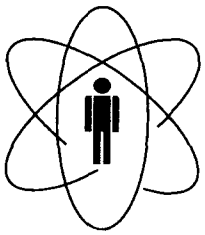
ISSN 0101-9201



CBPF - CENTRO BRASILEIRO DE PESQUISAS FÍSICAS
Rio de Janeiro

Notas Técnicas

CBPF-NT-001/10
março 2010

Introdução ao Toolbox de Redes Neurais de Kohonen

Elisângela Lopes de Faria, Marcelo Portes Albuquerque, Jorge Luiz Gonzalez Alfonso,
Márcio Portes Albuquerque & Jose Thadeu Pinto Cavalcante

Ministério da
Ciência e Tecnologia
BRASIL UM PAÍS DE TODOS
GOVERNO FEDERAL

Introdução ao Toolbox de Redes Neurais de Kohonen

Elisângela Lopes de Faria [(a)]

Marcelo Portes Albuquerque [(a)]

Jorge Luiz Gonzalez Alfonso [(b)]

Márcio Portes Albuquerque [(a)]

Jose Thadeu Pinto Cavalcante [(a)]

Centro Brasileiro de Pesquisas Físicas [(a)]

Departamento de Física [(b)]
Universidade Federal do Espírito Santo

Sumário

Resumo

Os Mapas de Kohonen são tipos de redes neurais que utilizam a técnica de aprendizado competitivo para a análise de agrupamentos e nos últimos anos vêm sendo utilizados em uma grande diversidade de aplicações. Neste sentido este trabalho visa uma breve introdução sobre os mesmos, bem como uma descrição técnica mais detalhada de como utilizar o Toolbox do Mapa Organizável de Kohonen (Som Toolbox) desenvolvido no ambiente Matlab. Diferentes funções do Som Toolbox foram abordadas ao longo do trabalho e um estudo de caso com dados correspondentes a doença da tireóide foram utilizados para facilitar o entendimento dessas funções. Por fim, foi possível constatar que a ferramenta utilizada apresenta ótimos recursos para análise de agrupamentos por meio da formação de grupos, visualização e métricas existentes no toolbox.

Abstract

The Kohonen maps are types of neural networks that using competitive learning technique for cluster analysis and in have been used recent studies in a wide range of applications. Thus, this work aims at a brief introduction to them, as well as a detailed technical description of how to use the Toolbox Organizing Map of Kohonen (SOM Toolbox) developed in Matlab. Different functions of the SOM Toolbox were addressed throughout the document and a case study with data for thyroid disease was used to facilitate the understanding of these functions. Finally, it was established that the tool used has great features for cluster analysis by means of clustering, visualization and metrics exist in the toolbox.

1. Introdução

O Mapa Auto Organizável de Kohonen (SOM) é um método de rede neural com aprendizado não supervisionado e competitivo [1], capaz de mapear um conjunto de dados de entrada em um arranjo de neurônios geralmente de uma ou duas dimensões. Por se tratar de uma rede não supervisionada ela se torna uma poderosa ferramenta quando se está trabalhando com estudos envolvendo agrupamentos ou clusterização, principalmente quando não se conhece a quantidade de clusters a priori. [1] destaca também o uso da SOM em problemas de engenharia, medicina, etc., além das potencialidades de visualização de dados multivariados, análise de agrupamentos, mineração de dados, descoberta de conhecimento e compressão de dados.

Geralmente quando se está trabalhando com rede neural artificial, seja qual for o modelo escolhido, uma série de parâmetros precisa ser testada, pois eles influenciam diretamente no resultado da rede. Para os mapas auto-organizáveis de Kohonen acontece o mesmo, ou seja, vários parâmetros tais como tamanho do mapa, tamanho do raio de vizinhança, taxa de aprendizagem e outros, precisam ser testados até que resultados satisfatórios sejam alcançados. Neste caso o uso de uma ferramenta computacional eficiente pode ajudar na obtenção dos resultados almejados.

O Som Toolbox assim como o Som_Pak são ferramentas computacionais muito conhecidas desenvolvidas na Universidade da Finlândia pela equipe de pesquisas em Mapas Auto-Organizáveis. Os dois pacotes implementam a rede SOM, porém o Som Toolbox [2] foi desenvolvido em Matlab [3] que é uma ferramenta computacional destinada a fazer cálculos com matrizes e possui uma grande vantagem que são os toolboxes (funções implementadas internamente aplicadas a áreas específicas como as redes neurais artificiais), o que torna possível a otimização do tempo. Neste caso o Som Toolbox funciona como mais uma biblioteca como as inúmeras já existentes no Matlab, com várias funções já implementadas que facilita o estudo envolvendo as redes Kohonen. Já o Som_Pak foi desenvolvido em C [4] e contou com a participação direta de Teuvo Kohonen, pesquisador que propôs o modelo dos Mapas Auto Organizáveis.

As duas ferramentas citadas acima são gratuitas, possuem código fonte aberto e tem sido muito utilizadas em diversos projetos, uma vez que possuem características como confiabilidade, disponibilidade de código fonte e funcionalidade. Outra grande vantagem do Som Toolbox é que mesmo sendo criado para a área de datamning ele pode ser utilizado em outras áreas, inclusive juntamente com novas funções criadas pelo

próprio usuário. A desvantagem do Som toolbox é que ele não possui suporte, o que pode dificultar um pouco o seu uso.

Esta nota técnica tem como objetivo fazer uma breve introdução à rede Kohonen (algoritmo de aprendizado, criação do mapa e outros). E tem como foco principal a introdução ao Som Toolbox, permitindo assim que os usuários que estão iniciando tanto no ramo de data mining como em outras aplicações dos Mapas Auto Organizáveis de Kohonen, tenham uma base para iniciar seus estudos.

