Instituto de Ciências Matemáticas de São Carlos

ISSN - 0109-2569

A Rede de Kohonen

ROSELI A. FRANCELIN
LUCIANE R. VIEIRA

Nº 31

RELATÓRIOS TÉCNICOS DO ICMSC

São Carlos
Dez./1994

SYSNO 872879
DATA / /
ICMC - SBAB

A Rede de Kohonen

Roseli A. Francelin
USP - ICMSC - SCE
13560 - São Carlos - SP - Brasil

Luciane R. Vieira *
USP - ICMSC - SCE
13560 - São Carlos - SP - Brasil

Sumário

O presente trabalho apresenta um estudo da rede neural de Kohonen, que vem sendo utilizada com muito sucesso em reconhecimento de vozes. As características da rede são discutidas e uma correspondência entre este modelo de rede neural e o método de Análise de Cluster é estabelecida. Exemplos ilustrativos são apresentados para mostrar esta correspondência e o desempenho da rede.

1 INTRODUÇÃO

O que nos motivou a escrever este trabalho foi a necessidade que sentimos, quando começamos a estudar a rede de Kohonen, de um material que fosse mais didático, fornecendo informações sobre esta rede, e que nos poupasse de uma pesquisa bibliográfica mais intensa, uma vez que se deseja apenas numa informação geral deste modelo.

Desta forma, o presente trabalho apresenta um estudo, baseado em diversos trabalhos existentes sobre este modelo, da rede neural de Kohonen, que vem sendo utilizada com muito sucesso para Classificação de padrões e particularmente, na difícil tarefa de reconhecimento de vozes.

Inicialmente, as caracterísctcias da rede são apresentadas e as seguintes correspondências são estabelecidas: Rede de Kohonen e o método de vetorização: "Vector Quantizer" e Rede de Kohonen e método de classificação: Análise de Cluster.

Em seguida, um algoritmo estruturado é proposto com a finalidade de fornecer ao leitor um meio mais rápido de verificar o funcionamento da rede.

*Este trabalho recebeu o apoio do CNPq

Finalmente, alguns exemplos ilustrativos são discutidos para mostrar o comportamento e desempenho desta rede.

2 A REDE DE KOHONEN

A Rede de Kohonen foi proposta por Teuvo Kohonen da Universidade Helsinki de Tecnologia na Finlândia e surgiu devido ao interesse que o grupo de Reconhecimento de Padrões, do Departamento de Estatística desta Universidade, encontrou na tarefa de extrair características de um conjunto de fonemas observados, usando elementos da Teoria de Probabilidades e da Teoria das Decisões Clássicas.

Partindo do fato que dois dos mais importantes campos deste Departamento são o **Método da Função Discriminante** e a **Classificação do Vizinho mais Próximo** e que o primeiro usa somente funções analíticas, T. Kohonen propôs um modelo de Rede Neural que desempenha bastante bem a tarefa de reconhecimento de padrões, mais especificamente a tarefa de reconhecimento de fonemas (vozes), baseando-se então no método de Classificação do Vizinho mais próximo.

Este método interpreta um conjunto de dados como padrões e usa estatísticas tradicionais para formar classes C_i de padrões. onde cada classe é representada por um pequeno conjunto de padrões protótipos P_i. Cada novo padrão p pode ser classificado através da identificação da classe C_p para a qual p é mais próximo com respeito a alguma métrica já definida.

A Rede de Kohonen apresenta um modelo relativamente simples e mesmo assim, fascinante, de como Redes Neurais podem "aprender sozinhas" ou "auto-organizar-se", sem que seja dado uma resposta correta para um dado modelo de entrada, condição esta, necessária em muitos outros modelos de redes neurais.

O modelo de Kohonen nos oferece uma intrigante visão de como nosso cérebro pode realmente trabalhar, uma vez que forma "mapas de preservação da topologia", que são também uma característica dos cérebros biológicos.

Estes mapas que preservam a topologia da rede e que são formados através de um mapeamento de organização própria, têm seu suporte na teoria de "Vector Quantizer", que será descrita brevemente na seção 2.2. Em outras palavras, isto significa que a ordem espacial dos padrões apresentados à rede é preservada no processo de aprendizagem e analisada através de uma visão externa da própria rede.

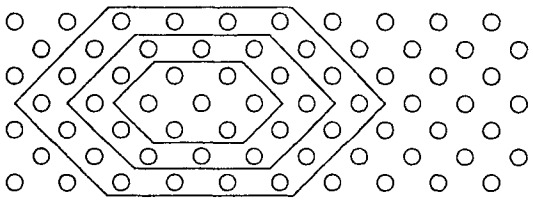


Figura 1: Estrutura Hexagonal

2.1 CARACTERÍSTICAS

A Rede de Kohonen é uma rede neural que possui:

- ESTRUTURA NÃO-HIERÁRQUICA: todos os neurônios da rede são do mesmo tipo, desempenham a mesma função e pertencem a uma única camada, chamada *Camada de Kohonen.*
- APRENDIZADO NÃO-SUPERVISIONADO: possui auto-organização; pode fornecer uma saída correta para uma dada amostra de entrada, sem que a saída esperada tenha sido fornecida.

A camada de Kohonen é um arranjo de neurônios numa aplicação prática de duas-dimensões, onde todos os neurônios são conectados aos seus vizinhos mais próximos, respeitando sempre a estrutura da rede, que pode ser estrutura hexagonal (Fig. 1) ou estrutura retângular (Fig. 2), e é esta estrutura que define o conjunto de vizinhos mais próximos de cada neurônio.

Todos os neurônios da rede atuam como:

- *Unidades de Entrada:* recebem os mesmos sinais de entrada em paralelo, além de receberem os sinais de seus vizinhos *(feedback)* através de conexões laterais com outros neurônios (Fig. 3).
- *Unidades de Treinamento:* cada neurônio da rede tem associado, inicialmente, um vetor de valores reais randômicos de mesma dimensão do vetor de entradas. Estes vetores são chamados *Vetores Peso* e conterão, depois da estabilização da rede (treinamento), o conhecimento do problema dado.

