

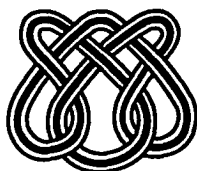
UNIVERSIDADE DE SÃO PAULO

**Estudo de Modelos de Redes Neurais
para a Indexação e Recuperação de Dados**

**Ernesto Cuadros Vargas
Roseli Aparecida Francelin Romero**

N^o 44

NOTAS DIDÁTICAS



Instituto de Ciências Matemáticas de São Carlos

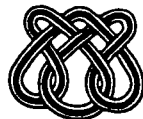
UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2585

**Estudo de Modelos de Redes Neurais
para a Indexação e Recuperação de Dados**

**Ernesto Cuadros Vargas
Roseli Aparecida Francelin Romero**

Nº 44

NOTAS DIDÁTICAS



São Carlos – SP
Ago./2000

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e
Estatística

Estudo de Modelos de Redes Neurais
para a Indexação e Recuperação de Dados ¹

Ernesto Cuadros Vargas

Roseli Aparecida Francelin Romero

Departamento de Ciências de Computação e Estatística

Instituto de Ciências da Computação e Estatística

Universidade de São Paulo – Campus de São Carlos

Caixa Postal 668

13560-970 – São Carlos – SP

Agosto de 2000

¹Este trabalho foi desenvolvido com o apoio financeiro da FAPESP. Nro Processo: 99/11835-7

Sumário

1	Introdução	1
2	Motivação	2
3	Objetivos	2
4	Descrição do Módulo Inteligente Proposto	2
5	Modelos de Redes Neurais Aplicáveis	4
5.1	Teoria da Ressonância Adaptativa	4
5.1.1	Introdução	4
5.1.2	Arquitetura Básica de ART	5
5.1.3	Treinamento	7
5.1.4	Aprendizado em ART1	7
5.1.5	Recuperação de Informação	9
5.1.6	Exemplo	9
5.1.7	Algoritmo de Aprendizado de ART1	11
5.1.8	Modelos ART	12
5.2	Mapas auto-organizáveis	13
5.2.1	Introdução	13
5.2.2	Processo de Aprendizado em um SOM	13
5.3	Growing Neural Gas	17
5.3.1	Introdução	17
5.3.2	Exemplos de reconhecimento com uma rede GNG	20
6	Proposta	27
6.1	Proposta 1	27
6.2	Proposta 2	28
6.3	Proposta 3	28
7	Conclusões e Trabalhos Futuros	28

Lista de Figuras

1	Arquitetura básica de ART.	5
2	Os pesos na arquitetura de ART.	6
3	Passo 1.	9
4	Passo 2.	10
5	Passo 3.	11
6	Arquitetura Básica de um Mapa Auto-Organizável.	14
7	Vizinhança do neurônio ganhador em um mapa Auto-Organizável.	15
8	Correspondência entre a distância física dos neurônios de uma rede de Kohonen e o grau de semelhança dos padrões por eles reconhecidos.	16
9	Clusters gerados por uma rede de Kohonen.	17
10	Objetos volumétricos utilizados para os experimentos (a) Toroide (b) Esfera.	21
11	Objetos volumétricos utilizados para os experimentos (c) Cubo (d) Cone.	21
12	Evolução do treino de uma rede GNG durante o reconhecimento de um toroide.	22
13	Evolução do treino de uma rede GNG durante o reconhecimento de um cubo.	23
14	Treinamento de uma rede GNG para o reconhecimento de volumes com forma de cubo depois de 3000 sinais apresentados.	23
15	Treinamento de uma rede GNG para o reconhecimento de volumes com forma de cubo depois de 3000 sinais apresentados.	24
16	Treinamento de uma rede GNG para o reconhecimento de volumes com intersecções depois de 3000 sinais apresentados.	24
17	Treinamento de uma rede GNG para o reconhecimento de volumes com intersecções depois de 20000 sinais apresentados.	25
18	Treinamento de uma rede GNG para o reconhecimento de volumes gerados de forma aleatória.	26
19	Treinamento de uma rede GNG para o reconhecimento de dois planos com intersecção (a) Com os Planos (b) Sem os planos.	26
20	Modelo Proposto	27

Lista de Tabelas

1	Resultado da primeira aplicação da regra 2/3.	10
2	Resultado da segunda aplicação da regra 2/3.	10

Resumó

Este trabalho apresenta o estado da arte em relação ao processo de Indexação e Recuperação de Informação em Bases de Dados Multimídia. Os Sistemas de Gerenciamento de Bases de Dados (SGBD) existentes são muito sofisticados, eficientes e rápidos no processamento e recuperação de informação envolvendo dados dos tipos convencionais (números, strings, etc), mas, existem algumas limitações dos mesmos quando o objetivo é procurar informação em campos com dados multimídia (vídeo, som, etc). Considerando os problemas existentes com os SGBD propõe-se a utilização de Redés Neurais como uma alternativa para realizar a Indexação e Recuperação de Informação em Bases de Dados Multimídia.