2. Mapas Auto Organizáveis de Kohonen

Os Mapas Organizáveis de Kohonen ou simplesmente redes kohonen tem como principal característica a formação de um mapa topográfico dos padrões de entrada no qual as localizações espaciais (ex: coordenadas) dos neurônios na grade são indicativas das características estatísticas intrínsecas contidas nos padrões de entrada. Sua topologia possui duas camadas, na qual todas as unidades de entrada encontram-se conectadas a todas as unidades de saída através de conexões sinápticas e esta camada de saída é organizada na forma de um grid (em geral de duas dimensões).

Conforme citado anteriormente a rede de Kohonen utiliza regras de aprendizado competitivo. Neste sentido os neurônios de uma camada competem entre si pelo privilégio de permanecerem ativos, de tal forma que o neurônio com maior atividade, ou seja, o neurônio vencedor (BMU) seja o principal participante do processo de aprendizado (Winner-takes-all).

O algoritmo responsável pela formação do mapa consiste de quatro etapas, sendo a primeira a inicialização do mapa, seguido de três processos essenciais que são o processo competitivo, o processo cooperativo e a adaptação sináptica. Uma breve descrição dos mesmos pode ser visualizada abaixo, porém mais detalhes podem ser encontrados em [5]:

- ✓ Competição: para cada padrão de entrada, os neurônios da grade calculam seus respectivos valores de uma função discriminante. Esta função discriminante fornece a base para a competição entre os neurônios. O neurônio particular com o maior valor da função discriminante é declarado vencedor da competição.
- ✓ Cooperação: o neurônio vencedor determina a localização espacial de uma vizinhança topológica de neurônios excitados, fornecendo assim a base para a cooperação entre os neurônios vizinhos.
- ✓ Adaptação sináptica: permite que os neurônios excitados aumentem seus valores individuais da função discriminante em relação ao padrão de entrada através de ajustes adequados aplicados a seus pesos sinápticos. Este processo pode ser decomposto em duas fases, ou seja, uma fase de ordenação seguida por uma fase de convergência. A idéia é fazer uma ordenação topológica dos vetores de pesos na primeira fase e na segunda fase realizar uma sintonia fina do mapa de

característica produzindo assim uma quantização estatística acurada do espaço de entrada.

As etapas citadas acima formam a base para o algoritmo SOM e um resumo do mesmo pode ser visualizado nos passos abaixo:

Passo 1: Inicialização da taxa de aprendizagem, raio topológico, função de vizinhança, número de iterações e da matriz de pesos sinápticos com valores aleatórios (entre 0 e 1).

Passo 2: Calcular as distâncias dos dados com a matriz de pesos de modo a encontrar o neurônio vencedor *i*(x) no passo de tempo *n* usando o critério da mínima distancia euclidiana:

$$i(x)_j = \arg\min_j \left\| x(n) - w_j \right\|, \; j = 1,2,...,l$$

Onde x(n) é a entrada ao neurônio *j* no instante *n e l* é o número de neurônios na grade.

Passo 3: Atualizar os vetores de pesos sinápticos de todos os neurônios usando a seguinte equação de atualização:

$$w_j(n+1) = w_j(n) + \eta(n) h_{j,i(x)}(n)(x(n) - w_j(n))$$

Onde $\eta(n)$ é o parâmetro da taxa de aprendizagem e $h_{j,i(x)}(n)$ é a função de vizinhança centrada em torno do neurônio vencedor *i*(x). Vale ressaltar que para ocorrer a convergência do mapa é necessário reduzir o grau de vizinhança relativo ao neurônio vencedor (BMU) ao longo do treinamento.

Passo 4: Retorne ao passo 2 e repita o processo até que um critério de parada seja alcançado.

Durante o processo iterativo do treinamento e devido à atualização da vizinhança, os vetores de pesos sinápticos tendem a seguir a distribuição dos vetores de entrada, conduzindo a uma ordenação topológica do mapa. Assim neurônios adjacentes tenderão a ter vetores de pesos sinápticos similares, ocorrendo assim os agrupamentos.

3. Som Toolbox

3.1 Instalação

O download do Som Toolbox pode ser feito a partir do seguinte endereço: http://www.cis.hut.fi/projects/somtoolbox/. O arquivo precisa ser descompactado em uma pasta e para que o toolbox possa ser utilizado no ambiente Matlab 5.1 ou superior é necessário incluir esta pasta contendo os arquivos na lista de diretórios do Matlab. Para isto é necessário clicar em File, Set Path e em add folder. O próximo passo então é adicionar a nova pasta na qual foi descompactado o toolbox (add folder) e por fim clicar em Save. A tela que aparecerá no workspace do Matlab durante o processo pode visualizada na figura abaixo.

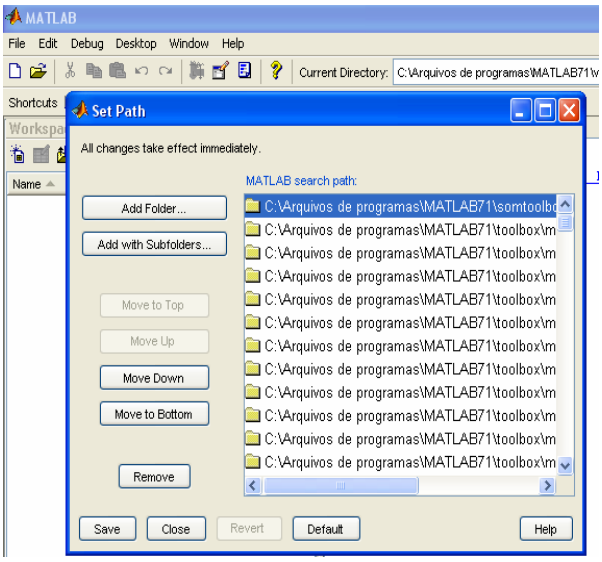
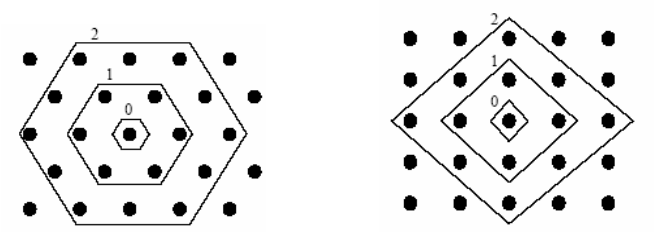


Figura 3.1 Tela para adicionar nova pasta de arquivos.

