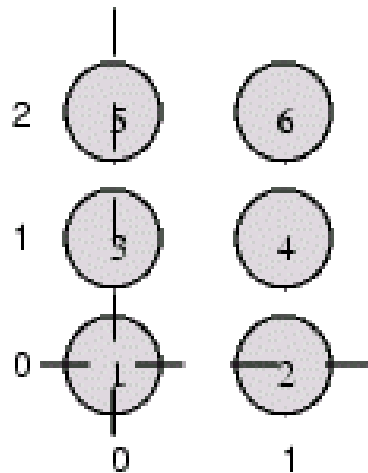
Topologías en MatLab

```
>> pos = gridtop(2,3)
```

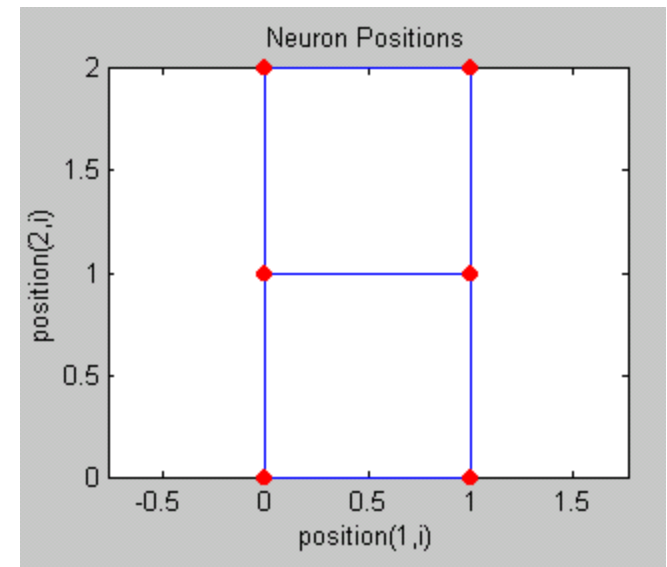
```
pos =
```

0	1	0	1	0	1
0	0	1	1	2	2

```
>> plotsom(pos)
```

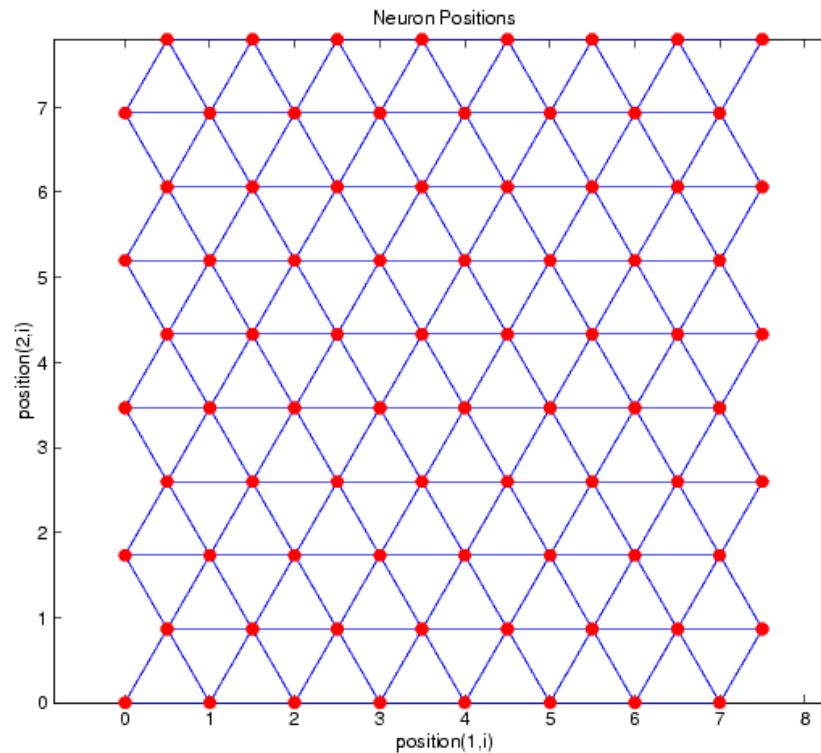


gridtop(2,3)



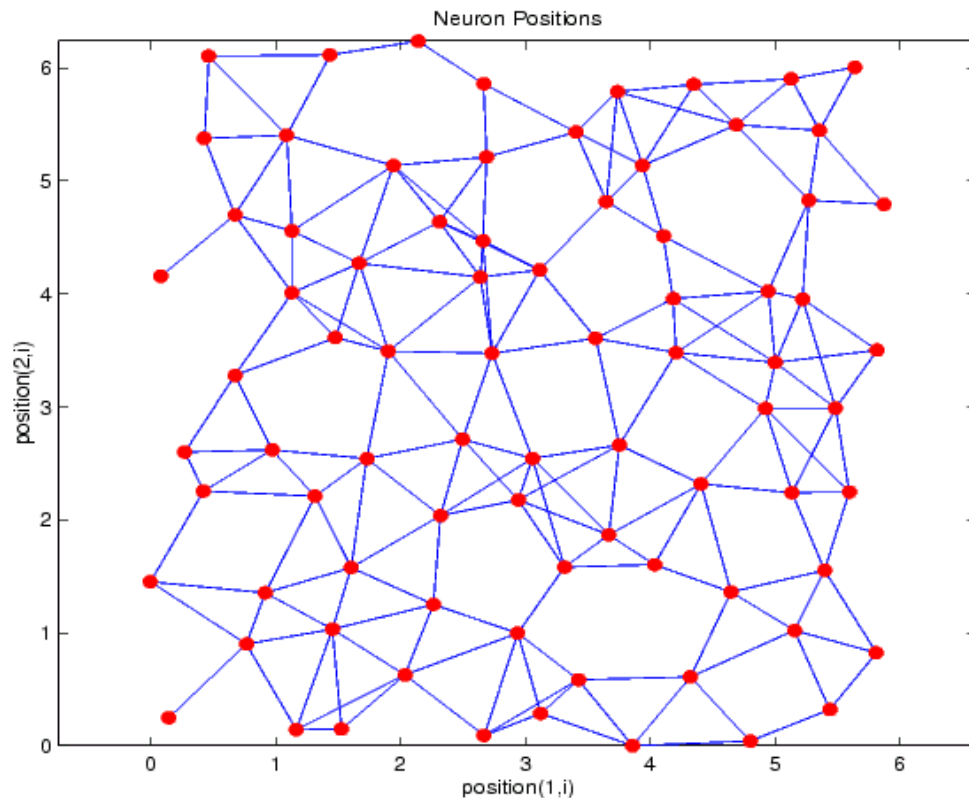
Topologías en MatLab

```
>> pos = hextop(8,10);  
>> plotsom(pos)
```



Topologías en MatLab

```
>> pos = randtop(8,10);  
>> plotsom(pos)
```



Distancias

```
>> pos = gridtop(2,3)
```

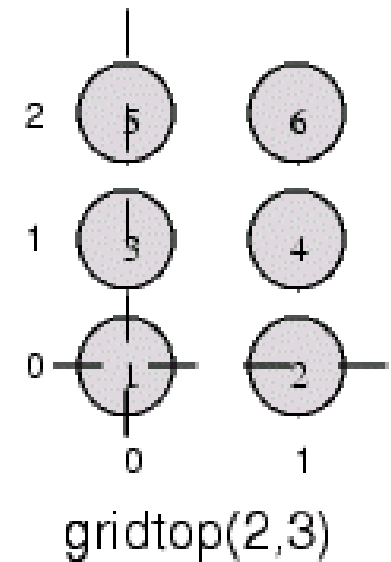
```
pos =
```

0	1	0	1	0	1
0	0	1	1	2	2

```
>> linkdist(pos)
```

```
ans =
```

0	1	1	2	2	3
1	0	2	1	3	2
1	2	0	1	1	2
2	1	1	0	2	1
2	3	1	2	0	1
3	2	2	1	1	0



Ejemplo: Distribuir 1000 puntos en forma random y utilizar una red SOM con 30 neuronas para caracterizarlos.

```
P = rand(2,1000);  
% Disposicion de las neuronas de la capa oculta  
fil_ocultas =6;  
col_ocultas = 5;  
pos = gridtop( col_ocultas, fil_ocultas);  
pasos=linkdist(pos);  
  
[entradas, CantPatrones] = size(P);  
ocultas = fil_ocultas * col_ocultas;  
% Pesos iniciales  
W = 0.5 * ones(entradas,ocultas);  
  
figure(1)  
plot(P(1,:), P(2,:), '+')  
hold on  
plotsom(W', pasos)
```

```
vecinas=3
ITE_MAX = 100;
alfa = 0.25;
ite = 0;
while ( ite <= ITE_MAX ) & "W no cambie mucho" ,

    for patr = 1:CantPatrones,
        %buscar el W mas proximo
        %mayor contiene el número de neurona ganadora

        %Actualizar la neurona ganadora y su vecindad
    end

    if (vecinas>=1) & (mod(ite,20)==0) ,
        vecinas = vecinas-1;
    end

    %redibujar
    ite = ite + 1
end
```

