

Sistemas de Apoio à Decisão— Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005



SOM
Kohonen's Self-Organising Maps
Victor Lobo

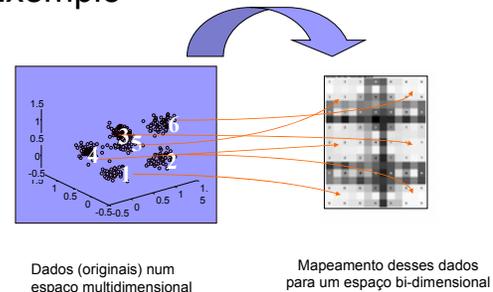
Sumário

- Para que serve ?
 - Visualizar dados multidimensionais
 - Detectar agrupamentos (clusters)
 - Fazer segmentação de mercado
 - Detectar outliers
 - Resolver o TSP, controlar Robots, Alarmes, etc,etc,etc
- O que é um SOM ?
 - Perspectiva histórica
 - Princípios básicos
 - A matemática
- Como posso usá-lo
 - SOM-PAK
- Aplicações de SOM

O que é um SOM ?

- SOM = Self Organized Map
 - Mapa auto-organizado
 - Mapas de KOHONEN, ou Redes Neurais de KOHONEN
- Rede neuronal para aprendizagem não supervisionada
 - Visualização de dados multidimensionais, projecção de dados sobre um espaço de dimensão mais baixa, clustering, detecção de novidades

Exemplo



Introdução

- Inspiração
 - Códigos para quantização de vectores / Memórias associativas
 - Preservar a topologia nos mapeamentos: padrões vizinhos devem ser mapeados para neurónios vizinhos
- Tuevo Kohonen
 - 1970s - Memórias associativas
 - 1982 - Primeiros artigos sobre SOM
 - 1988 - Livro sobre SOM, artigos sobre SOM no IEEE
 - 1995,1997,2001 – Livro “Self Organizing Maps”

Para que é que o posso usar ?

- “Visualização e análise de dados de dimensionalidade elevada” Kohonen
 - Projecções de espaços de dimensão N para espaços de dimensão $M < N$
 - Normalmente N para 2 or 1
- Algoritmo de Clustering (detecção de agrupamentos)
 - Utilizado em conjunto com “U-Matrices”
- Algoritmo de Classificação (com LVQ ...)
- Amostragem, extracção de características, data mining, detecção de novidade, etc

Sistemas de Apoio à Decisão— Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005

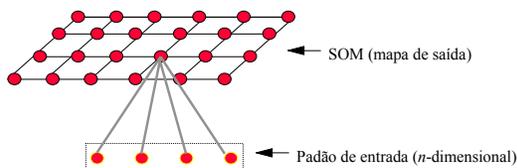
Principais referências

- “Self-Organizing Maps”, Prof.Tuevo Kohonen
 - Springer Verlag 2001
- “www.cis.hut.fi/projects/somtoolbox/links”
 - Public-domain software
 - SOM-PAK para DOS:
(www.cis.hut.fi/research/som_lvq_pak.shtml)
 - SOM Toolbox para MATLAB
 - Manuais, guias, e documentação
 - Bibliografia extensa
 - “www.cis.hut.fi/research/som-bibl”
 - 5384 referências em Abril de 2005

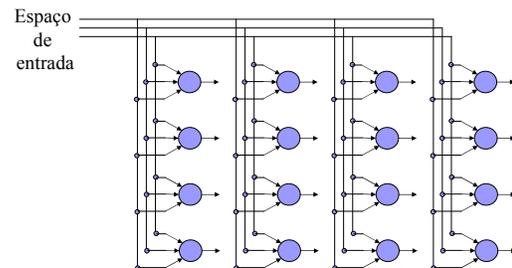
Princípios básicos

SOM básico

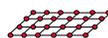
- Rede neuronal como uma única camada
- Aprendizagem competitiva (quase “winner-take all”)
- Neurónios dispostos numa grelha N-dimensional
 - Mapas bi-dimensionais são os mais comuns



SOM básico (outra visão...)