3.2 Topologia e Tipos de Treinamento

Os neurônios do SOM são organizados em uma grade de 1,2 ou mais dimensões. Cada neurônio da *grade* é representado por um vetor de pesos d-dimensional $m=[m_1, m_2, ..., m_d]$, onde d é igual a dimensão do vetor de entrada. Os neurônios são conectados entre si por uma relação de vizinhança determinados pela topologia ou estrutura do mapa. A topologia no toolbox é dividida em dois fatores: a estrutura local lattice (formato de vizinhança) que pode ser hexagonal ou retangular (figura 3.2) e a forma global do mapa que podem ser visualizados na figura 3.3.



(a) (b)

Figura 3.2 Exemplos de Vizinhança discreta: (a) Estrutura Hexagonal e (b) Estrutura Retangular. Fonte: [2]

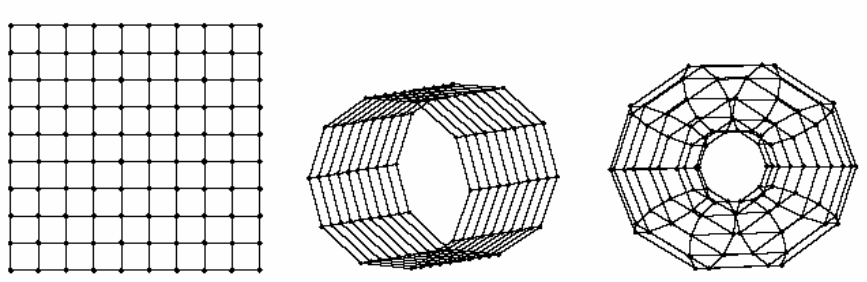


Figura 3.3 Topologia dos Mapas Auto-Organizáveis: Fonte [2]

O algoritmo de treinamento no Toolbox pode ser feito de modo seqüencial (incremental) ou em batch (lote). Para o modo incremental a atualização dos pesos sinápticos dos neurônios no arranjo é feita toda vez que um item de dados é apresentado à rede, ou seja, toda vez que uma amostra do vetor de entrada (x) é apresentada à rede, a distância entre ele e todos os vetores pesos do SOM são calculados usando a medida da

distância euclidiana. O neurônio vencedor (chamado BMU) será aquele que tiver a menor distância entre o próprio e a entrada (x) conforme figura 3.4. Depois de encontrado o BMU os vetores de peso do SOM são atualizados fazendo com que eles se aproximam cada vez mais da entrada (x). Para os vizinhos mais próximos topologicamente o mesmo procedimento é feito. O treinamento é geralmente feito em duas fases. A primeira fase começa com a taxa de aprendizagem e raio de vizinhança bem grande. Na segunda fase tanto a taxa de aprendizagem quanto o raio de vizinhança são pequenas.

Já no treinamento em lote os pesos sinápticos são atualizados apenas após a apresentação de todos os elementos do conjunto de dados utilizados. Na maioria das vezes este algoritmo é significativamente mais rápido que o anterior e mais detalhes podem ser encontrados em [2].

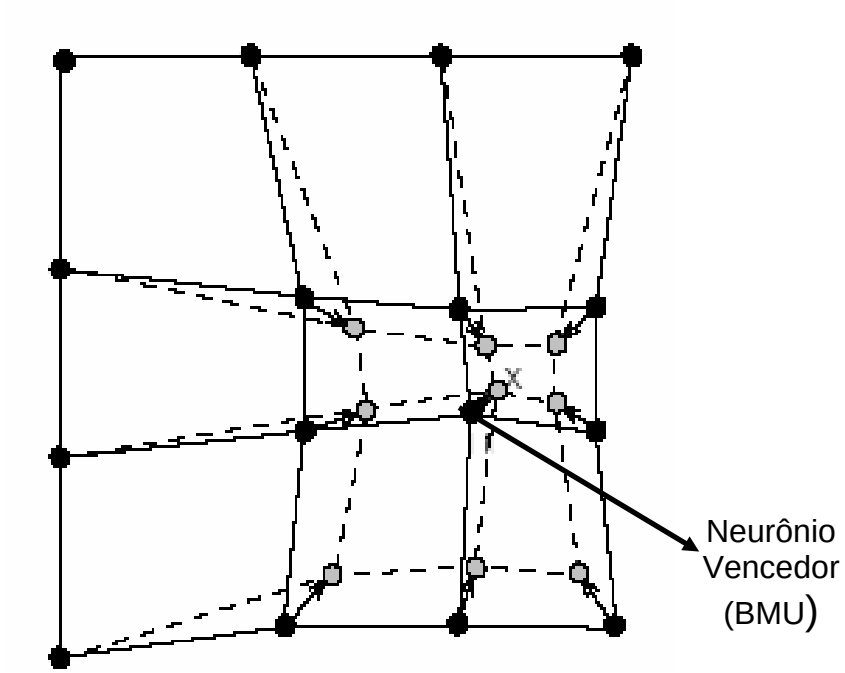


Figura 3.4 Atualização dos pesos do neurônio vencedor (BMU) e de seus vizinhos mais próximos em direção à entrada x. Os círculos pretos e cinzas correspondem às situações antes e depois da atualização dos neurônios respectivamente. (Adaptado de Fonte: [2])