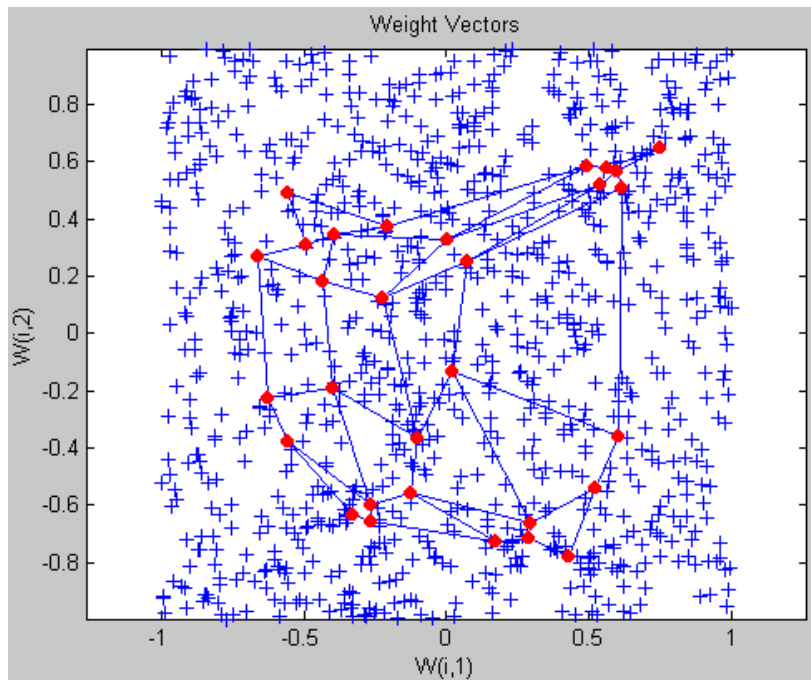
Actualización de la neurona ganadora y sus vecinas

```
for j = 1:ocultas,  
    if pasos(mayor, j) <= vecinas,  
        W(:,j) = W(:,j) + alfa * (P(:,patr) - W(:,j));  
    end  
end
```

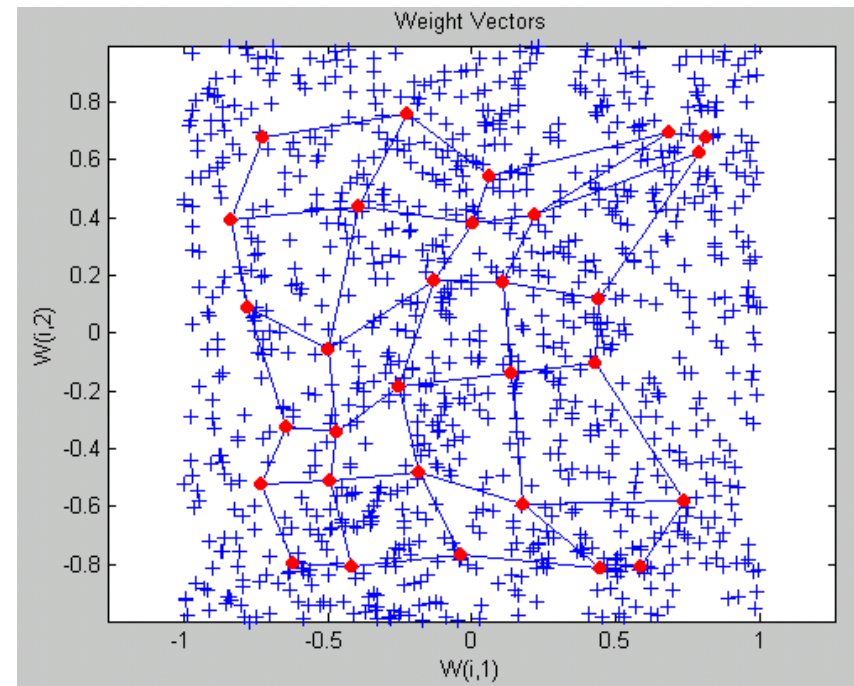
Recuerde que:

```
pos = gridtop( col_ocultas, fil_ocultas);  
pasos=linkdist(pos);
```

Ejemplo: Distribuir 1000 puntos en forma random y utilizar una red SOM con 30 neuronas para caracterizarlos.

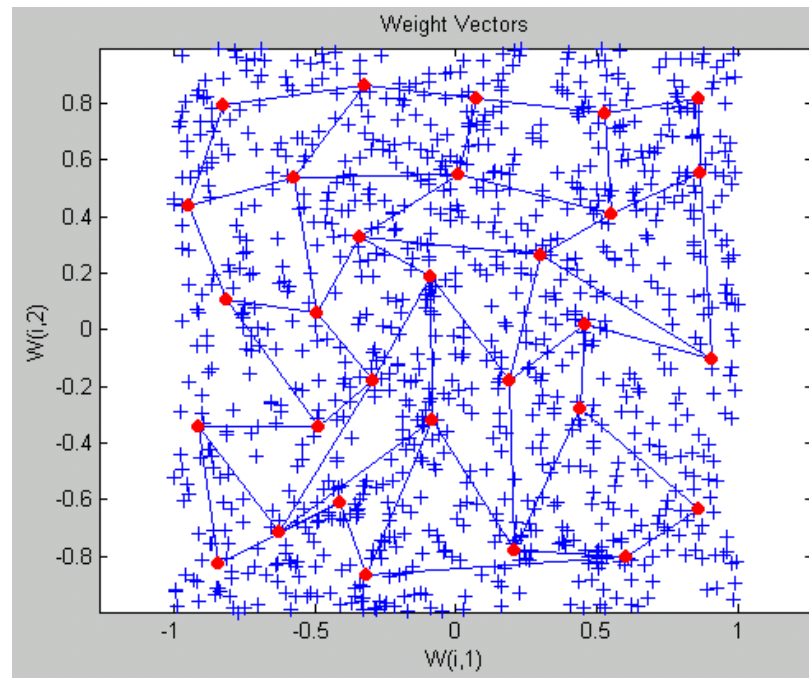


20 épocas
(vecindad = 3)