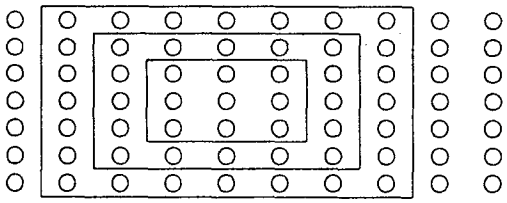


Figura 2: Estrutura Retangular

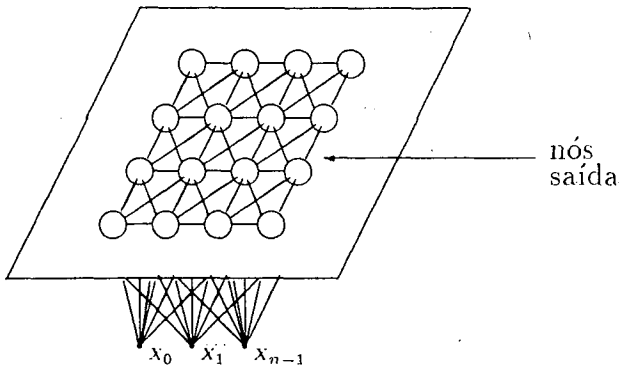


Figura 3: Rede de Kohonen

O treinamento da rede de Kohonen consiste na tarefa específica de encontrar um conjunto correto de Vetores Pesos que representam um dado conjunto de amostras de entradas.

- *Unidades de Saída:* fornecem em paralelo uma visão externa do resultado obtido, isto é, o resultado obtido pelo treinamento da rede é visto na própria camada de Kohonen e é um mapeamento das entradas n-dimensionais para o espaço bi-dimensional de saída (Fig. 3).

A rede de Kohonen possui duas habilidades fundamentais:

- MODELAR A DISTRIBUIÇÃO DAS ENTRADAS: a rede modela a função distribuição de probalidade dos vetores entrada usados durante o treinamento. Este modelo é representado através da separação dos vetores Peso em diferentes agrupamentos. Cada agrupamento (ou classe) representa entradas que são similares.

  Acredita-se, que quando o modelo estatístico de distribuição dos dados não é conhecido, pode-se usar a rede de Kohonen para se obter uma aproximação da distribuição de probabilidade dos dados.

- CRIAR MAPAS QUE PRESERVAM A TOPOLOGIA DA REDE: isto significa que após o processo de aprendizagem da rede, um mapa topológico do espaço dos vetores de entrada da rede, que é visto como uma matriz ou um subconjunto discreto da função R^2, é formado.

  Assim, supondo que os vetores de entrada são de dimensão N (pertencem ao espaço R^n) e a rede é de dimensão 2 com r linhas e c colunas, a rede treinada é um mapeamento de

  $$R^n \rightarrow [1,r]\mathbf{x}[1,c].$$

  Este mapeamento preserva a topologia natural, ou seja, se tivermos dois vetores do conjunto de entrada geometricamente próximos um do outro, então, eles serão mapeados em dois pontos também próximos um do outro (vizinhos) na matriz rede [1,r]x[1,c] .

2.2 REDE KOHONEN POSSUI AUTO-ORGANIZAÇÃO

A seguir descreveremos brevemente sobre a definição de "Vector Quantizer", mais comumente chamado de "Vetorização dos dados". Maiores detalhes podem ser encontrados em [GRAY 84] e [KANG 91].

2.2.1 VECTOR QUANTIZER (VQ)

Um VQ é um sistema de mapeamento de uma sequência de vetores contínuos ou discretos numa sequência digital adequada para comunicação sobre um canal digital ou armazenagem dos mesmos numa memória digital. Na verdade, a relação entre uma sequência de vetores dados e a sequência digital requerida é estabelecida. A regra dos sistemas de quantização é produzir a melhor sequência de reprodução possível para uma dada taxa de dados.

Para se ter uma idéia da utilização de VQ os seguintes exemplos são citados:

1. quando uma sequência de dados analógicos ou discretos de taxa muito alta é convertida numa sequência de dados de taxa relativamente baixa num canal de comunicação.

2. quando a quantidade de dados a ser armazenada é muito grande e precisa-se garantir a eficiência, segurança e confiabilidade dos dados.

Para quantificar esta idéia, foi definido o desempenho de um "quantizer" e para completar a definição de um "quantizer", foi requerida a idéia de *Medida de Distorção.*

Assim, podemos quantificar o desempenho de um sistema pela distorção média entre a entrada e a saída reproduzida. E um sistema será bom se produzir uma distorção pequena.

Medidas de Distorção mais usadas:

- DISTORÇÃO DO ERRO QUADRADO: usada na Rede de Kohonen, onde as entradas e o espaço de reprodução são os espaços euclidianos k-dimensionais.

- DISTORÇÃO DE ITAKURA-SAITO: bastante usada para aplicações de codificação de vozes e também em classificações de modelos estatísticos.

Devemos notar que a palavra "quantizer" é quase sempre associada com mapeamentos de distorções mínimas. Com base em VQ, outros métodos de mapeamento foram obtidos: Aprendizado Vector Quantizer (LVQ) que possui várias modificações: LVQ1, LVQ2, LVQ2.1, LVQ3 [KANG 91].

No método LVQ1, modificações são aplicadas para cada apresentação, tanto para punir as classificações incorretas quanto para recompensar uma correta.

No método LVQ2, modifica-se o vizinho mais próximo do elemento escolhido ou o segundo mais próximo, sempre que o primeiro mais próximo é de uma classe diferente e o segundo mais próximo é da mesma classe do elemento escolhido.

O método LVQ2.1 aplica modificações sobre classificações incorretas tanto quanto sobre classificações corretas.

LVQ3 é a combinação dos algoritmos anteriores.

O mapeamento para cada vetor pode ou não ter memória, no sentido de dependência de ações passadas do código.