3.3 Base de Dados

Os tipos de dados manuseados pelo Toolbox são do tipo de tabela, onde cada linha da tabela representa uma amostra dos dados e cada coluna representa uma determinada variável (característica ou componente), conforme pode ser visualizado na figura abaixo:

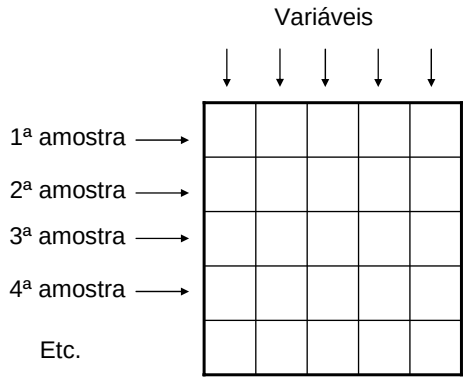


Figura 3.5 Tabela de dados (Adaptado de Fonte: [2])

A base de dados pode ser lida para o workspace do Matlab através de uma função disponível no toolbox chamada de “som_read_data”, que pode ser usada para ler arquivos no formato texto com colunas separadas por espaço. A primeira linha da base de dados deve conter o número de variáveis, na segunda linha precedidos de #n deve conter os nomes das variáveis. Para cada amostra de dados é possível inserir labels ou rótulos que podem facilitar a análise dos dados após o treinamento. Vale ressaltar que estes labels que determinam à classe na qual a amostra pertence é desconsiderado pelo algoritmo na fase de treinamento. Uma base de dados chamada Íris utilizada como exemplo no toolbox, pode ser visualizada na tabela 3.1 abaixo juntamente com seus respectivos rótulos.

Dados da Flor Íris				
Atributos (características)				*Rótulos (espécies)*
SepalL	*SepalW*	*PetalL*	*PetalW*	
4.9	3.0	1.4	0.2	Setosa
4.7	3.2	1.3	0.2	Setosa
4.6	3.1	1.5	0.2	Setosa
6.4	3.2	4.5	1.5	Versicolor
6.9	3.1	4.9	1.5	Versicolor
6.3	2.9	5.6	1.8	Virginica
6.5	3.0	5.8	2.2	Virginica
7.6	3.0	6.6	2.1	Virginica

Tabela 3.1 Base de dados Íris com algumas amostras. Fonte: [2]

A base acima disponível no toolbox contém um total de 150 amostras, sendo 50 de cada uma das 3 espécies (Setosa, Virginica e Versicolor) com os valores referentes a 4 características da flor Íris, comprimento da sépala (*SepalL*), largura da sépala (*SepalW)*, comprimento da pétala (*PetalL)* e largura da pétala (*PetalW*).

3.4 Pré-Processamento dos Dados

Antes de serem apresentados à rede, os dados devem passar por uma fase de pré-processamento. Nesta fase a utilização de funções de normalizações é importante para colocar todas as variáveis em uma mesma escala, pois o algoritmo utilizado no toolbox usa a distância euclidiana entre vetores, sendo assim se tivermos valores em escalas muito grandes e valores muito pequenos no conjunto de dados, certamente os valores muito grandes terão um maior impacto nas medidas das distâncias.

No Som toolbox existem seis tipos de normalizações que são usadas na fase de pré-processamento que podem ser feitas através da função som_normalize.

- 'var' – A variância é normalizada a um.
- 'range' – Os valores são normalizados entre [0,1] (operação linear).
- 'log' - Logaritmo natural é aplicado aos valores: novox = log (x-m+1) onde m = min (x).
- 'logistic' – Método que escala todos os valores possíveis entre [0,1]
- 'histD' - Equalização do histograma, valores escalonado entre [0,1]
- 'histC' – Idem ao anterior, porém com equalizações aproximadas. Valores escalonados entre [0,1].

É possível fazer um tipo de normalização para uma determinada variável e ao mesmo tempo utilizar outro tipo de normalização para outra variável do mesmo conjunto de dados. Um exemplo pode ser visualizado na tabela 3.2 abaixo:

Normalização dos Dados do Toolbox Som	
Parâmetros de normalização	Comentário
som_normalize (sD,'log',1)	normaliza as amostras da variável 1 (1ª coluna) com tipo 'log'
som_normalize(sD,'var',3);	normaliza as amostras da variável 3 (3ª coluna) com tipo 'var'
som_normalize(sD,'histD',1:3);	normaliza as amostras das variáveis 1 até 3 com tipo 'histD'

Tabela 3.2

Na tabela 3.2 sD é a base de dados em estudo. Uma vez que os dados foram normalizados, basta usar a função som_denormalize para obter os dados na escala original novamente digitando a seguinte linha de código: sD = som_denormalize (sDN). Onde sDN é a base de dados em estudo porém normalizada.

3.5 Inicialização, Treinamento, Visualização e Análise

Aplicações envolvendo o SOM exigem a elaboração de diferentes mapas variando a topologia, tamanho do mapa, função de aprendizado, função de vizinhança, entre outras, de tal forma a encontrar aquele mapa que obtenha os melhores resultados. No Som Toolbox existem algumas rotinas que torna possível realizar inúmeras combinações dos parâmetros citados além de funções de visualização e análises dos resultados. A forma como estas funções podem ser usadas será descrita a seguir.

Uma das principais funções do toolbox e também muito utilizada é a "som_make". É uma função básica que cria, inicializa e treina um SOM com parâmetros default e em duas fases: a primeira fase é a de ordenação do mapa e a segunda é a fase de convergência. É importante ressaltar que para se ter um melhor controle das escolhas dos parâmetros e funções a serem utilizadas, a som_make não deve ser utilizado e sim as funções som_lininit e som_randinit para a inicialização da rede e as funções som_seqtrain e som_batchtrain para o treinamento.