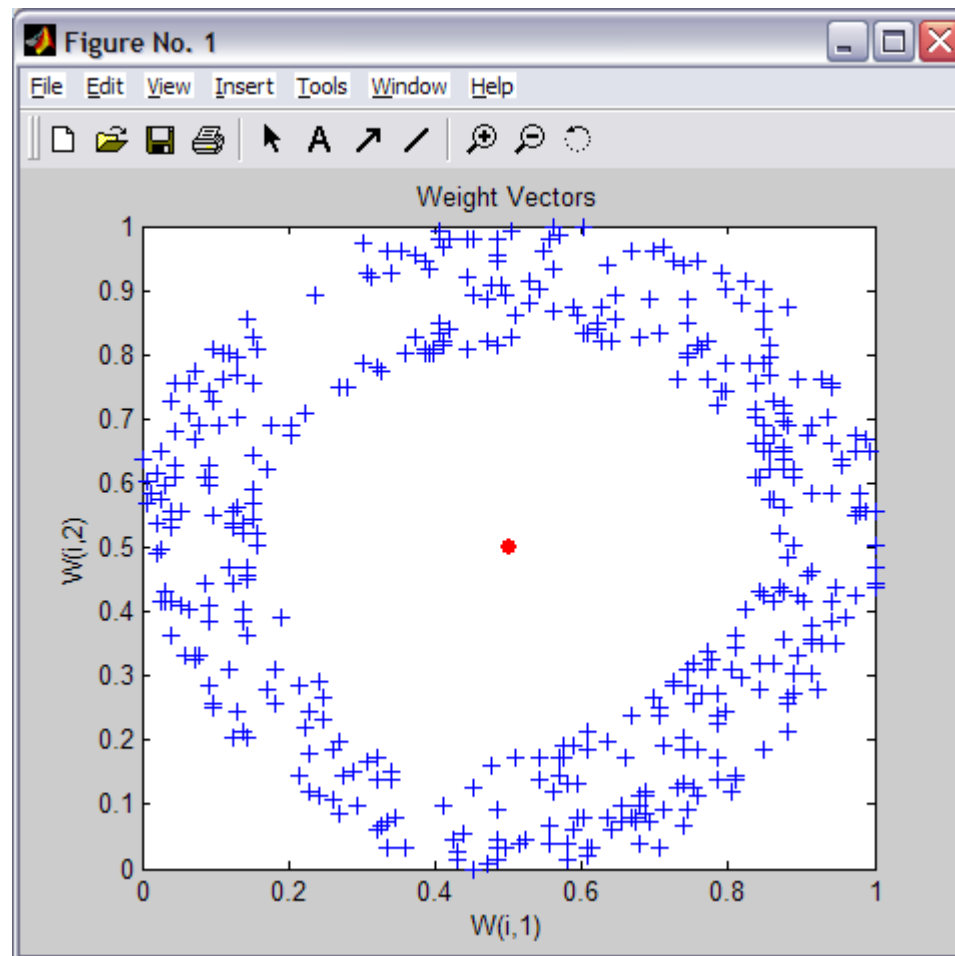
40 épocas
(vecindad = 2)

Ejemplo: Distribuir 1000 puntos en forma random y utilizar una red SOM con 30 neuronas para caracterizarlos.

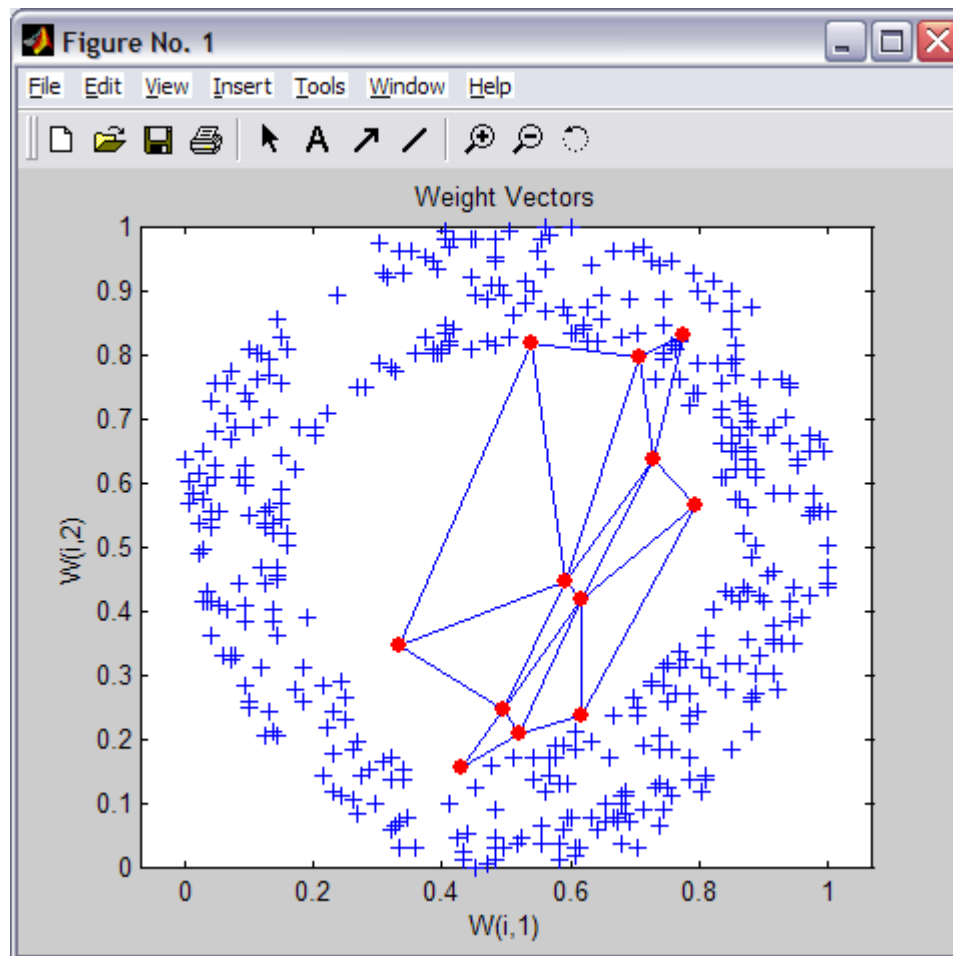


100 épocas
(vecindad = 0)