"Vector Quantizer" com memória ou VQ sem memória são bastante usados em estudos para aplicações em códigos de vozes e códigos de imagens.

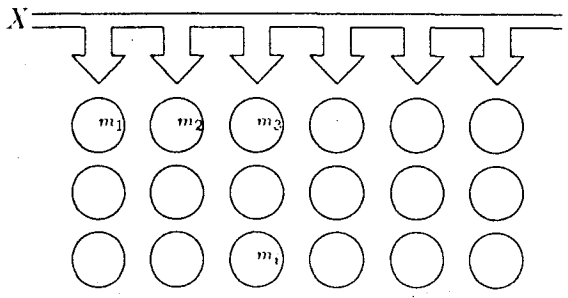


Figura 4: Entradas na Rede de Kohonen

Em virtude da Rede de Kohonen fornecer um mapeamento do $R^n \rightarrow [1,r]\mathbf{x}[1,c]$, diz-se, então, que o algoritmo de Kohonen cria um *Vector Quantizer*, através do ajuste das entradas às saídas no *"grid"* bi-dimensional *(matriz)*.

2.2.2 VECTOR QUANTIZER X REDE KOHONEN

Como já foi dito, VQ é um sistema de representação (mapeamento) de uma sequência de vetores contínuos ou discretos numa sequência digital adequada. Desta forma, mais especificamente para a Rede Kohonen, VQ é um sistema de mapeamento de vetores de entrada do espaço n-dimensional para um espaço de saída bi-dimensional que, após o aprendizado, representará as entradas.

O aprendizado VQ descrito por Kohonen é um método de classificação do tipo KNN *(vizinho mais próximo)* [GEVA 91], no qual um número fixo de vetores protótipos (vetores pesos iniciais) são progressivamente modificados para mapear o espaço de entradas.

Durante o aprendizado VQ da rede de Kohonen, temos que todos os neurônios do conjunto da rede, que formam o chamado mapa de preservação de topologia, recebem o vetor de entrada X (Fig. 4): o neurônio deste mapa que apresentar menor distância a esta entrada é dito *neurônio escolhido*.

A partir dele uma vizinhança de raio progressivamente decrescente vai sendo modificada, isto é, os pesos dos neurônios pertencentes a esta vizinhança são modificados (Fig. 5). Este processo é repetido para todas as entradas apresentadas a rede e por fim, o mapa bi-dimensional de saída representará o espaço n-dimensional das entradas.

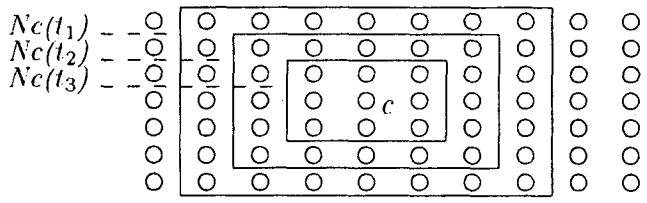


Figura 5: Vizinhanças em torno do nó **c**

3 CLASSIFICAÇÃO DE PADRÕES

O método de classificação de padrões no qual a Rede de Kohonen é baseada é denominado Análise de Clusters. O conhecimento deste método facilita a compreensão do funcionamento da Rede de Kohonen. A seguir, descreveremos o método de Análise de Clusters.

3.1 Análise de Clusters

O método de classificação de Análise de Clusters foi proposto para solucionar problemas do tipo:

"dada uma amostra de **n** objetos, cada um deles representado por **p** variáveis, necessita-se de um esquema para agrupar os objetos em diferentes classes de modo que os similares fiquem em uma mesma classe".

Nota-se que o número de classes não é conhecido, como acontece na maioria dos métodos de classificação de padrões.

Existem várias situações onde a aplicação de Análise de Clusters é adequada. Uma delas ocorre quando se deseja encontrar as classes verdadeiras. Por exemplo, em psiquiatria, havia um grande desacordo na classificação de pacientes depressivos, e Análise de Clusters foi usada para definir grupos objetivos. Outra situação, é que Análise de Clusters pode ser útil na redução dos dados. Como um exemplo, vamos supor que um grande número de cidades devem ser usadas num teste de mercado para um novo produto. Entretanto, como para a realização do teste um grande número de recursos estão envolvidos, seria conveniente, usar um número menor de cidades, sem prejudicar os resultados do teste.

Desta forma, se as cidades forem agrupadas em um número pequeno de grupos de cidades similares, então um membro de cada grupo poderá ser usado para o teste de mercado.

Muitos algoritmos foram propostos para Análise de Clusters (ver [HART 75]) para maiores informações). A seguir são citados alguns deles.

- Técnica Hierárquica: este método começa calculando a distância de cada indivíduo a cada um dos outros indivíduos. Grupos são então formados por um processo de **aglomeração** ou **divisão** [JENS 91].

  No processo de **aglomeração**, inicia-se com cada objeto sendo um grupo. Em seguida, grupos similares são gradualmente unidos até que, finalmente, todos indivíduos estejam num único grupo.

  No processo de **divisão**, cada objeto pertence ao mesmo e único grupo. Este grupo é dividido em dois grupos, esses dois grupos são novamente divididos e assim sucessivamente, até que todos os objetos estejam em seus próprios grupos.

- Técnica de Particionamento: nesta técnica, os objetos movimentam-se dentro ou fora dos grupos no decorrer dos diferentes estágios da análise. Inicia-se, com alguns centróides de grupos mais ou menos arbitrários, e indivíduos são colocados no grupo com distância mais próxima a eles. Um indivíduo é então movido para um novo grupo se ele é mais próximo a este grupo do que ao centróide do seu próprio grupo. Desta forma, grupos similares são unidos e um grupo que não possua todos seus elementos similares pode ser dividido. O processo continua, iterativamente, até que a estabilidade seja encontrada para um número pré-determinado de grupos.