Um exemplo utilizando a função som_make pode ser visualizado na tabela 3.3.

Criação do SOM	
Parâmetros	Comentário
som_make (base de dados)	Cria e treina o SOM com parâmetros default

Tabela 3.3

Com esta linha de código os seguintes parâmetros são utilizados: algoritmo de treinamento (batch), função de inicialização (lininit), topologia dos neurônios (grade), formato de vizinhança (hexagonal), função de vizinhança (gaussiana), função para decréscimo da taxa de aprendizagem (inversa), raio inicial ($\frac{1}{8}$ do maior lado do mapa), raio final (valor padrão é 1, mas se for uma fase de ordenamento, o raio final corresponde a $\frac{1}{4}$ do raio inicial), duração do treinamento (na fase de ordenamento é de 10 vezes a razão entre número de unidades e quantidade de dados e se for na fase de convergência o valor é 40 vezes esta mesma razão).

Para o tamanho do mapa é utilizado uma fórmula heurística que determina quantos neurônios devem ter o mapa. O cálculo é feito da seguinte forma: cinco vezes a raiz quadrada do tamanho da amostra (n), ou seja, 5 x $\sqrt{n}$. As dimensões do mapa são determinadas também através de uma heurística. A razão entre os comprimentos dos dois lados do mapa é calculada como a raiz quadrada da razão dos dois maiores autovalores do conjunto de dados. Os comprimentos são então ajustados de modo que o seu produto seja próximo do número desejado de neurônios do mapa.

Se a som_make não for utilizada a inicialização da rede pode ser feita através de uma das duas rotinas existentes no toolbox chamadas de Lininit (inicialização linear) e Randinit (inicialização aleatória). Na Lininit, primeiramente são calculados os autovalores e autovetores dos dados de entrada. Em seguida, os vetores de peso são inicializados de uma forma ordenada ao longo dos dois maiores autovetores da matriz de covariância dos dados de entrada. Isto faz com que a convergência da rede seja mais rápida dispensando assim a fase de ordenamento, sendo necessária apenas a fase de convergência [2]. Abaixo na tabela 3.4 podemos visualizar um exemplo da inicialização utilizando a rotina Lininit.

Rotina de Criação e Inicialização do SOM	
Parâmetros	Comentário
sM =som_lininit(base_de_dados)	Cria e inicializa um mapa
sM = som_lininit(base_de_dados, ‘msize’ ,[4 9])	Idem ao anterior, porém o tamanho do mapa é dado.

Tabela 3.4

Na rotina Randinit as posições dos vetores de entrada estão totalmente desorganizadas. Para cada componente x_i, os valores dos vetores de pesos do mapa são uniformemente distribuídos no intervalo de $[\min(x_i),\max(x_i)]$. O mesmo exemplo citado acima pode ser utilizado com a rotina randinit, basta trocar o som_lininit por som_randinit.

O treinamento dos mapas é feito através das rotinas som_seqtrain que treina a rede SOM com o algoritmo de treinamento seqüencial, ou a rotina som_batchtrain que utiliza o algoritmo batch (lote) no treinamento da rede. Detalhes dos algoritmos citados aqui podem ser visualizados na seção3.2 e em [2]. Parâmetros como tamanho do raio de vizinhança podem ser especificados em ambos os treinamentos conforme podemos ver na tabela 3.5.

Rotina de Treinamento do SOM	
Parâmetros	Comentário
sM = som_seqtrain(SM,base_de_dados, ’radius’,[4 1])	Treina a rede com raio de vizinhança começando com 4 e terminando com 1.

Tabela 3.5

Onde sM é o mapa criado e inicializado acima com as funções de inicialização (lininit ou randinit).

Idem para o treinamento em lote, basta trocar som_seqtrain por som_batchtrain. Se nenhum parâmetro for especificado valores default são utilizados.

As funções de visualização permitem uma melhor análise dos dados no mapa. Geralmente ela é feita através da matriz-U que no toolbox é obtida através da função som_umat. A matriz–U retorna a matriz de distâncias unificadas do SOM dado. A idéia principal é usar a mesma métrica que foi utilizada durante o treinamento pra calcular as distâncias entre todos os neurônios vizinhos no arranjo. O resultado obtido a partir da aplicação da matriz-U sobre o mapa é uma imagem f(x,y) onde o nível de intensidade de

cada pixel corresponde a uma distância calculada. Citando como exemplo um mapa 2D de tamanho NxM, o resultado gerado será uma imagem (2N-1) X (2M-1). Detalhes desta técnica de visualização podem ser encontrados em [6]. A principal função utilizada para visualização é a som_show que mostra dentre a matriz_U, o plano de componentes, os labels dos dados, e outras visualizações que possa vir a ajudar na análise dos resultados. Um exemplo pode ser visualizado na tabela 3.6.

Rotinas de Visualização do Som	
Parâmetros de normalização	Comentário
som_show(sM,'umat','all','comp',1:N,' empty','labels','norm','d')	Mostra a matriz U e o plano de componentes
sM = som_autolabel(sM,sD,'vote')	Adiciona os labels (rótulos) no mapa
som_show_add('label',sM,'subplot',6)	Mostra o mapa com seus respectivos labels
som_show_add('hit', som_hits(sM,sD)	Coloca hits no mapa

Tabela 3.6

Onde sD é a base de dados e sM é o mapa já treinado e plano de componentes são as varáveis da base de dados e N é o número de variáveis.

3.6 Medidas de Erro

O Som toolbox apresenta duas métricas para avaliação da qualidade do mapa gerado após o processo de aprendizagem. As métricas são o Erro da Quantização Vetorial e o Erro Topográfico.