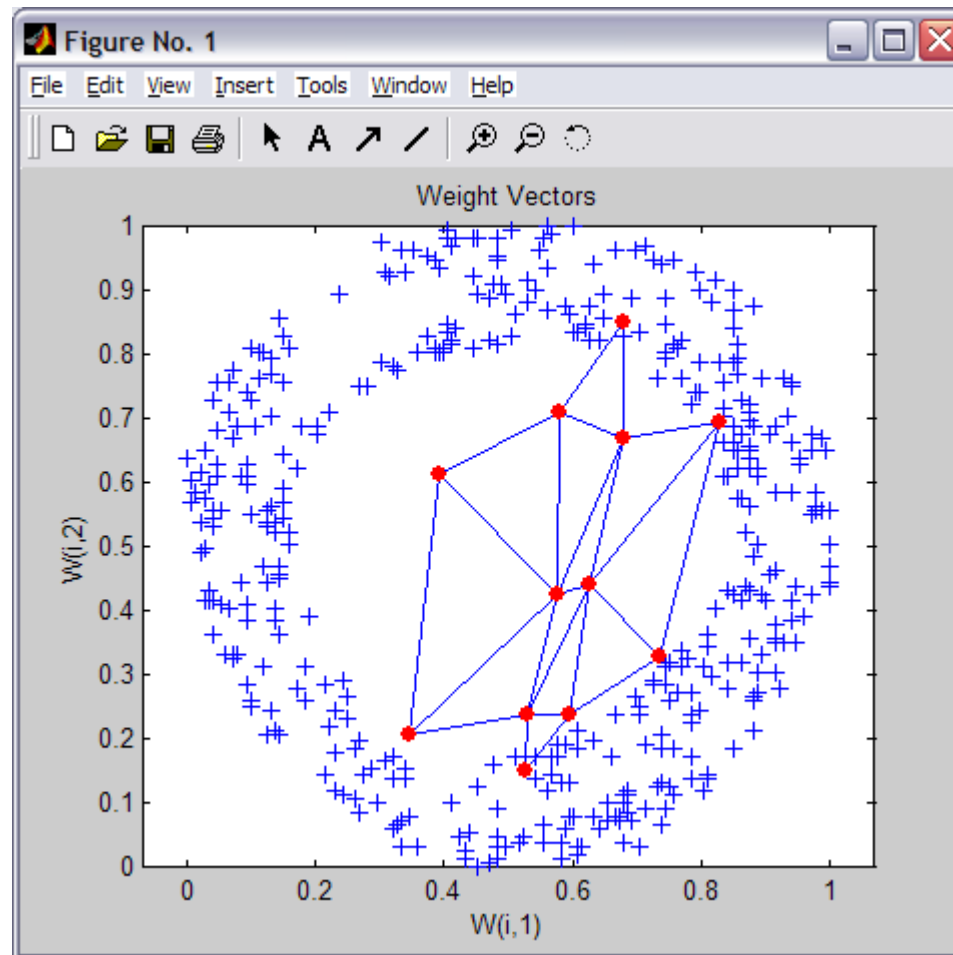
Ejemplo



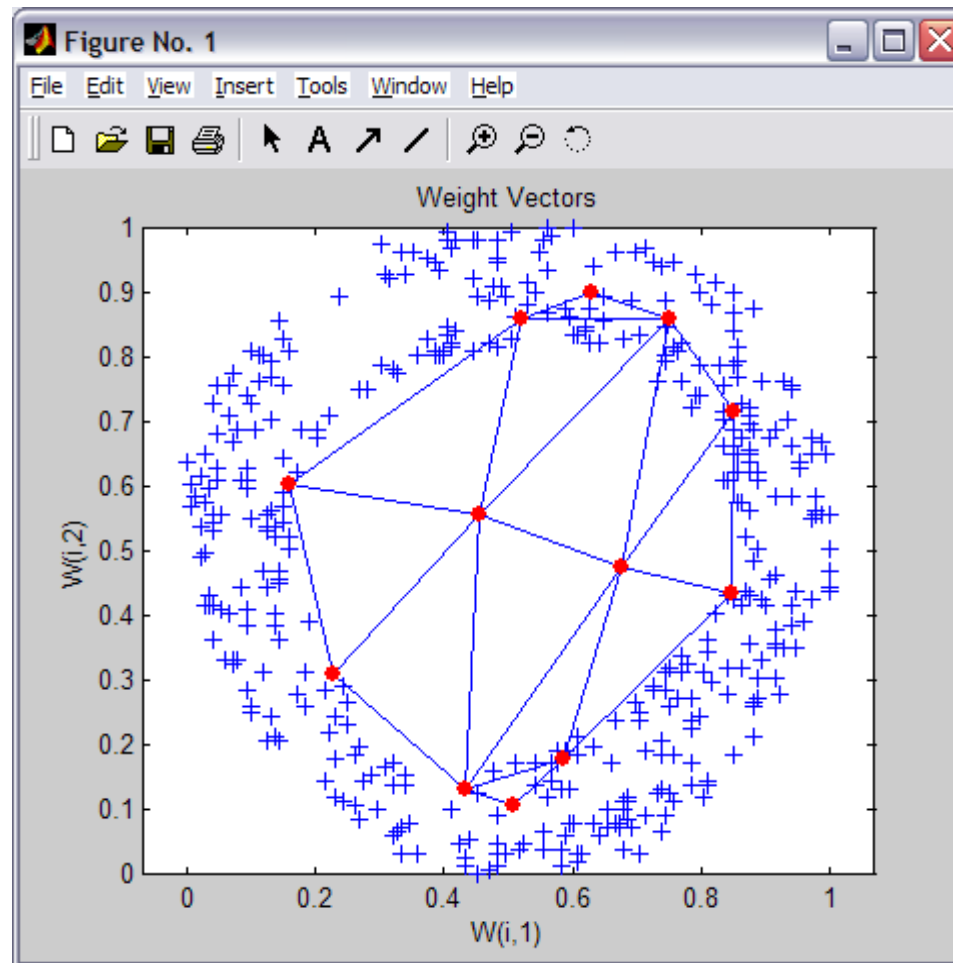
Ejemplo



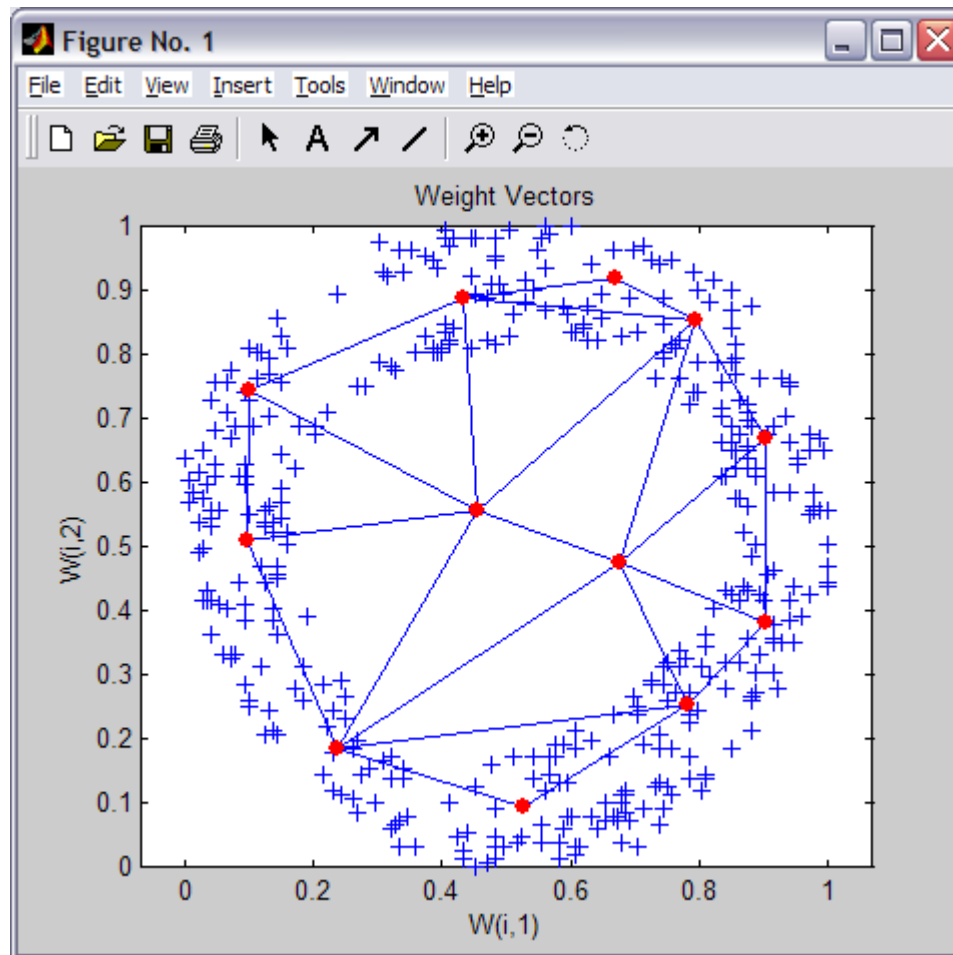
Ejemplo



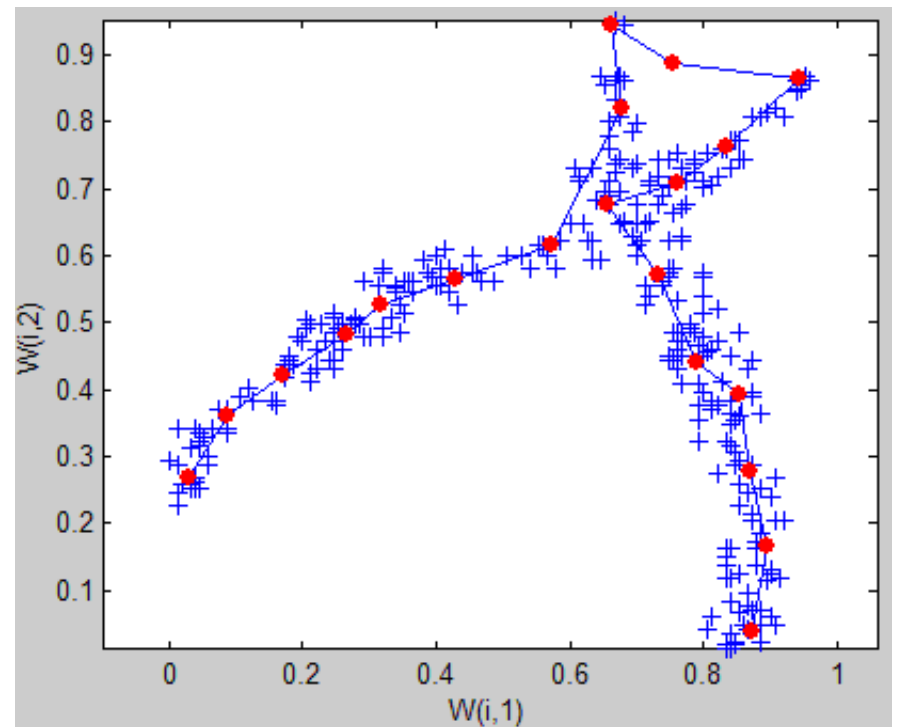