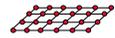


SOM básico



- Cada neurónio é um **vector no espaço de entrada**, tal como os padrões de dados.
- Durante o treino, os neurónios são **puxados** para as posições dos dados de entrada, **arrastando** consigo os seus vizinhos no espaço de saída
- O mapa pode ser visto como uma **superfície de borracha** que é esticada e torcida de modo a passar pelos padrões de dados (ou pelo menos a ficar perto)

SOM básico



- Padrões de entrada são comparados com todos os neurónios, e o mais próximo é considerado o **neurónio vencedor**.
- Consideramos que o padrão de entrada é **mapeado** para o neurónio vencedor.
- O **vencedor actualiza-se** (de modo a aproximar-se mais do padrão de dados que representa), e os seus vizinhos actualizam-se também um pouco
- Há sempre uma ligeira diferença entre os dados e os neurónios que os representam. Essa diferença é o **erro de quantização**.

Sistemas de Apoio à Decisão– Redes de Kohonen (SOM)

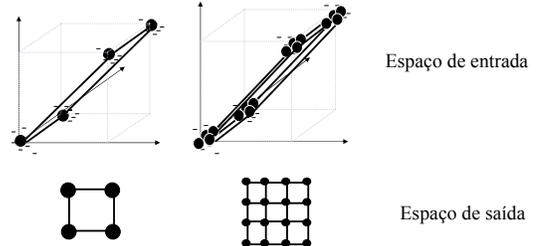
V 1.0, V.Lobo, EN/ISEGI, 2005

Comparações com biologia

- Sistemas biológicos têm que usar algum tipo de auto-organização e adaptação
- Há evidência de:
 - Uma estrutura de camadas no cérebro
 - Essas camadas aparentam organizar espacialmente a informação
 - “Conceitos” similares são mapeados para áreas adjacentes
 - Trabalho experimental com visão em animais sugere uma organização similar ao SOM no córtex

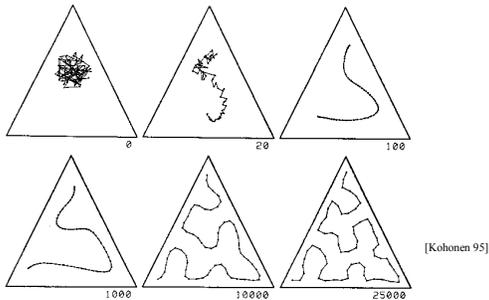
Exemplo 1: mapeamento de 3D para 2D

- Pontos agrupados em 4 cantos de um cubo



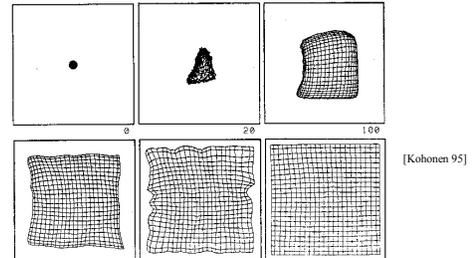
Exemplo 2: mapeamento de 2D para 1D

- Dados distribuídos uniformemente num triângulo

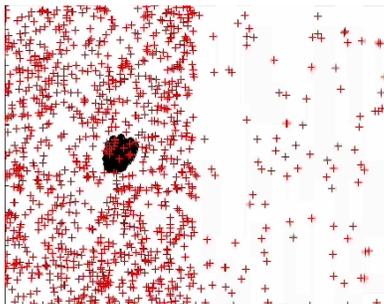


Exemplo 3: mapeamento de 2D para 2D

- Dados distribuídos uniformemente num quadrado
- Usado na demo do Matlab



Treino de um mapa...



Outro exemplo...

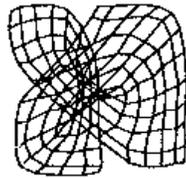
- Mapa a 1 dimensão
- Mapa a 2 dimensões
- Mapa de cores:
<http://www.eas.asu.edu/~eee511/demos/som-color/sofm.html>

Sistemas de Apoio à Decisão— Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005

Possíveis falhas

- Como podemos saber se o SOM treinou correctamente ?
- Problemas de desdobraimento
 - Existem mínimos locais
- Overfitting
 - Por vezes bom (!)
 - Por vezes mau



[Ritter 91]

Detecção de agrupamentos (clusters)

- Ao olhar para o espaço de saída, como podemos detectar agrupamentos?
- U-Matrix [Ultsch 93]
 - Calcular a distância, no espaço de entrada, de vizinhos no espaço de saída
 - Valores baixos \Rightarrow Neurónios próximos \Rightarrow cluster
 - Valores altos \Rightarrow Neurónios longe \Rightarrow Espaço vazio