- O algoritmo K-means é descrito em detalhes em [HART 75]. Este algoritmo é uma variação do método análise de clusters. Neste método supõe-se que o número de clusters já está definido. Inicialmente, os dados são distribuídos nestes clusters de acordo com um determinado critério. Em seguida, o método vai alterando esta separação dos dados, em função das médias dos clusters atuais. Em função destas médias, um elemento poderá migrar-se de um cluster para outro. O processo [DUDA 73] termina quando os elementos pertencerem as respectivas classes corretas de acordo com uma taxa de erro permitida.

  O algoritmo Kohonen será similar ao *Algoritmo de classificação K-means* por apresentar um VQ sequencial viável quando o número de classes desejado pode ser especificado antes do uso e quando os dados do treinamento são relativamente grande em relação ao número de classes desejadas. K-means, como citado acima, é um dos métodos existentes para a Classificação de Padrões. [LIPP 87].

As principais funções da Análise de Clusters são: nomear, exibir, sintetizar e prever. Assim todos os objetos num mesmo cluster terão o mesmo nome. Objetos são apresenta-

dos, de forma que pequenas diferenças podem tornar-se aparentes através dos diferentes agrupamentos. Os dados são sintetizados pois as propriedades dos clusters passam ser analisadas no lugar das propriedades dos objetos individualmente. Se alguns objetos num cluster têm uma certa propriedade é esperado que outros objetos neste cluster também gozem desta mesma propriedade.

Alguns dados de leite de mamíferos serão usados para ilustrar estas funções. Estes dados encontram-se em [HART 75], págs. 6 e 7 e são exibidos na Tabela 1, onde animais com proporções de constituintes do leite são separados em diferentes grupos.

Os grupos são nomeados de acordo com as propriedades dos grupos. Por exemplo, o grupo Veado-Baleia com porcentagem alta de gordura, enquanto o grupo Cavalo-Mula é caracterizado por porcentagem alta de lactose. Já o grupo Coelho-Rato é um grupo que apresenta alta taxa de proteina.

Uma síntese dos dados é apresentada na Tabela 2. Esta síntese faz com que os dados possam ser analisados de uma forma mais simples. Por exemplo, pode-se observar que as porcentagens de Água e Lactose crescem juntas. O mesmo acontece com as porcentagens de Proteina e Gordura e que estes dois grupos, por sua vez, são inversamente proporcionais, isto é, para porcentagens altas de Água e Lactose corresponde à baixas porcentagens de Proteina e Gordura.

Finalmente, previsão pode ocorrer de dois modos distintos. Primeiro, se um objeto é classificado num destes grupos por algum outro método, os mesmos valores seriam esperados para as variáveis. Então, espera-se que um rato esteja no grupo Coelho-Rato e que tenha um conteúdo de proteina em torno de 10,8 %. Segundo, se uma nova medida de um objeto de um determinado grupo foi obtida, pode-se prever que outro elemento deste grupo também terá esta nova medida. Por exemplo, se um cavalo tem 1,3 % de uma proteina especial, é esperado que um burro também tenha 1,3 % aproximadamente daquela proteina.

Mamíferos	Água	Proteina	Gordura	Lactose
Cavalo	90.1	2.6	1.0	6.9
Burro	90.3	1.7	1.4	6.2
Mula	90.0	2.0	1.8	5.5
Camelo	87.7	3.5	3.4	4.8
Lhama	86.5	3.9	3.2	5.6
Zebra	86.2	3.0	4.8	5.3
Carneiro	82.0	5.6	6.4	4.7
Bufalo	82.1	5.9	7.9	4.7
Porco da Guinea	81.9	7.4	7.2	2.7
Raposa	81.6	6.6	5.9	4.9
Porco	82.8	7.1	5.1	3.7
Coelho	71.3	12.3	13.1	1.9
Rato	72.5	9.2	12.6	3.3
Veado	65.9	10.4	19.7	2.6
Rena	64.8	10.7	20.3	2.5
Baleia	64.8	11.1	21.2	1.6

Tabela 1 - Composição de Leites de Mamíferos

	Número	Água	Proteina	Gordura	Lactose
Cavalo-Mula	3	90.1 ±0.2	2.1 ±0.5	1.4 ±0.4	6.2 ±0.7
Camelo-Zebra	3	87.0 ±0.8	3.5 ±0.5	4.0 ±0.8	5.2 ±0.4
Carneiro-Porco	5	82.2 ±0.6	6.5 ±0.9	6.5 ±1.4	3.8 ±1.1
Coelho-Rato	2	71.9 ±0.6	10.8 ±1.6	12.9 ±0.3	2.6 ±0.7
Veado-Baleia	3	65.3 ±0.6	10.8 ±0.3	20.5 ±0.8	2.1 ±0.5

Tabela 2 - Agrupamento dos dados da Tabela 1

3.2 Classificação de Padrões através da Rede de Kohonen

Existem duas fases distintas no processo de classificação de padrões:

- 1A. FASE (TREINAMENTO OU APRENDIZADO): Nesta fase, os modelos (dados) são apresentados sequencialmente e formam sozinhos, após um aprendizado, classes distintas (clusters), onde entradas semelhantes agrupam-se e entradas distintas afastam-se. Durante o processo de aprendizado modificações ocorrem nos pesos para que possam representar as entradas preservando sempre suas características.

  O algoritmo de Kohonen formará clusters *(classes)* durante o treinamento e o conjunto de dados de entrada deverá ser relativamente grande se comparado ao número de classes [LIPP 87].

A Rede de Kohonen possui uma fase de aprendizado, chamada *Aprendizado Competitivo*, pois é um processo no qual um neurônio "Vencedor" relativo a cada entrada é identificado e as células da rede neural são adaptadas para especificar as características de cada entrada. Isto acarretará que as respostas da rede tendam a se tornar localizadas.

O princípio básico fundamental do aprendizado competitivo fundamenta-se no algoritmo *Análise de Classes (Cluster Analysis)*, consistindo de:

- uma sequência de exemplos na forma vetorial $X = X(t)$ onde X pertence ao espaço R^n e t é a coordenada tempo;
- um conjunto de variáveis vetores referência $W^i(t)$ onde $W^i(t) \in R^n$, $i = 1, 2, 3, \ldots, k$. Para $t = 0$ os $W^i(t)$ são inicializados com valores randômicos.