Ejemplo



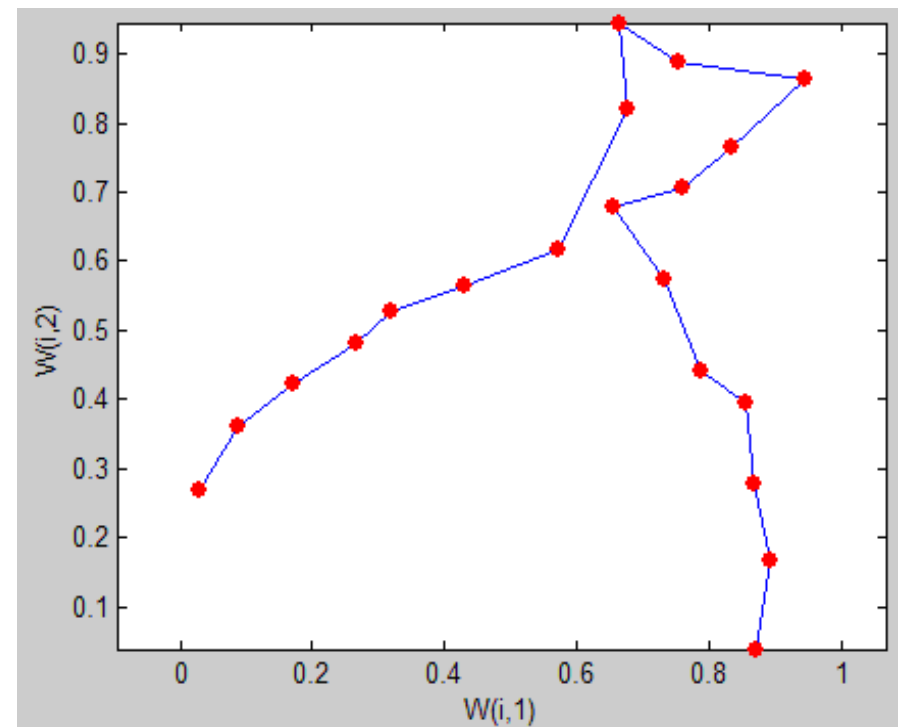
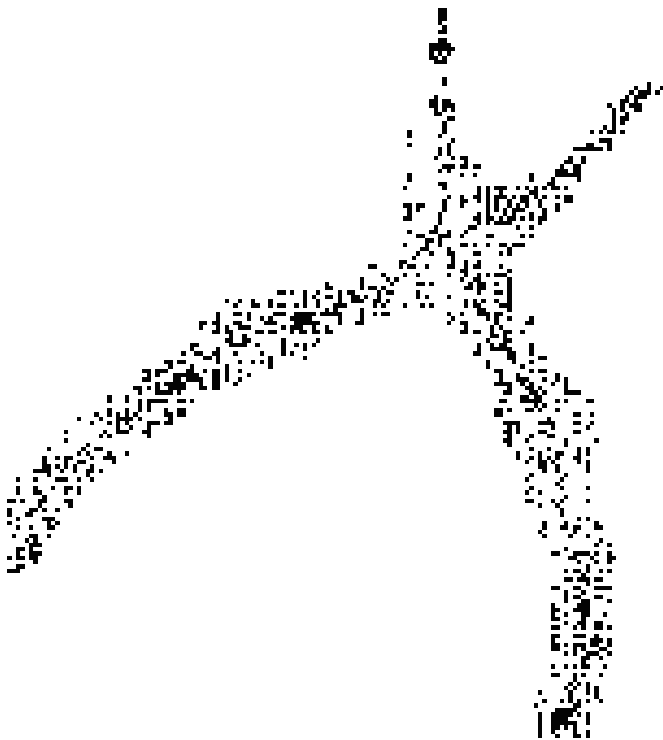
Ejemplo



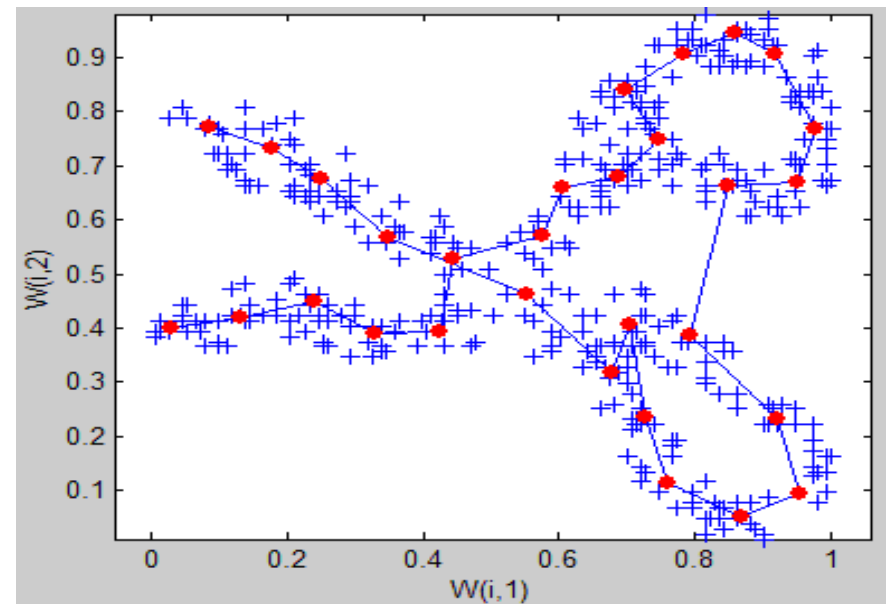
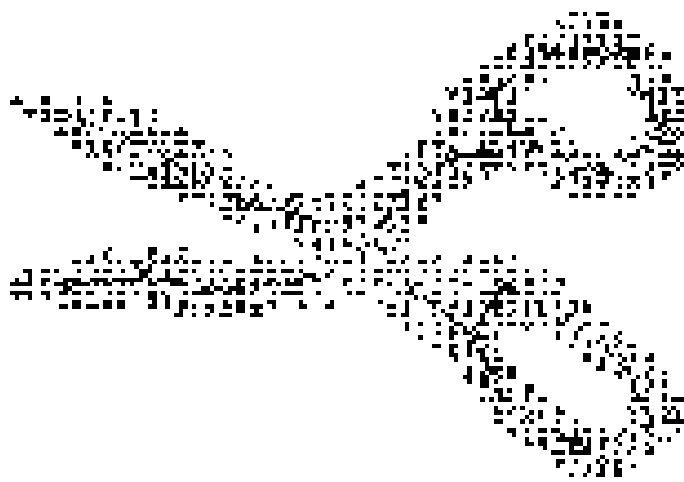
Ejemplo