O erro de quantização representa a média das distâncias entre cada vetor de dados (m_c) e o correspondente vetor de pesos (v_n) do neurônico vencedor (BMU). Estima-se que quanto menor o erro de quantização, mais bem ajustado o neurônio vencedor estará aos vetores de entrada. O erro é calculado pela equação abaixo.

$$Qe = \frac{1}{N}\sum_{n=1}^{N} \| m_c - v_n \| \qquad \text{Equação 3.1}$$

O erro Topográfico quantifica a capacidade do mapa em representar a topologia dos dados de entrada. É calculado verificando-se para todas as entradas qual é o neurônio mais bem ajustado e também o segundo neurônio mais bem ajustado. É dado pela equação.

$$Te = \frac{1}{N}\sum_{n=1}^{N} u(v_n) \qquad \text{Equação 3.2}$$

A função que implementa estas duas métricas no toolbox é chamada de som_quality. Abaixo segue um exemplo para a chamada da função.

[Qe,Te] = som_quality (sM,sD).

4. Estudo de Caso

Um exemplo utilizando a base de dados da Tireóide e que está disponível para download no site ftp://ftp.ics.uci.edu/pub/machine-learning-databases/ foi utilizado nesta nota técnica como mais um objeto de estudo, com a finalidade de ajudar na utilização do Toolbox.

Na base de dados constam 215 amostras de pacientes do Instituto Garavan em Sydney, Austrália distribuídas em 3 grupos (classes) diferentes de pacientes. As classes são as seguintes:

1) Paciente Normal – classe1 com 150 amostras
2) Paciente com hypertiroidismo – classe 2 com 35 amostras
3) Paciente com hypotiroidismo – classe 3 com 30 amostras

As amostras são referentes a cinco atributos (no caso do mapa de Kohonen serão as variáveis de entrada) colhidos através dos exames de sangue dos pacientes. Os atributos são: Resina T3, Tiroxina total (T4), Triidotironina total (T3), Hormônio estimulador da tireóide (TSH), diferença absoluta máxima no valor da TSH após a injeção de 200 micros gramas de hormônio de liberação de tirotropina (na base de dados abaixo recebe o nome de TSHapos200). Parte da base de dados utilizada pode ser

visualizada na tabela 4.1 abaixo, juntamente com seus respectivos rótulos (1,2 ou 3). Vale ressaltar que os rótulos apresentados na tabela não influenciam no resultado do treinamento. Os mesmos foram apresentados na tabela apenas para facilitar a visualização do resultado do treinamento.

Amostras da Base de Dados Tireóide					
Resina	T3Tiroxina	Triidotironina	TSH	TSHapos200	Label
112	95	2	12	7	1
98	86	16	16	6	1
109	124	23	17	8	1
139	164	38	11	-2	2
111	16	21	9	-1	2
113	172	18	1	0	2
126	5	2	122	88	3
121	47	18	112	53	3
131	27	8	99	47	3
134	2	5	122	22	3

Tabela 4.1

Para a criação do mapa foi utilizada a função som_make do Toolbox que cria, inicializa e treina o mapa com parâmetros default. Sendo assim após os cálculos para determinação dos valores dos parâmetros defaut (ver seção 3.5) teremos os seguintes valores para os mesmos:

- Tamanho do mapa: 5 x $\sqrt{215}$ = 73
- Dimensões do mapa: 12 x 6, que dá um total de 72 neurônios
- Algoritmo de treinamento: em lote com duas fases (fase de ordenação e fase de ajuste fino ou convergência)
- Inicialização: rotina Lininit
- Topologia: em forma hexagonal
- Formato de vizinhança: dado pela função gaussiana
- Taxa de aprendizagem: o algoritmo em lote não a utiliza
- Número de Épocas
  - Rough training phase (Fase de Ordenamento): (10* (72/215)) = 4
  - Finetuning phase (Fase de convergência): (40* (72/215)) = 14
- Raio de vizinhança inicial: 12/8 = 2
- Raio final: igual a 1

- Erro de Quantização Final: 0,697
- Erro Topográfico Final: 0,019

Depois de treinado o mapa pode ser visualizado através da matriz-U (rotina som_show). A matriz-U (matriz de distâncias unificadas) conforme citado anteriormente é um método de visualização de um SOM treinado, que permite a detecção visual das relações topológicas dos neurônios. A figura 4.1 abaixo mostra à esquerda a matriz-U e à direita o mapa com seus respectivos labels, onde podemos visualizar nitidamente as classes 1 2 e 3 referente aos três agrupamentos.

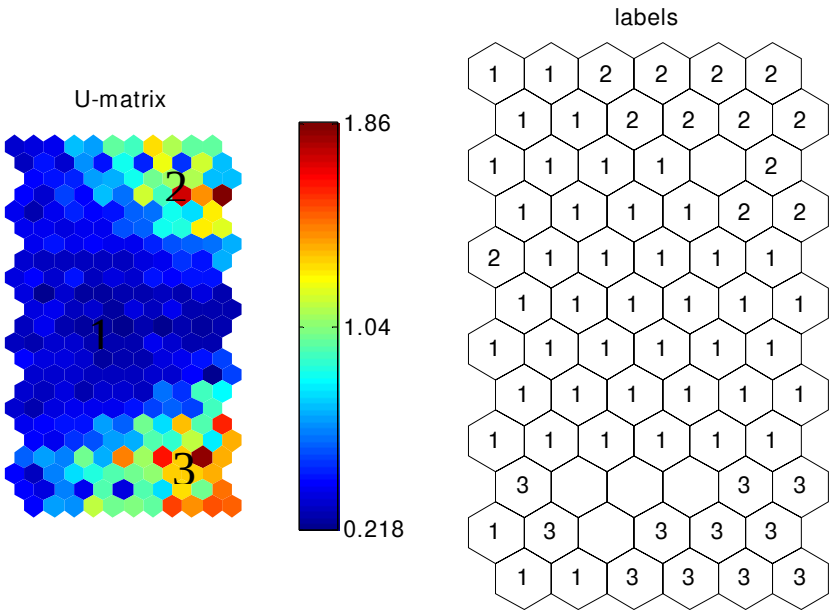


Figura 4.1 Matriz-U e Mapa com seus respectivos labels

Na matriz –U acima é possível visualizar que os dados foram separados em três conjuntos conforme apresentado na figura (clusters 1,2 e 3). A divisão ocorre nas linhas mais claras tendendo ao vermelho, pois os valores altos da legenda de distância (1,04 até 1,86) é que indicam a borda dos clusters e a cor azul escuro (0,218) representa os neurônios mais próximos, ou seja, os próprios agrupamentos [7].