Ideal



U-mat real



A matemática

Algoritmo de treino

- Para cada padrão de entrada:
 - 1) **Calcular** a distância entre o padrão de dados e todos os neurónios:
 $(d_j = \|x_k - w_j\|)$
 - 2) **Escolher** o neurónio vencedor
 $w_{winner} (w_j : d_{ij} = \min(d_{jn}))$
 - 3) **Actualizar** cada neurónio de acordo com a regra
 $w_j = w_j + \alpha h(w_{winner}, w_j) \|x_k - w_j\|$
 - 4) **Repetir** o processo até que um critério de paragem seja atingido.

Função de distância

- Normalmente é a euclidiana
- Outras medidas de distância (ou similitude)
 - Outras medidas de Minkowski (blocos de cidade ou Manhattan, de 3ª ordem, etc)
 - Produtos internos
 - Hamming
 - Coeficientes de Tanimoto
 - Ângulo entre vectores
 - Distâncias de Hausdorff

Função de actualização

- $w_j = w_j + \alpha h(w_{winner}, w_j) \|x_k - w_j\|$
 - α : Ritmo de aprendizagem
 - Controla a plasticidade
 - Deve tender para 0 para garantir espabilidade
 - Na realidade é $\alpha(t)$
 - h : Função de vizinhança
 - Controla a interacção lateral entre neurónios vizinhos
 - Depende da distância ao vencedor (no ESPAÇO DE SAÍDA)
 - É normalmente uma função radial monótona decrescente

Sistemas de Apoio à Decisão— Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005

Função de vizinhança

■ A forma

- Gaussiana
- Rectangular (bolha)
- Rampa
- Outras

$$h_g(w_j, w_{min}) = e^{-\frac{1}{2} \left(\frac{\sqrt{(i-n)^2 + (j-m)^2}}{r} \right)^2}$$


$$h_r(w_j, w_{min}) = \begin{cases} 1 & \text{if } \sqrt{(i-n)^2 + (j-m)^2} \leq r \\ 0 & \text{if } \sqrt{(i-n)^2 + (j-m)^2} > r \end{cases}$$


■ O raio

- Função to tempo $r(t)$
 - Raio grande \Rightarrow Muitos neurónios actualizados \Rightarrow Permite desdobração
 - Raio pequeno \Rightarrow Apenas os vizinhos mais próximos são actualizados \Rightarrow Ajuste fino

Fundamentos teóricos

■ Função de energia a minimizar: [Hertz 91]

$$V(w) = \frac{1}{2} \sum_x \sum_i \Lambda(i, i^*) |\bar{x} - \bar{w}_i|^2 = \frac{1}{2} \sum_x \sum_k M_k \sum_i \sum_j \Lambda(i, k) (x_j - w_{ij})^2$$

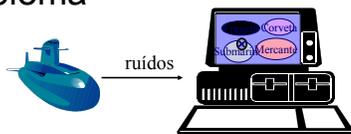
- Altamente não-linear devido ao conceito de neurónio vencedor
- Resultados mais completos para dimensão 1 [Cottrell]
- Algumas boas aproximações para 2D [Ritter]
- Densidade de neurónios \propto (densidade dos dados)^k, com $k < 1$
- Existe um factor de "ampliação" das zonas com menor densidade

Como escolher os parâmetros

- Tamanho ?
- Ritmo de aprendizagem inicial ?
- Raio inicial ?
- Número de iterações ?

Uma aplicação de SOM

O problema

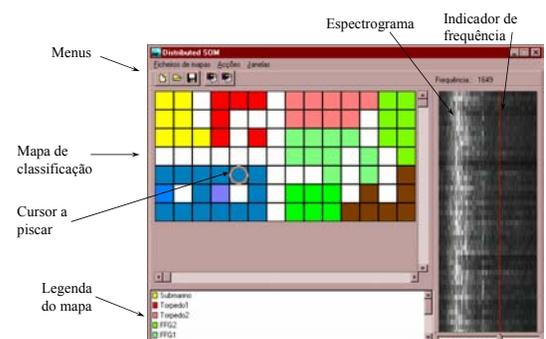


- Classificação automática de sons submarinos
- Sistema de baixo custo usando PC com placa de som

Objectivos adicionais:

- ↳ Análise exploratória dos dados / Data mining
- ↳ Detecção de novidades/semelhanças
- ↳ Ferramenta de pesquisa

Sistema produzido



Sistemas de Apoio à Decisão– Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005

Características

- Treino e classificação paralelo
 - Usa PVM, fácil de instalar em redes de PC
 - Corre em qualquer rede com máquinas em MS-Windows ou Linux
- SOM Binário
 - Usa distância de hamming distance e uma regra de actualização binária
- Fácil usar outras medidas de distância
- Menús amigáveis
- Pre-processamento e classificação em tempo real

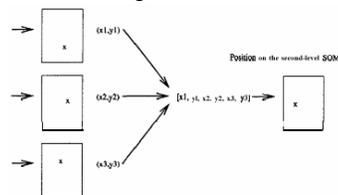


Outros trabalhos com SOM

SOM multicamada e Redução da dimensionalidade

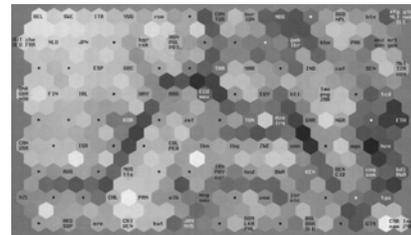
- Análise de relatórios Finaceiros [Kiviluoto 98]
 - Coordenadas do neurónio vencedor são os dados para a camada seguinte

- Por vezes todos os dados do nerónio vencedor são usados



Som Para clustering

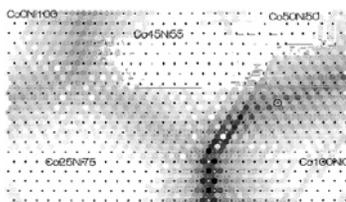
- Pobreza no mundo [Kohonen 95]
 - Agrupa países de acordo com uma série de indicadores económicos



[Kohonen 95]

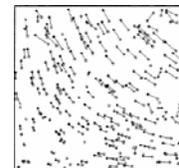
Som para clustering/classificação

- Análise química (dados de espectroscopia) [Tokutaka 98]



SOM para seguimento

- Análise do movimento de fluidos [Labonté 98]
 - Segue a posição de partículas num fluido
 - Analisa a trajetória dos neurónios durante a aprendizagem



Sistemas de Apoio à Decisão— Redes de Kohonen (SOM)

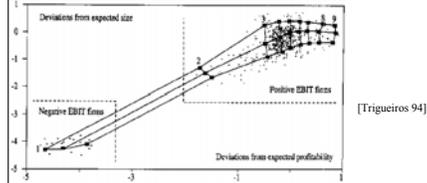
V 1.0, V.Lobo, EN/ISEGI, 2005

SOM para amostragem

- Avaliação de empresas (contabilidade)

[Trigueiros 94]

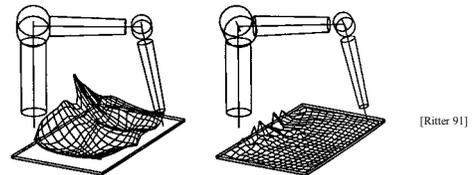
- Mapeamento de 2D para 2D
- Usado para seleccionar situações representativas do estado de empresas



SOM em robótica

- Controlo de um braço robot [Ritter 91]

- SOM contém os sinais de controlo necessários para atingir uma dada localização



SOM para monitorização

- Trajectórias em SOM

- Shu-Ching Kuo 2004

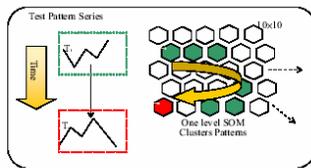


Figure 3. Trajectory analysis concept

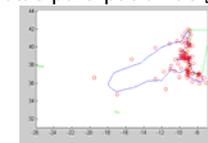
SOM para o problema do caixeiro viajante

- Problema exacto

- Passar exactamente por n locais, fazer o planeamento de C.I. [Hueter 88][Choy 95]

- Problema relaxado

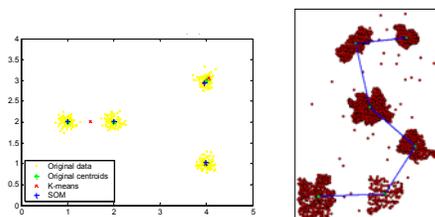
- Determinação de rotas para patrulhas [Lobo 2005]