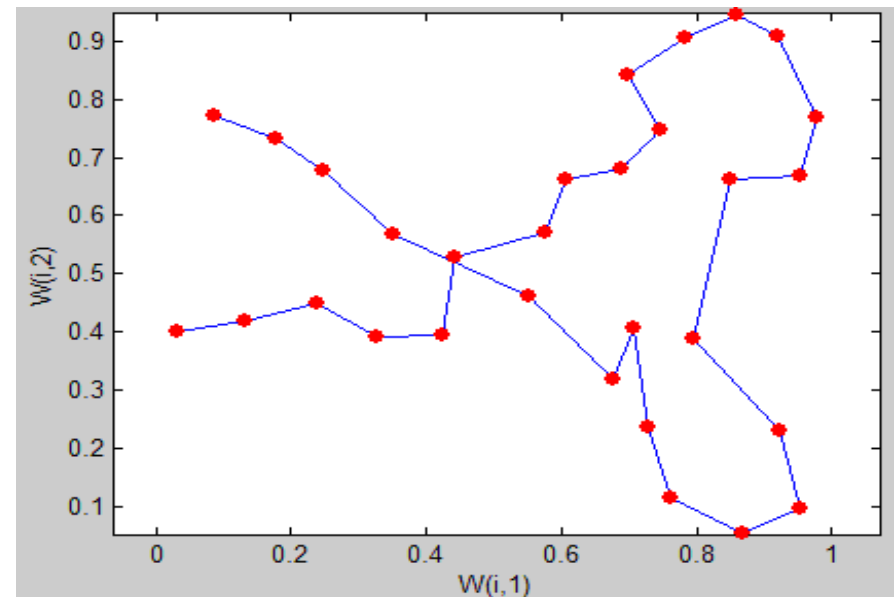
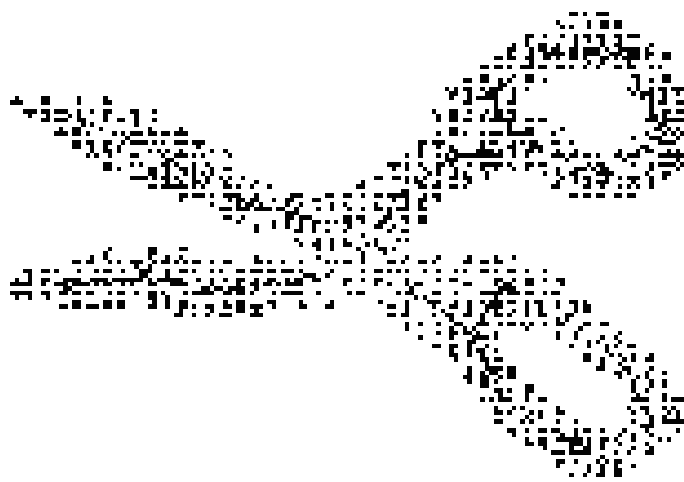
Ejemplo



Ejemplo



Ejemplo



SOM aplicado al reconocimiento de tipos de Vinos

'c3-->2'	'c3-->5'	'c1-->9'	'c1-->9'	'c1-->7'
'c3-->10'	'c3-->5'	'c2-->2'	'c1-->11'	'c1-->5'
'c3-->8'	'c3-->5; c2-->1'	'c2-->1'	'c1-->4'	'c1-->12'
'c3-->3; c2-->1'	'c3-->1; c2-->8'	'c2-->8; c3-->1'	'c2-->5'	'c2-->2; c1-->2'
'c3-->6'	'c2-->7'	'c2-->6'	'c2-->5'	'c2-->4; c3-->1'
'c3-->4'	'c2-->3'	'c2-->5'	'c2-->7'	'c2-->4; c3-->1'

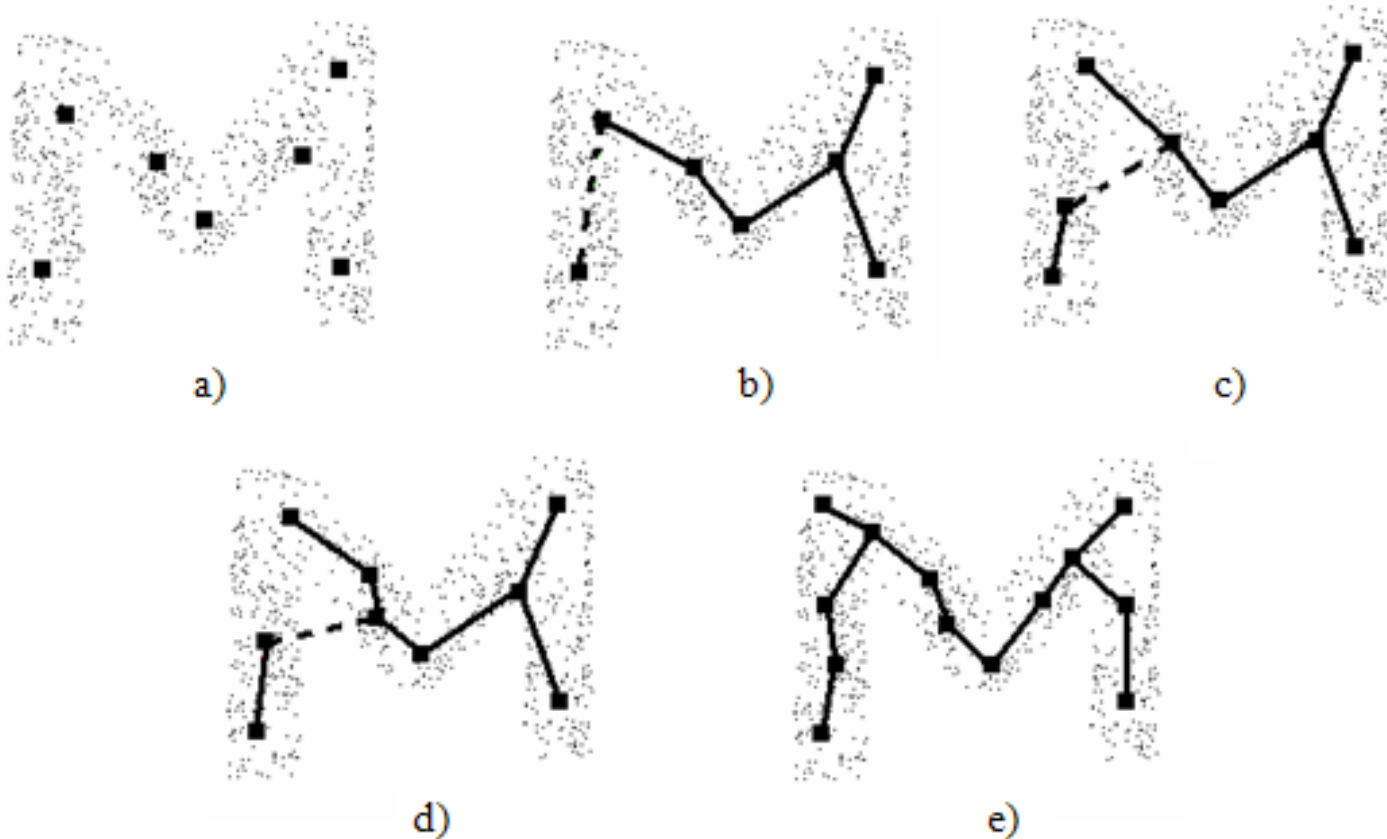
Construcción del mapa

```
% calcular para c/clase la lista de patrones que contiene  
clases = [];  
clases = zeros(ocultas,max(NroClase));  
for i = 1:CantPatrones,  
    %Calcular la neurona ganadora  
    [DMax mayor] = max(-sqrt(  
        sum(((ones(ocultas,1)*P(i,:) - W).^2),2 ) ));  
  
    c = NroClase(i); %venía con los patrones  
    clases(mayor, c) = clases(mayor, c ) + 1;  
end
```

Construcción del mapa

```
mapa = cell(fil_ocultas, col_ocultas);
for nro = 1:ocultas,
    texto = "";
    for cl = min(NroClase):max(NroClase),
        if clases(nro, cl) > 0,
            if length(texto) > 0, texto = strcat(texto, ' ; '); end
            texto = strcat(texto, char(cl), '-->', int2str(clases(nro, cl)));
        end;
    end;
    fila = fil_ocultas - floor( (nro-1)/col_ocultas );
    col = mod( (nro-1), col_ocultas ) + 1;
    mapa(fila, col) = {[texto]};
end;
mapa
```

SOM + MST (minimun spaning tree)



a) Neuronas de la red SOM, b) Conexión usando MST. El arco punteado supera el umbral, c) Resultado del primer reemplazo, d) Esqueleto después del 2do. reemplazo, e) Esqueleto después del 5to. reemplazo.