A idéia básica do aprendizado competitivo consiste no seguinte critério: se a entrada $X(t)$ puder de algum modo ser comparada com todos os $W^i(t)$ em cada sucessivo instante de tempo t, então o $W^i(t)$ mais próximo da entrada $X(t)$ é modificado de forma a melhor representar esta entrada. A comparação é feita por uma medida de distância. O melhor $W^i(t)$ encontrado é modificado e os demais ficam intactos. Assim, depois de um longo aprendizado seguindo este critério, os diferentes domínios das variáveis de entrada X, e os $W^i(t)$ tendem a descrever as classes *(clusters)*.

- 2A. FASE (CLASSIFICAÇÃO OU RECONHECIMENTO) : Um número de classes distintas já foi definida durante o treinamento. Nesta fase cada nova entrada apresentada é comparada aos conjuntos existentes. A nova entrada é classificada como pertencente a classe que lhe for mais similar.

O Algoritmo Kohonen funciona como classificador devido ao fato de uma parte do treinamento da rede, engrandecer a competição da entrada estimulada num conjunto limitado de neurônios. Deste modo, regiões de neurônios com vetores Peso similares são criadas. Estas regiões tendem a ser ordenadas topologicamente do mesmo modo que os correspondentes vetores de entrada. Assim, a rede pode ser vista como um classificador de vetores entrada em classes de similaridade topológicas. O vetor Peso do melhor neurônio da competição será um novo padrão protótipo da classe na qual a entrada apresentada é classificada. E este é um exemplo típico de *Classificação pelo Método do Vizinho Mais Próximo* [GEVA 91].

Quando mapas ou qualquer método VQ são usados para classificação, como é o que ocorre na Rede Kohonen; cada categoria (classe) de exemplos é usualmente aproximada por muitos vetores Peso. A exata localização dos vetores Peso no interior da classe não é importante; ter poucos vetores no interior das classes pode ser suficiente, mas muitos vetores localizados nos limites das classes, onde ocorre a intersecção das classes de distribuição, causam erros de mistura entre classes.

4 ALGORITMO

Para compreensão da dinâmica da Rede de Kohonen, apresentamos nesta seção um algoritmo sequencial, na forma estruturada, correspondente ao processamento da mesma.

4.0.1 DINÂMICA DA REDE KOHONEN

- Cria-se uma matriz de m linhas por c colunas, que representará a chamada camada de Kohonen. Cada elemento desta camada será um vetor de elementos igual ao número de componentes (características) de cada entrada. A Camada de Kohonen deve ser suficientemente grande em relação ao número de entradas.

- Para cada vetor de entrada $X(t_k)$, onde X é um vetor de componentes e t_k é o tempo discreto correspondente, executa-se os seguintes passos:

  1. Identifica-se um neurônio c com a entrada apresentada, isto é, o neurônio com menor distância em relação a entrada apresentada é escolhido. Esta distância é calculada pela seguinte fórmula:

     $$d_j = \sqrt{\sum_{i=1}^{n} (W_i^j - X_i)^2}$$

     onde d_j representa a distância entre o j-ésimo neurônio da rede e a entrada X.

  2. Atualizar gradualmente este neurônio e sua vizinhança Nc. Por gradualmente entende-se atualizar desde um raio inicial que é igual a largura da rede até um raio final igual a zero, que corresponde a atualizar somente o neurônio escolhido.

     A vizinhança Nc é centralizada ao redor do neurônio selecionado e é chamada *bolha ou núcleo.*

     A atualização gradual de um neurônio j ocorre na direção do vetor entrada X pela seguinte fórmula geral:

     $$\frac{dW^j}{d_t} = \begin{cases} \alpha X - \beta W^j & , j \in Nc \\ 0 & ,\text{c.c.} \end{cases}$$

     onde α e β são funções do tempo e determinam o quanto W^j deve caminhar na direção X. Na prática assume-se que $\alpha = \beta$. Assumindo, também, uma discretização do tempo temos que a fórmula acima pode ser reescrita na seguinte forma:

$$W^j(t+1) = \begin{cases} W^j(t) + \alpha(t)[X(t) - W^j(t)] & , i \in Nc(t) \\ W^j(t) & , \text{c.c.} \end{cases}$$

onde t é o tempo discreto indexado, que aumenta de uma unidade a cada variação de vizinhança e a cada nova entrada apresentada.

CRITÉRIO DE CONVERGÊNCIA

Para assegurar convergência dos vetores pesos, dois requisitos devem ser satisfeitos:

1) α deve satisfazer as seguintes condições:

$0 < \alpha(t) < 1$	$\rightarrow \forall t$
$\alpha(t)$	deve ser uma função decrescente

Obs: No algoritmo, descrito a seguir, nós consideraremos $\alpha(t) = 1/t$, mas poderíamos adotar qualquer função que satisfaça a condição acima. Lippmann [LIPP 87] sugere que se adote uma função $\alpha(t)$ como sendo a Função Gaussiana (ou Função Normal).

2) O tamanho da vizinhança Nc deve satisfazer duas condições:

$1 < Nc(t) < max(lin, col)$	$\rightarrow \forall t$
$Nc(t)$	deve ser uma função decrescente

Obs : max(lin,col) é considerado o diâmetro da rede.

CRITÉRIO DE PARADA

Como $\alpha(t_k)$ deve ser decrescente durante o processo de aprendizado, decidimos atualizar esta função a cada novo padrão (modelo) apresentado à rede. Depois que todas as entradas foram apresentadas, verifica-se se um certo limiar foi atingido pela função decrescente $\alpha(t_k)$, este limiar deve ser um valor bem próximo de zero. Se este limiar foi atingido o processo de treinamento está concluído ; caso contrário, as entradas devem ser reapresentadas, até que este limiar seja atingido por $\alpha(tk)$.

Terminado o treinamento, cada entrada escolherá pela fórmula da distância, o neurônio que melhor a representar e o marcará. Cada neurônio marcado deve representar apenas uma entrada.