Uma outra forma também de visualizar os agrupamentos é através de um dendograma que pode ser visualizado na figura abaixo. No dendograma é preciso determinar a distância de corte para definir quais serão os grupos formados. Este corte deve ser feito de acordo com o número de grupos desejados. Neste sentido esta forma de visualização só é apropriada quando se conhece a quantidade de grupos a priori, pois a

determinação da quantidade de grupos é fundamental para a determinação correta do ponto de corte. No exemplo citado como o número de grupos já é conhecido é possível determinar o ponto de corte na altura de 1,04, ou seja, a reta tracejada nesta altura intercepta o gráfico em três pontos indicando a existência de três agrupamentos.

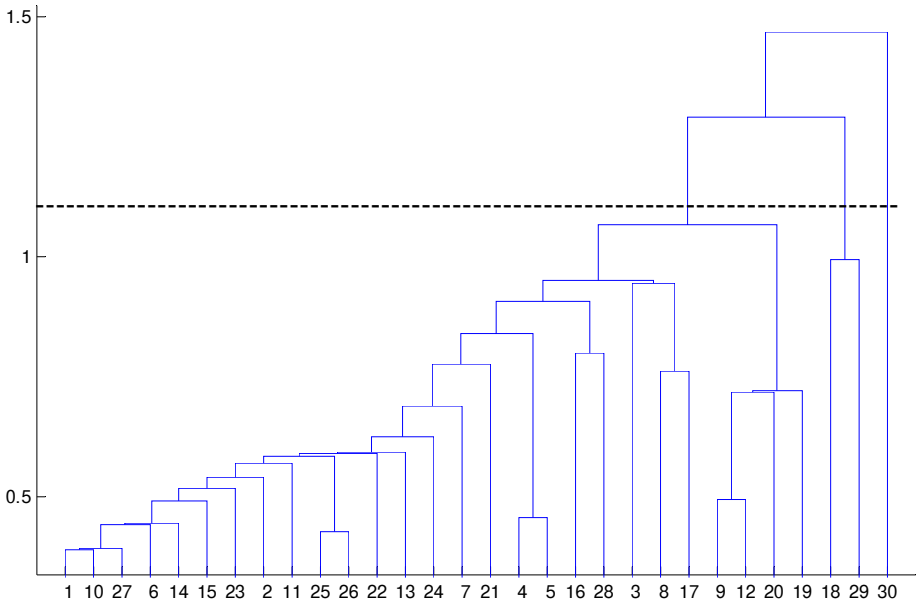


Figura 4.2 Dendograma dos agrupamentos

Na figura 4.3 abaixo pode ser visualizado a matriz-U novamente e o plano das componentes, onde é visualizada uma matriz-U para cada variável do conjunto de dados (Resina, T4, T3, TSH, TSHapós200). Assim como citado anteriormente a cor azul representa a proximidade dos vetores de pesos aos neurônios enquanto a cor vermelha significa o contrário. Do ponto de vista visual é possível perceber que as matrizes TSH e TSHapós200 são muito semelhantes, o que pode indicar uma correlação entre os atributos. Outras características que também podem ser visualizadas são as seguintes: o cluster 2 referente a hipertiroidismo apresenta Resina com valores baixo, T3 e T4 valores altos, TSH com valores baixo e TsHapos200 também com valores baixo. Já o cluster 3 referente à hipotiroidismo apresenta Resina alta, T3 e T4 baixo, TSH e TSHapos200 alta.