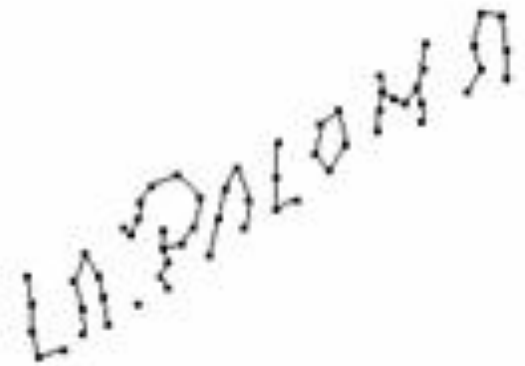
a)



b)



c)



d)

a) Imagen Original, b) Imagen luego de un pobre preprocesamiento, c) Zona de interés ampliada, d) Esqueleto

Mapas Auto-organizativos Dinámicos

♦ Características generales

- La RN es un grafo formado por la interconexión de neuronas competitivas. La conexión es regular y juega un papel importante en la visualización de la reducción del espacio de entrada.
- Cada neurona de la RN representa a un conjunto de patrones. La medida de similitud depende del problema.
- El entrenamiento se realiza de la manera habitual actualizando la información de la ganadora y su entorno. Esto permitirá preservar la topología de los datos de entrada.

Mapas Auto-organizativos Dinámicos

♦ Características generales (cont)

- En cada paso de adaptación, se incrementa el error de la neurona ganadora. Con esto se busca evitar que la misma neurona represente a la mayoría de los patrones. El cálculo del error depende de la aplicación.
- El valor de error almacenado en cada neurona es utilizado para determinar donde deben insertarse nuevas neuronas.

Base de Datos Zoo

(<http://mlearn.ics.uci.edu/MLSummary.html>)

Zoológico

Nombre: oso hormiguero Cantidad de patas: 4

Características: Marque con tilde lo que posee

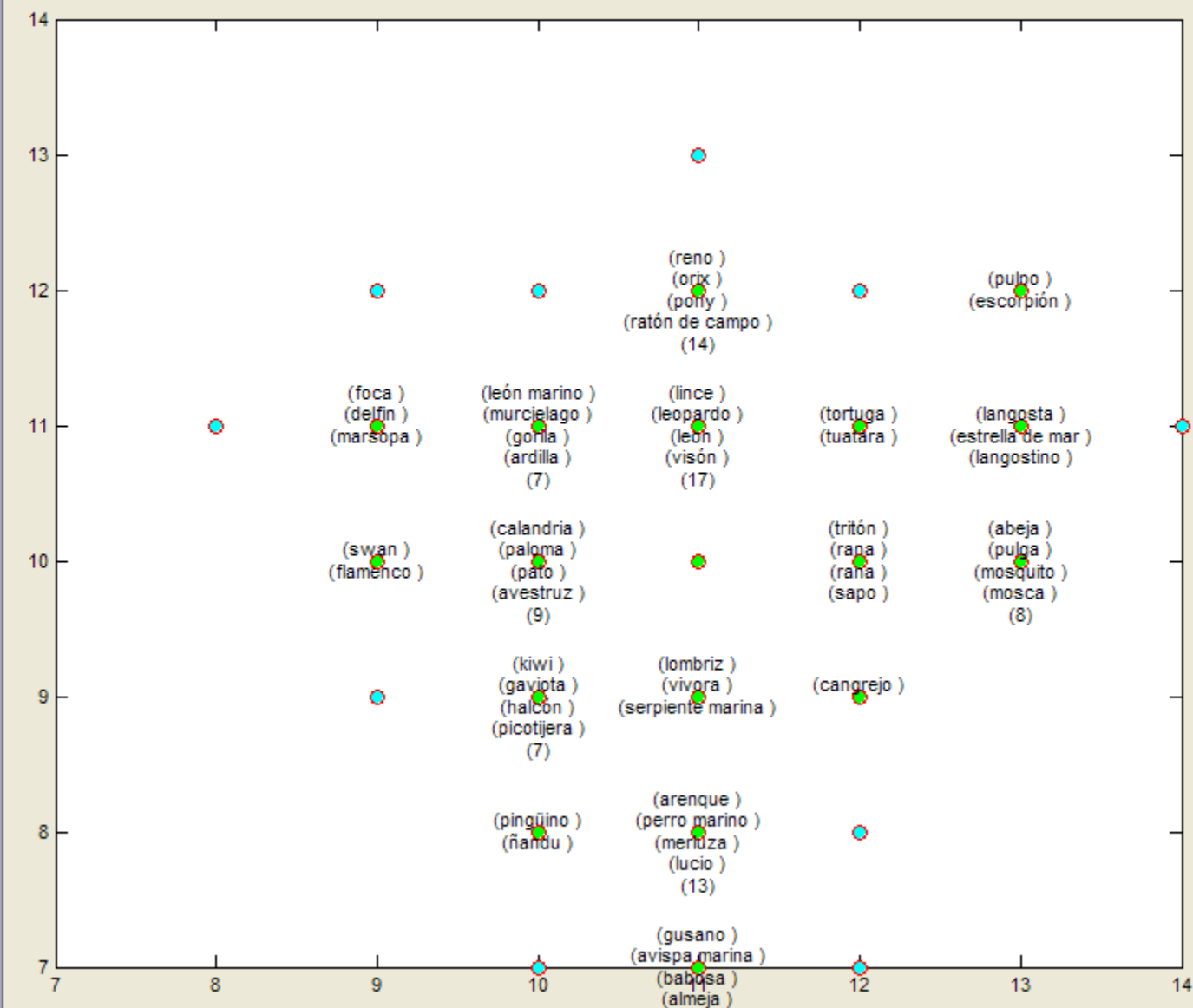
<input checked="" type="checkbox"/> Tiene pelo	<input checked="" type="checkbox"/> Tiene dientes	<input type="checkbox"/> Es acuático
<input type="checkbox"/> Tiene plumas	<input checked="" type="checkbox"/> Es vertebrado	<input checked="" type="checkbox"/> Es depredador
<input type="checkbox"/> Nace de huevo	<input checked="" type="checkbox"/> Respira	<input type="checkbox"/> Tiene cola
<input checked="" type="checkbox"/> Toma leche	<input type="checkbox"/> Es venenoso	<input type="checkbox"/> Es doméstico
<input type="checkbox"/> Vuela	<input type="checkbox"/> Tiene aleta	Total de atributos: 6

Clase a la que pertenece

<input checked="" type="radio"/> Mamífero	<input type="radio"/> Pez	<input type="radio"/> Invertebrado
<input type="radio"/> Ave	<input type="radio"/> Anfibio	
<input type="radio"/> Reptil	<input type="radio"/> Insecto	

Navigation buttons: [Back] [Previous] [Next] [Forward] [Add] [Remove] [Up] [Check] [Cancel]