SOM para k-médias

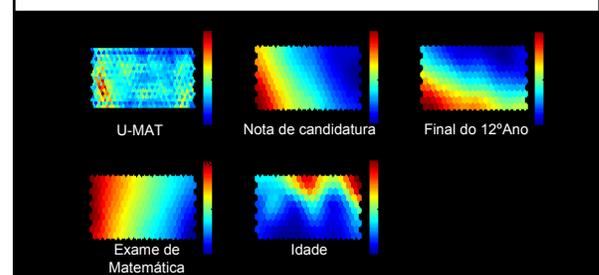
- SOM 1-dimensional, com k -neurónios

- Serve de inicialização robusta [Bação 05]



SOMs e planos de componentes

- Permitem ver as diferentes variáveis



Sistemas de Apoio à Decisão— Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005

Outras aplicações de SOM

- Previsão
 - Consumos energéticos [Osowski 98]
- CIM
 - Agrupamento de ferramentas [Guerrero 98]
- Monitorização de processos (análise de trajectórias no espaço de saída), do estado de condição de máquinas, reconhecimento de voz, análise de imagem, estudos musicais, desengo de circuitos, análise geopolítica, linguística, pesquisa na Web, Economia, biologia, química, ... etc, etc, etc, etc, etc, etc...

Como posso usar ?

Software disponível

- SOM-PAK
 - Código C, compilável em UNIX ou MS DOS
 - Rápido e fiável, fácil de utilizar
- MATLAB
 - Somtoolbox (www.cis.hut.fi/projects/somtoolbox)
- Muitos others
 - SAS Enterprise Miner (!), Clementine, etc...

© DSOM: Our very own software ! ☺

SOM-PAK 3.1

■ Colecção de programas que recebem parâmetros através do comando-linha:

- randinit- Inicializa um SOM
- vsom - Treina um SOM
- qerror - Erro de quantização
- visual - Calcular o neurónio vencedor para cada dado
- vcal - Calibrar um SOM (dar nomes aos neurónios)
- umat - Calcular a U-Matrix de um SOM
- plane - Visualizar uma das dimensões de um SOM

Formato dos dados

- Texto simples
 - Primeira linha tem informação de controlo
 - Nº de atributos de cada padrão
 - Outros: tamanho da rede, função de vizinhança, etc
 - Linhas com dados
 - Valores dos atributos, opcionalmente com nomes
 - Comentários
 - #

```
3
#Isto é um comentário
10 10 5 label_1
3 10 4
7 8 2 label_2
```

Sessão típica

1) Inicialização do SOM

- randinit : Inicializa com valores aleatórios

2) Treino do SOM

- vsom : Implementa o algoritmo de treino
- Usado 2 vezes:
 - Primeiro com vizinhanças e ritmos de aprendizagem grandes, para permitir o **desdobramento**
 - Depois, com pequenas vizinhanças e ritmos de aprendizagem para ajustes finos

Sistemas de Apoio à Decisão– Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005

Sessão típica

3) Medição do erro de quantização (opcional)

- `qerror` : O mapa está a representar bem os dados ?

4) Visualização

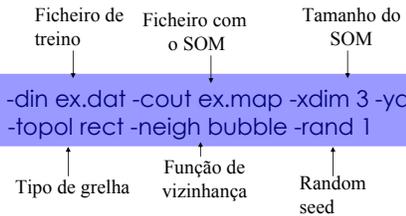
- `vcal` : Põe nomes (ou classes, ou labels) nos neurónios (só para problemas supervisionados)
- `visual`: Encontra o vencedor para cada padrão
 - Usar outro programa para visualizar de facto...
- `umat` : desenha a U-Matrix em ps
- `plane` : visualiza um dos planos

Inicialização do mapa

■ `randinit`

- Usa um ficheiro de dados como protótipo

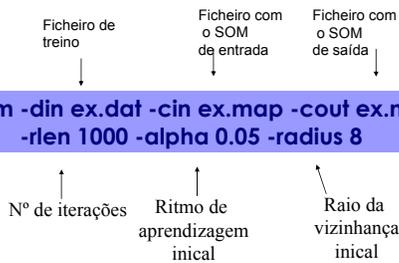
```
randinit -din ex.dat -cout ex.map -xdim 3 -ydim 4
-topol rect -neigh bubble -rand 1
```