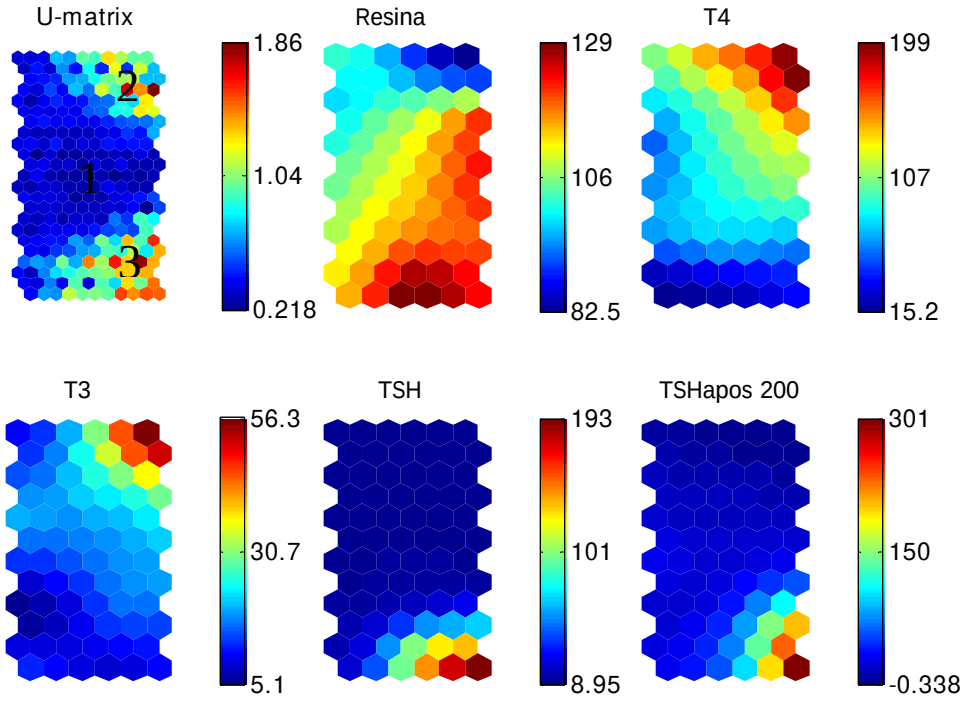


Figura 4.3 matriz-U e Plano de Componentes

Outra forma de visualização que facilita na análise dos resultados é através dos hits de histograma (figura 4.4). Através deles é possível determinar qual a parte do mapa que melhor corresponde aos dados. Um BMU é o neurônio vencedor que melhor representa um determinado vetor de dados, sendo assim a visualização dos hits mostra quantos dados foram mapeados para cada unidade do mapa.

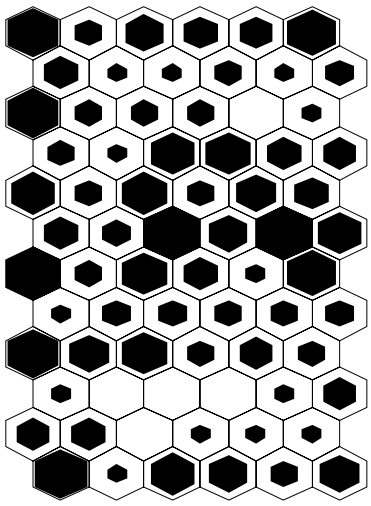


Figura 4.4: Hits

Através da figura é possível perceber que a distribuição dos dados em relação aos seus neurônios vencedores (BMUs) está compatível com a quantidade de dados, pois temos poucos neurônios inativos e aparentemente nenhum neurônio com sobrecarga.

5. Conclusão

Os Mapas Auto Organizáveis de Kohonen (SOM) apresentam recursos muito úteis quando se trata de agrupamentos de dados. Quando se está trabalhando com estes tipos de mapas é muito importante uma boa ferramenta, pois pode ajudar não só na formação dos grupos, mas também no entendimento dos mesmos.

Esta nota técnica apresentou uma ferramenta muito utilizada na implementação das redes Kohonen ou SOM (Som Toolbox para Matlab 5.1 ou superior). Foi feito primeiramente uma breve introdução sobre as Redes Kohonen a fim de facilitar o entendimento de como as mesmas são abordadas pela ferramenta. Em seguida foi apresentada a forma como o Som Toolbox funciona e por fim foram apresentadas também as principais rotinas responsáveis pela criação, treinamento e visualização do SOM.

Um estudo de caso foi utilizado para exemplificar as principais funções e mostrar como o mapa pode ser analisado. Alguns pontos relevantes sobre o estudo são destacados a seguir:

- É possível destacar como ponto positivo a rapidez com que o mapa foi gerado e treinado, não havendo a necessidade de um número muito grande de iterações para se obter um bom resultado.
- Outro ponto a ser destacado é a forma de visualização do mapa já treinado (através da matriz-U). Esta técnica permite a visualização da quantidade de grupos formados através de uma matriz de cores. Neste sentido a utilização da matriz-U como técnica de visualização é melhor quando comparada a utilização de outras técnicas de visualização como o dendograma, pois ela não necessita do número de grupos a priori.
- Podemos destacar também que uma vez determinada à quantidade de grupos é possível através do plano de componentes visualizarem as características dos grupos gerados.

Por fim podemos concluir que a ferramenta apresentada neste relatório possui recursos que nos permite fazer um estudo completo sobre agrupamentos de dados.

6. Bibliografia

[1] Kohonen, Teuvo "***Self Organizing Maps***". Series in Information Sciences, Vol 30, 2nd edition Springer – Verlag, Heidelberg, 1997.

[2] Vesanto, J.; Himberg, J.; Alhoniemi, E.; Parhankangas, J. "***SOM Toolbox for Matlab 5"***: report A57, April 2000. Libella Oy: Finland: SOM Toolbox Team, Helsinski University of Technology, 2000b. 59 p., ill. Disponível em: http://www.cis.hut.fi/projects/somtoolbox/download/. Visualizado em 02/07/08

[3] MATLAB for Windows User's Guide, The Math Works Inc., 1991.

[4] kohonen,Teuvo; Hynninen, Jussi; Kangas, Jarí; Laaksonen, Jorma. "***Som_Pack. The Self – Organizing Map Program Package"***. Version 3.1. Helsinki University of Technology, Laboratory of computer and Information Science, Finland, April 7, 1995. URL: http://www.cis.hut.fi/research/som_pak/som_doc.txt. Visualizado em 02/07/08.

[5] S. Haykin, ***"Redes Neurais. Princípios e prática".*** Ed. Bookman (2001).

[6] Vesanto, J.; Himberg, J.; Alhoniemi, E.; Parhankangas, J. "***Self-Organizing Map in Matlab: the SOM Toolbox"***. Finland: Helsinski University of Technology. Laboratory of Computer and Information Science, 2000a. 8 p., ill. Disponível em: http://www.cis.hut.fi/projects/somtoolbox/package/papers/toolbox2paper.pdf. Visualizado em 07/10/08

[7] Vesanto, Juha: Himberg, Johan: Alhoniemi, Esa: Parhankangas, Juha. ***"SOM Toolbox for Matlab 5".*** Technial report A57. Helsinki University of Tecnology. Finland 2000, URL: http://www.cis.hut.fi/projects/somtoolbox/package/papers/techrep.pdf. Visualizado em 10/10/2008.