GSOM 1.0 - Neural Networks - February 2003 - UNLP



Up

< Left Right >

Down

Zoom in Zoom out

Grow Threshold: 5.89175

Spread Factor 0.5

1- Initialize

2- Start Training

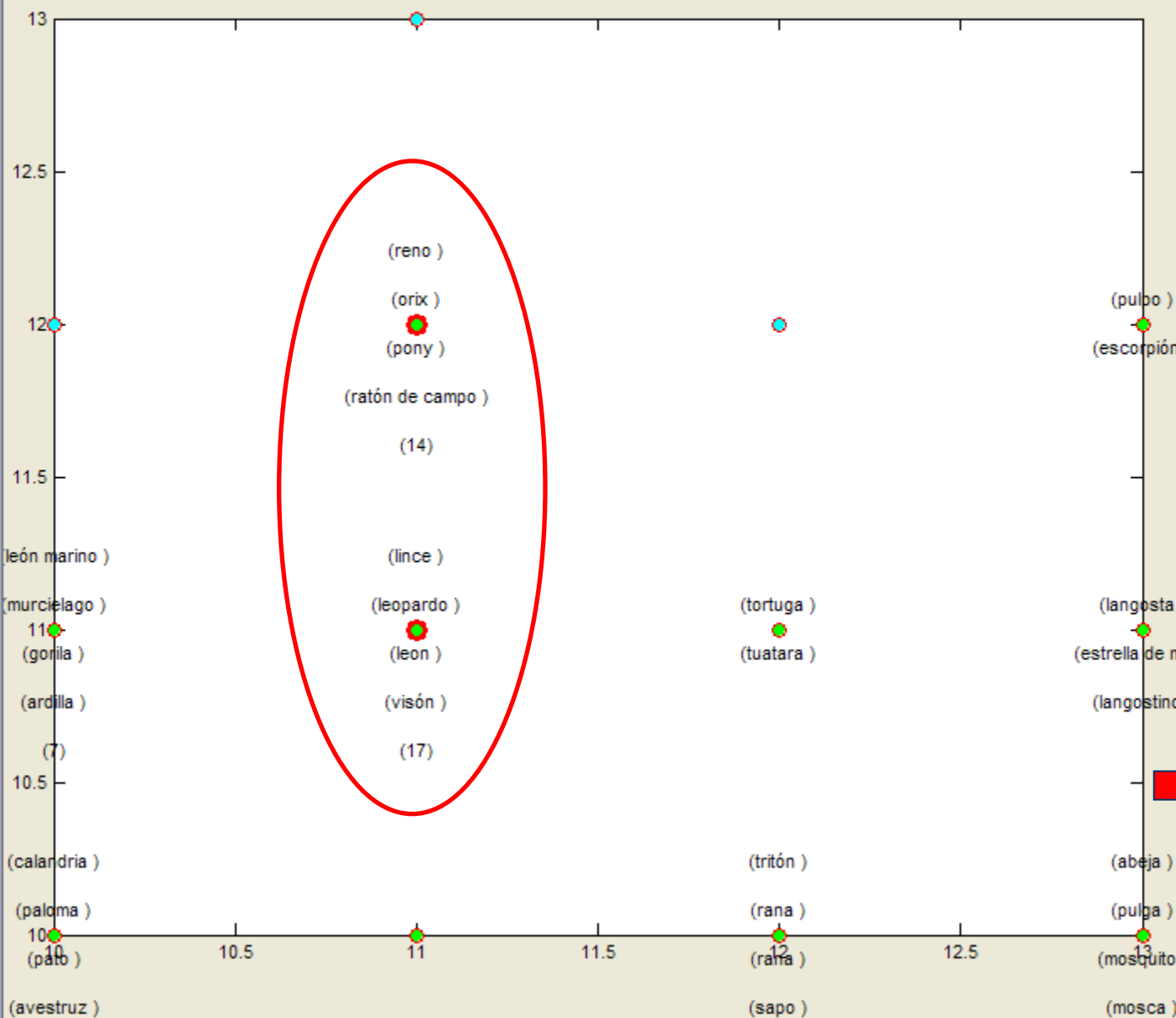
3- Stop Training

4- Spread out

Configure Net Status

Close About

GSOM 1.0 - Neural Networks - February 2003 - UNLP



Up

< Left Right >

Down

Zoom in Zoom out

Grow Treshold: 5.89175

Spread Factor 0.5

1- Initialize

2- Start Training

3- Stop Training

4- Spread out

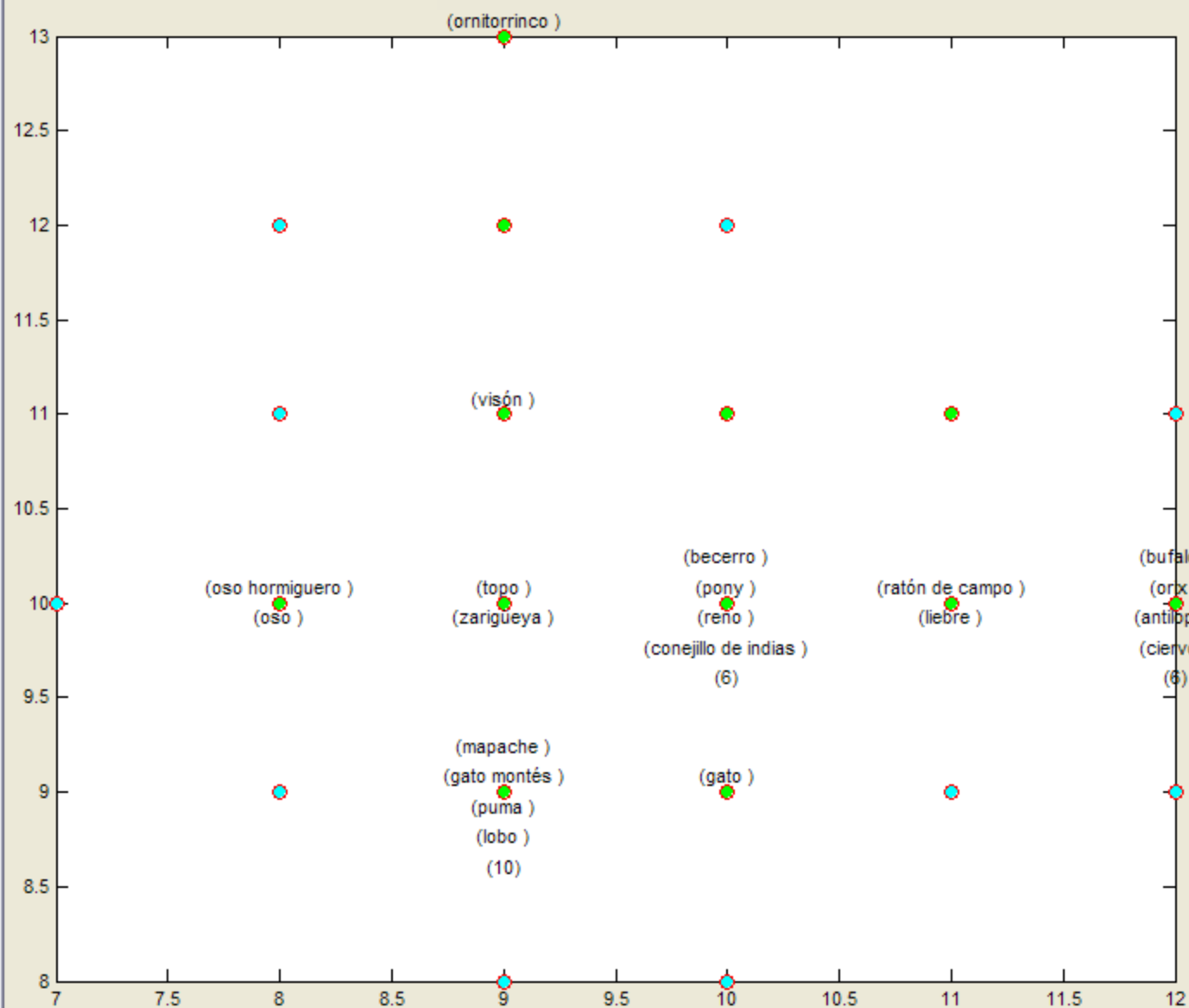
Configure Net Status

Close About



File Edit View Insert Tools Window Help

GSOM 1.0 - Neural Networks - February 2003 - UNLP



Up

< Left

Right >

Down

Zoom in

Zoom out

Grow Treshold:

3.34086

Spread Factor

0.675

1- Initialize

2- Start Training

3- Stop Training

4- Spread out

Configure

Net Status

Close

About