1 Introdução

Uma das primeiras preocupações com o acesso a uma Base de Dados foi criar um mecanismo de acesso direto às informações armazenadas nela. Para isso foram criadas técnicas como a ISAM (*Indexed Sequential Access Methods*). Estas técnicas, basicamente, foram utilizadas para a Indexação de pequenos números e cadeias de caracteres, sendo que os gerenciadores de bases de dados comerciais existentes os utilizam quase que exclusivamente.

Atualmente, os Sistemas de Gerenciamento de Bases de Dados têm sido cada vez mais utilizados para o armazenamento e recuperação de dados que não são apenas numéricos ou textuais. Entre esses novos tipos de dados considerados não tradicionais, encontram-se, por exemplo: audio digitalizado, partituras musicais [Beker 97], hipertextos, imagens em geral [Santos 97], séries temporais, estruturas moleculares, cadeias de DNA, etc.. Esse tipo de informação gera conjuntos de dados tipicamente multidimensionais ou adimensionais.

O tipo de estrutura utilizado pelos métodos ISAM não pode ser aplicado com esse tipo de informação. Para o tratamento de dados do tipo n-dimensional existem diversas Estruturas de Dados Espaciais as quais permitem um melhor aproveitamento da natureza multi-dimensional dos dados. Essas técnicas também são conhecidas como Métodos de Acesso Espacial - SAM (*Spatial Access Methods*). O trabalho pioneiro nessa área foi o de R-Trees [Guttman 84] e existem diversas variações desse primeiro trabalho tais como, em [Sellis 87], [Beckmann 90], [Hellerstein 95], [Papadias 95] e [Gaede 98]. Este último é um resumo de muitos Métodos de Acesso Espacial em Bases de Dados.

Com dados em espaços adimensionais, nenhuma dessa estruturas poderia ser aplicada. Existe pouca bibliografia referente ao tratamento de espaços métricos (que engloba tanto dados espaciais com dimensão definida quanto dados adimensionais). Entre as principais estruturas abordadas na literatura estão: *Query Fixed Trees* [Baeza-Yates 94], *mvp-trees (multi-vantage point tree)* [Bozcaya 97], *vp-tree (vantage-point tree)* [Chiueh 94], GNAT [Brin 95], M-Tree [Ciaccia 97]. A estrutura proposta por Ciaccia em [Ciaccia 97] apresenta vantagens em relação às outras, ela é dinâmica, escalável, e mantém a eficiência quando armazenada em memória secundária. No entanto os testes preliminares realizados mostraram que de fato ela se comporta bem para dados simples (como conjuntos de dados

em duas dimensões), mas seu desempenho cai muito rápido com o aumento do número de dimensões como é o caso de dados multimídia. Um outro trabalho relacionado ao mapeamento de dados multi-dimensionais é o algoritmo FastMap [Faloutsos 95]. Algumas destas estruturas serão apresentadas com mais detalhe neste trabalho.

O presente trabalho está organizado como segue: na Seção 2 uma motivação ao problema é colocada. Os objetivos são apresentados na Seção 3. Na Seção 4 é apresentada uma descrição do Módulo Inteligente Proposto. Alguns modelos de Redes Neurais apropriadas para resolver este problema são apresentados na Seção 5. Na Seção 6 é colocada a proposta deste trabalho. Finalmente, na Seção 7 são apresentadas as conclusões deste trabalho e a bibliografia utilizada.

2 Motivação

Existe uma série de perguntas que envolvem um custo computacional muito alto se resolvidas com técnicas convencionais, por exemplo, dado um conjunto de cidades junto com suas coordenadas em um mapa, “Quais são as duas cidades cuja distância entre elas seja a menor do país?”. Perguntas aparentemente simples como essa envolvem um custo computacional muito alto (atualmente tem um custo de N^2); ainda não existe uma técnica adequada para responder eficientemente uma pergunta desse tipo.

Um outro problema, ainda mais importante, é que o conceito de “similaridade” ainda não está incorporado nos SGBD, a incorporação desse conceito permitiria responder consultas como: “Quais são as imagens que são no mínimo 90% parecidas com uma imagem dada?”.

Uma das características mais conhecidas das Redes Neurais é a sua Tolerância a Falhas, isto é, a capacidade de associar um padrão que possui ruído à classe com que ele tem maior grau de similaridade, incorporando assim, o conceito de similaridade no processo de classificação [Rumelhart 86].

3 Objetivos

O objetivo principal deste trabalho é investigar alguns modelos de Redes Neurais que possam ser utilizados no processo de Indexação e Recuperação de Dados. Isto é, utilizar Redes Neurais para responder a consultas em relação à similaridade entre os elementos do conjunto de dados multi-dimensionais ou adimensionais, com um custo computacional menor do que aquele obtido pelos Métodos de Acesso Espacial.

4 Descrição do Módulo Inteligente Proposto

Inicialmente, os dados devem ser transformados em uma representação adequada para trabalhar com RNAs. O pré-processamento do conjunto de dados envolve a Normalização

e a Redução do número de atributos com o intuito de diminuir o tamanho da RNA, assim como, o tempo requerido para a incorporação de novos padrões. Para diminuir o número de atributos podem ser utilizadas técnicas de Aprendizado de Máquina, tais como, Árvores de Decisão (*Decision Trees*) [Mitchell 97] ou técnicas estatísticas, tais como, Análise da Componente Principal (PCA, *Principal Component Analysis*) [Haykin 99].