Treino do SOM

■ `Vsom` – Correr duas vezes !

```
vsom -din ex.dat -cin ex.map -cout ex.map
-rlen 1000 -alpha 0.05 -radius 8
```



Visualização dos resultados

■ `U-Mat`

```
umat -cin ex.map -ps 1 > output.ps
```

■ `visual`

```
visual -din ex.dat -cin ex.map -dout out.txt
```

Exemplo prático

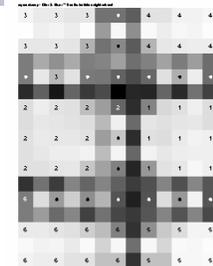
- Vamos mapear um conjunto de pontos de um espaço 3 dimensional, que estão em vértices de um cubo

```

      3 5      6
    4      1  2
  
```

Exemplo prático

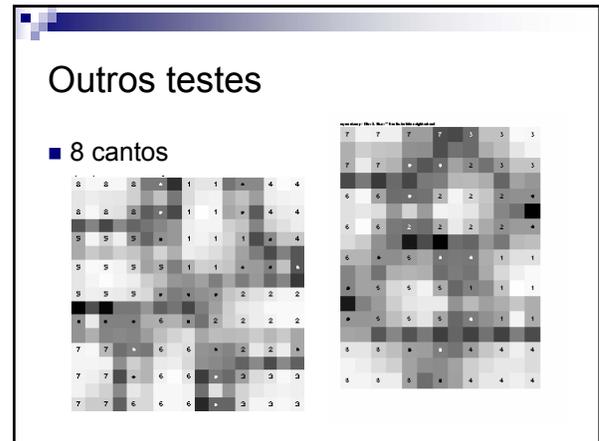
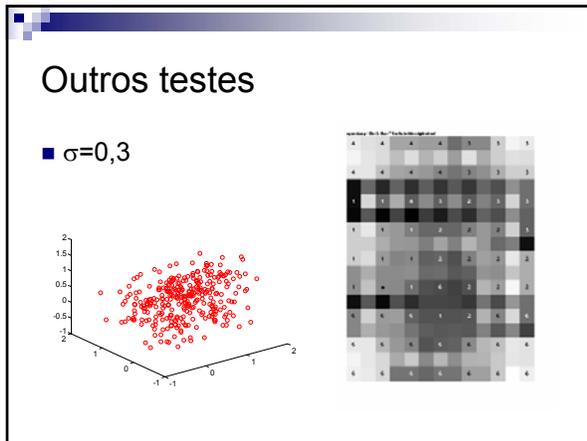
- 306 pontos centrados em 6 vértices com $\sigma=0.1$
 - Áreas escuras indicam fronteiras entre clusters



```
randinit -xdim 7 -ydim 9 -din square.dat -cout square.map -topol rect -neigh bubble
vsom -din square.dat -cin square.map -cout square.map -rlen 1000 -alpha 0.1 -radius 7
vsom -din square.dat -cin square.map -cout square.map -rlen 10000 -alpha 0.02 -radius 3
vcal -din square.dat -cin square.map -cout squarel.map
umat -cin squarel.map -ps 1 > squarel.ps
```

Sistemas de Apoio à Decisão— Redes de Kohonen (SOM)

V 1.0, V.Lobo, EN/ISEGI, 2005



Testes com o SOMToolbox

- Excelentes gráficos
- Fácil de alterar

```

load concehos2
sD=som_data_struct(data);
sM = som_randini_geo(sD,'msize',[5 10],'vect','sheet');
sM.neigh = 'bubble';

iterations_1 = 1000; iterations_2 = 2000;
radius_in_1 = 5; radius_in_2 = 2;
alpha_in_1 = 0.7; alpha_in_2 = 0.1;

% Standard SOM
sM1 = som_soptain(sM,sD,...
    'radius_in',radius_in_1,...
    'radius_fin',delta,...
    'alpha_in',alpha_in_1,...
    'trainlen', iterations_1, 'epochs');
sM1 = som_soptain(sM1,sD,...
    'radius_in',radius_in_2,...
    'radius_fin',delta,...
    'alpha_in',alpha_in_2,...
    'trainlen', iterations_2, 'epochs');
save concehos_som sM1
[get_fer]([som_qualify(sM1,sD);
som_show(sM1,'umaf','off','nocolor','Standard SOM')
sU=som_umaf(sM1);
    
```