O resultado obtido na rede já pode então ser visualizado na matriz.

4.0.2 ALGORITMO ESTRUTURADO

```
  Sejam:
    diam - numero de neuronios que compoem o
           diametro da rede
    n_comp - numero de componentes de cada entrada
    entr - numero de modelos de entrada
    L - linha do neuronio escolhido
    C - coluna do neuronio escolhido

1.---------Inicializa pesos randomicamente--------

    Para I <- 1 Ate diam Faca
      Para J <- 1 Ate diam Faca
        Para K <- 1 Ate n_comp Faca
           Pesos[I,J,K] <- nros randomicos [0,1]

2. --------Leitura dos modelos de entrada---------

    Para I <- 1 Ate entr Faca
      Para J <- 1 Ate n_comp Faca
         Leia (Entrada[I,J]);

3. -----Escolha do neuronio com menor distancia---
   ---- entre uma entrada e os neuronios pesos----

   Seja:  cont - entrada analisada

    SomaAnt <- 0;
    Para I <- 1 Ate diam Faca
      Para J <- 1 Ate diam Faca
        Se Pesos[I,J,K] for nao marcado Entao
           Soma <- 0;
           Para K <- 1 Ate n_comp Faca
            Soma<-Soma+(Entrada[cont,K] - Pesos[I,J,K])^2*;
              Se (1o. neuronio analisado) Entao
                SomaAnt <- Soma;
              Fim Se;
              Se (SomaAnt <= Soma) Entao
                SomaAnt <- Soma;
                L <- I;  (linha deste neuronio)
```

```
                C <- J;  (coluna deste neuronio)
              Fim Se;
            Fim Para;
          Fim Se;
        Fim Para;
      Fim Para;
```

4. ----------Atualizacao da vizinhanca---------

```
   Sejam:  C - coluna do neuronio escolhido
           L - linha do neuronio escolhido

     Para I <- 1 Ate diam Faca
       Para J <- 1 Ate diam Faca
        Se ( (I <= C-raio) e (I >= C+raio) ) Entao
         Se ( (J <= L-raio)e(J >= L+raio) ) Entao
           normaliza <- 0;
           Para K <- 1 Ate n_comp Faca
                Pesos[I,J,K] <-Pesos[I,J,K]+(NETA*
                *(Entrada[cont,K]-Pesos[I,J,K]) );
                aux<-(Pesos[I,J,K] * Pesos[I,J,K] );
                normaliza <- normaliza + aux;
           Fim Para;
           normaliza <- sqrt( normaliza );
           Para K <- 1 Ate n_comp Faca
            Pesos[I,J,K]<-Pesos[I,J,K]/normaliza;
           Fim Para;
        Fim Se;
       Fim Se;
      Fim Para;
    Fim Para;
```

5.----------Treinamento - Modulo Principal ------------

```
     Leia (precisao da rede);
     tempo <- 1;
     Enquanto ( precisao < NETA )  Faca
       Para Q <- 1 Ate entr Faca
         Distancia em relacao a entrada Q;
         raio <- diam div 2;
         Enquanto ( raio >= 0 ) Faca
```

```
            Atualiza vizinhanca do neuronio;
            raio <- raio + 1;
          Fim Enquanto;
          tempo <- tempo + 1;
          NETA <- 1/tempo;
        Fim Para;
      Fim Enquanto;

6. --------------Resposta da Rede ----------------

      Para Q <- 1 Ate entr Faca
         Distancia em relacao a entrada Q;
         Neuronio escolhido marcado com valor de Q;
      Fim Para;

7. ---------Imprimir matriz resultado ------------

     Para I <- 1 Ate diam Faca
       Para J <- 1 Ate diam Faca
         Se ( Mat_Saida[I,J] esta' marcado ) Entao
            imprime valor da marca na posicao [I,J]
         Senao
            imprime 0 na posicao [I,J];
         Fim Se;
       Fim Para;
     Fim Para;
}
```

5 EXEMPLOS ILUSTRATIVOS

Exemplo 1 Utilizamos os dados fornecidos na Tabela 1 para testar o desempenho da Rede de Kohonen, uma vez que já conhecemos como estes dados são agrupados. Os resultados são apresentados a seguir, considerando-se diferentes valores para a constante NETA.

Exemplo do treinamento do modelo TIPOS DE LEITE :

Exemplo do treinamento do modelo TIPOS DE LEITE :

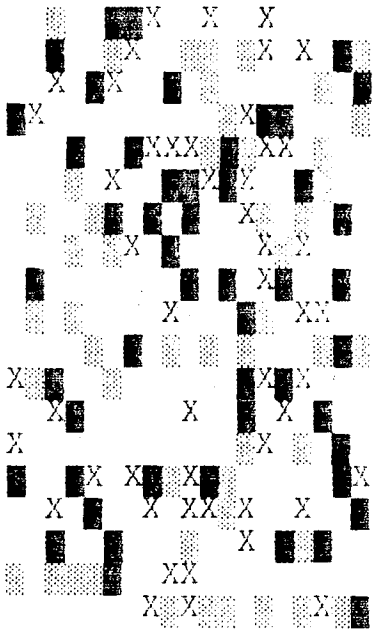
```
 0   0   0   0   0  10   1   0
 0   0   0   0   0  11   0   0
 0   0   0   0   0   6   0   0
14   0   0   0   3  13   0  12      Camada de Kohonen
 0   0   0   0   4  15   0   0           ANTES
 0   0   8   0   0   0   7   0       do treinamento
 0   0   0   0   0   0   5   0
 9   2  16   0   0   0   0   0

 0   9   0  13   0   0  15  14
 0   0   0   0  12   0   0  16      Camada de Kohonen
11   0   0   0   0   0   0   0           APOS
10   0   0   0   0   0   0   0       o treinamento
 0   0   8   0   0   0   0   0
 0   0   0   7   0   0   0   0            com
 0   0   0   0   0   0   5   3    precisao de 0.001
 0   0   0   0   6   4   2   1

 9   0   0   0  12   0  16  14
 0   0   0   0  13   0   0  15      Camada de Kohonen
 0   0   0   0   0   0   0   0           APOS
 0   0   0   0   0   0   0   0       o treinamento
 0  10   7   8   0   0   0   0
11   0   0   0   0   6   0   3            com
 0   0   0   0   0   5   0   2    precisao de 0.0001
 0   0   0   0   0   4   0   1
```

Iris setosa				*Iris versicolor*				*Iris virginica*			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5·1	3·5	1·4	0·2	7·0	3·2	4·7	1·4	6·3	3·3	6·0	2·5
4·9	3·0	1·4	0·2	6·4	3·2	4·5	1·5	5·8	2·7	5·1	1·9
4·7	3·2	1·3	0·2	6·9	3·1	4·9	1·5	7·1	3·0	5·9	2·1
4·6	3·1	1·5	0·2	5·5	2·3	4·0	1·3	6·3	2·9	5·6	1·8
5·0	3·6	1·4	0·2	6·5	2·8	4·6	1·5	6·5	3·0	5·8	2·2
5·4	3·9	1·7	0·4	5·7	2·8	4·5	1·3	7·6	3·0	6·6	2·1
4·6	3·4	1·4	0·3	6·3	3·3	4·7	1·6	4·9	2·5	4·5	1·7
5·0	3·4	1·5	0·2	4·9	2·4	3·3	1·0	7·3	2·9	6·3	1·8
4·4	2·9	1·4	0·2	6·6	2·9	4·6	1·3	6·7	2·5	5·8	1·8
4·9	3·1	1·5	0·1	5·2	2·7	3·9	1·4	7·2	3·6	6·1	2·5
5·4	3·7	1·5	0·2	5·0	2·0	3·5	1·0	6·5	3·2	5·1	2·0
4·8	3·4	1·6	0·2	5·9	3·0	4·2	1·5	6·4	2·7	5·3	1·9
4·8	3·0	1·4	0·1	6·0	2·2	4·0	1·0	6·8	3·0	5·5	2·1
4·3	3·0	1·1	0·1	6·1	2·9	4·7	1·4	5·7	2·5	5·0	2·0
5·8	4·0	1·2	0·2	5·6	2·9	3·6	1·3	5·8	2·8	5·1	2·4
5·7	4·4	1·5	0·4	6·7	3·1	4·4	1·4	6·4	3·2	5·3	2·3
5·4	3·9	1·3	0·4	5·6	3·0	4·5	1·5	6·5	3·0	5·5	1·8
5·1	3·5	1·4	0·3	5·8	2·7	4·1	1·0	7·7	3·8	6·7	2·2
5·7	3·8	1·7	0·3	6·2	2·2	4·5	1·5	7·7	2·6	6·9	2·3
5·1	3·8	1·5	0·3	5·6	2·5	3·9	1·1	6·0	2·2	5·0	1·5
5·4	3·4	1·7	0·2	5·9	3·2	4·8	1·8	6·9	3·2	5·7	2·3
5·1	3·7	1·5	0·4	6·1	2·8	4·0	1·3	5·6	2·8	4·9	2·0
4·6	3·6	1·0	0·2	6·3	2·5	4·9	1·5	7·7	2·8	6·7	2·0
5·1	3·3	1·7	0·5	6·1	2·8	4·7	1·2	6·3	2·7	4·9	1·8
4·8	3·4	1·9	0·2	6·4	2·9	4·3	1·3	6·7	3·3	5·7	2·1
5·0	3·0	1·6	0·2	6·6	3·0	4·4	1·4	7·2	3·2	6·0	1·8
5·0	3·4	1·6	0·4	6·8	2·8	4·8	1·4	6·2	2·8	4·8	1·8
5·2	3·5	1·5	0·2	6·7	3·0	5·0	1·7	6·1	3·0	4·9	1·8
5·2	3·4	1·4	0·2	6·0	2·9	4·5	1·5	6·4	2·8	5·6	2·1
4·7	3·2	1·6	0·2	5·7	2·6	3·5	1·0	7·2	3·0	5·8	1·6
4·8	3·1	1·6	0·2	5·5	2·4	3·8	1·1	7·4	2·8	6·1	1·9
5·4	3·4	1·5	0·4	5·5	2·4	3·7	1·0	7·9	3·8	6·4	2·0
5·2	4·1	1·5	0·1	5·8	2·7	3·9	1·2	6·4	2·8	5·6	2·2
5·5	4·2	1·4	0·2	6·0	2·7	5·1	1·6	6·3	2·8	5·1	1·5
4·9	3·1	1·5	0·2	5·4	3·0	4·5	1·5	6·1	2·6	5·6	1·4
5·0	3·2	1·2	0·2	6·0	3·4	4·5	1·6	7·7	3·0	6·1	2·3
5·5	3·5	1·3	0·2	6·7	3·1	4·7	1·5	6·3	3·4	5·6	2·4
4·9	3·6	1·4	0·1	6·3	2·3	4·4	1·3	6·4	3·1	5·5	1·8
4·4	3·0	1·3	0·2	5·6	3·0	4·1	1·3	6·0	3·0	4·8	1·8
5·1	3·4	1·5	0·2	5·5	2·5	4·0	1·3	6·9	3·1	5·4	2·1
5·0	3·5	1·3	0·3	5·5	2·6	4·4	1·2	6·7	3·1	5·6	2·4
4·5	2·3	1·3	0·3	6·1	3·0	4·6	1·4	6·9	3·1	5·1	2·3
4·4	3·2	1·3	0·2	5·8	2·6	4·0	1·2	5·8	2·7	5·1	1·9
5·0	3·5	1·6	0·6	5·0	2·3	3·3	1·0	6·8	3·2	5·9	2·3
5·1	3·8	1·9	0·4	5·6	2·7	4·2	1·3	6·7	3·3	5·7	2·5
4·8	3·0	1·4	0·3	5·7	3·0	4·2	1·2	6·7	3·0	5·2	2·3
5·1	3·8	1·6	0·2	5·7	2·9	4·2	1·3	6·3	2·5	5·0	1·9
4·6	3·2	1·4	0·2	6·2	2·9	4·3	1·3	6·5	3·0	5·2	2·0
5·3	3·7	1·5	0·2	5·1	2·5	3·0	1·1	6·2	3·4	5·4	2·3
5·0	3·3	1·4	0·2	5·7	2·8	4·1	1·3	5·9	3·0	5·1	1·8

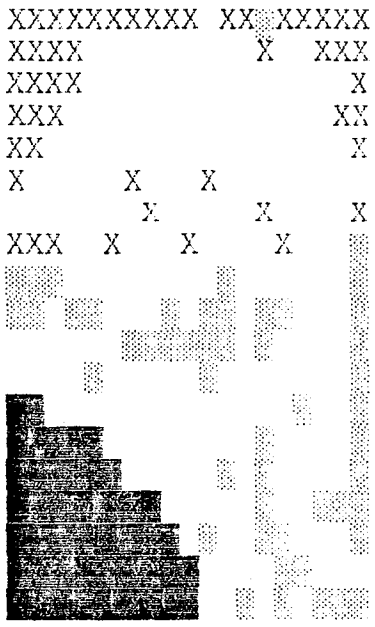
Figura 6: Três variedades de Iris

Exemplo 2 Utilizamos também a base de dados citada por Fisher (1936), conhecida como **Medidas sobre três variedades da planta Iris**. Apresentamos a classificação fornecida pela rede considerando-se diferentes valores para a constante NETA.

Exemplo do Treinamento do modelo IRIS, com Precisao de 0.001

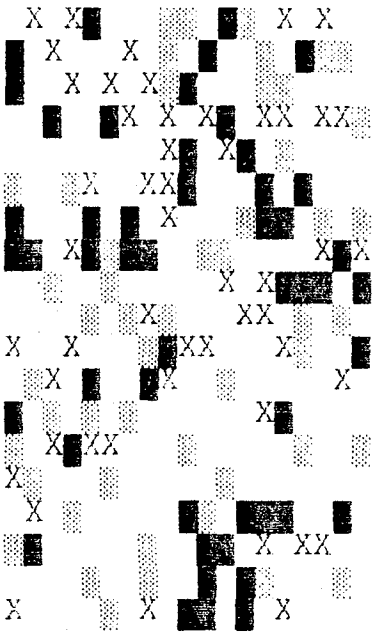


Camada de Kohonen ANTES do treinamento

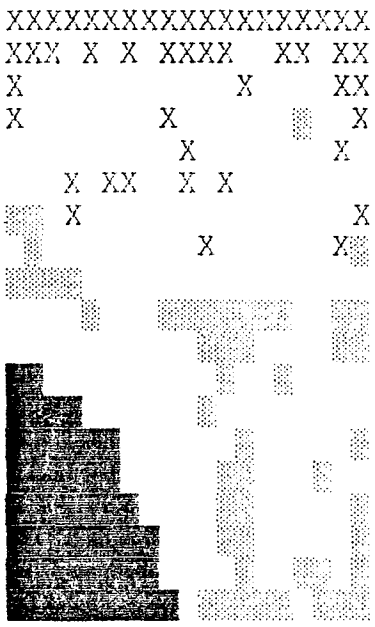


Camada de Kohonen APOS o treinamento

Exemplo do Treinamento do modelo IRIS, com precisao de 0.0001



Camada de Kohonen ANTES do treinamento



Camada de Kohonen APOS o treinamento

6 Conclusão

Este trabalho apresentou um estudo da Rede de Kohonen utilizada na Classificação de Padrões. Suas características e o seu relacionamento com o algoritmo Análise de Clusters foram discutidos. Para verificar o comportamento desta rede duas bases de dados, com resultados previamente conhecidos, foram utilizadas. Os resultados mostraram que a Rede de Kohonen é um modelo eficiente de rede neural na classificação de padrões. Apesar dos exemplos testados serem conjuntos de dados relativamente pequenos, a rede apresentou um excelente desempenho para as diferentes precisões adotadas.

A utilização da Rede de Kohonen para classificação de bases de dados de grande porte faz parte dos trabalhos futuros, uma vez que já dispomos da implementação deste modelo na Linguagem C.

Referências

[KOHO 84] KOHONEN, T., Self-Organization and Associative Memory, Springer-Verlag, Berlin, 1984.

[KOHO 88] KOHONEN, T., "An introduction to neural computing", *Neural Networks*, vol. 1, no. 1, pp. 3-16, 1988.

[GRAY 84] R. M. GRAY, "Vector Quantization", *IEEE ASSP Magazine*, pp. 4-29, abril 1984.

[CAUD 87] CAUDILL, M., "Neural networks primer - Part IV", *AI Expert*, dez/87, fev/88, jun/88, ago/88.

[HART 75] HARTIGAN, J. A., "Clustering Algorithms", John Wiley & Sons, New York, 1975.

[JENS 91] JENSEN, O. B.; ROHDE, T., "Automatic Speech Recognition e Neural Networks", DAIMI IR-101, Computer Science Department, Aarhus University - Denmark, abril 1991.

[GEVA 91] GEVA, S.; SITTE, J., "Adaptive Nearest Pattern Classification", IEEE Trans. on Neural Networks, vol. 2, no. 2, março 1991.

[KANG 91] KANGAS, J. A.; KOHONEN, T.; LAAKSONEN, J. T., "Variants of Self-Organizing Maps", IEEE Trans. on Neural Networks, vol. 1, no. 1, março 1990.

[LIPP 87] LIPPMANN, R. P., "An Introduction to Computing with Neural Nets", IEEE ASSP, vol. 4, no. 2, abril 1987.

[DUDA 73] DUDA, R. O.; HART, P. E., "Pattern Classification and Scene Analysis", NY: John Wiley and Sons, 1973.

[SEB 62] "Decision Making Processes in Pattern Recognition", McMillan, 1962.