Como não se conhece quantos aglomerados existem nos dados, o modelo de rede a ser utilizado deverá utilizar o paradigma Não Supervisionado. Partindo desse ponto, pode-se utilizar uma Rede Neural que possa realizar o mapeamento de um vetor multi-dimensional para uma superfície tridimensional. Nessa superfície, aqueles padrões que possuam maior grau de similaridade serão classificados mais perto um do outro. Um mapeamento desse tipo pode ser realizado utilizando uma rede do tipo Auto-Organizável tal como a rede de Kohonen [Kohonen 84].

As redes de Kohonen têm a capacidade de detectar aglomerados (*clusters*) nos dados incorporando assim, o conceito de "similaridade". No caso de um Banco de Dados, novos padrões podem ser inseridos a qualquer momento, não sendo possível determinar uma topologia fixa, isso dificulta a utilização de uma rede de Kohonen da forma original.

Considerando esse requisito, a rede escolhida deve ser, além de auto-organizável, construtiva. O modelo básico da Rede de Kohonen (ver Seção 5.2) não é construtivo, porém, um outro grupo de Redes Neurais, conhecidas como Construtivas (RNCs), têm a capacidade de mudar a sua arquitetura ao longo do processo de treinamento, gerando assim uma rede diferente para cada problema [Parekh 95] Isto é desejável pois quando não se conhece a relação existente entre os dados nem o número de agrupamentos. O fato de não conhecer o número de agrupamentos significa que a rede, além de incorporar técnicas construtivas, deverá trabalhar sob o paradigma Não Supervisionado. Existem algumas técnicas que cumprem com os requisitos antes mencionados, por exemplo, *Growing Cell Structures* [Fritzke 1994a] e a Rede *Growing Neural Gas* (GNG) [Fritzke 1995a], esta última será explicada com maior detalhe na Seção 5.3.

Uma alternativa para diminuir a carga dessa rede poderia ser a realização do processo de "clusterização" como um passo prévio. Para atingir este objetivo deve-se considerar é aproveitar a natureza da aplicação. Dependendo disso podem ser utilizados alguns padrões como base. Um padrão base poderia ser, por exemplo, uma imagem típica de uma determinada patologia de um órgão do corpo humano obtida através de um Tomógrafo de Ressonância Magnética (TRM). Se os dados fossem fotos de pessoas, as raças poderiam ser consideradas como as bases.

Um tipo de RNAs que têm a capacidade de detectar novos grupos dinamicamente sem perder o conhecimento relacionado aos padrões já armazenados são as redes da família ART²[Carpenter 87a]. As redes ART serão apresentadas com mais detalhe na Seção 5.1.

Na próxima seção, alguns modelos de RNA, que acreditamos ser de utilidade para auxiliar na solução do problema proposto, serão descritos em detalhes.

²As siglas correspondem ao nome em inglês: *Adaptive Resonance Theory*.

5 Modelos de Redes Neurais Aplicáveis

5.1 Teoria da Ressonância Adaptativa

5.1.1 Introdução

Uma das características mais interessantes da mente humana é a sua capacidade de continuar aprendendo novas coisas sem esquecer aquilo que já foi aprendido previamente. Na maioria das vezes, uma RNA esquece um pouco a informação já aprendida quando da apresentação de novos padrões.

O processo típico de utilização de uma RNA envolve a codificação da informação em uma forma adequada, e a apresentação dos mesmos à RNA para que ela possa ajustar os seus pesos. Uma vez finalizado esse processo, a rede estará pronta para passar à fase de produção não sendo possível a modificação dos pesos nessa etapa. Esse cenário é aceitável sempre que o problema em questão tenha limites bem definidos e seja estável. Em muitos casos da vida real o entorno não está bem definido e não é estável.

Considere o exemplo onde a rede foi treinada para reconhecer diferentes tipos de sons. Isto pode ser uma tarefa muito custosa devido aos cálculos envolvidos. Contudo, se depois de finalizado este processo de treinamento, fosse apresentado à rede um novo som o processo teria que ser reinicializado mais uma vez. Também não seria recomendável treinar a RNA só com aquele novo padrão porque corre-se o risco de que a RNA decore só o novo padrão e esqueça tudo aquilo que já tinha aprendido. Tudo isso que está sendo explicado nesta seção corresponde ao que Stephen Grossberg e Gail Carpenter chamaram do **Dilema de Estabilidade e Plasticidade** [Carpenter 87a].

Este dilema pode ser explicado através de diversas perguntas [Carpenter 88] tais como:

- Como um sistema (capaz de aprender) pode continuar sendo adaptável (plástico) em resposta a uma entrada significativa e, ao mesmo tempo, se manter estável em resposta a entradas irrelevantes ?
- Como o sistema será capaz de alternar entre os modos plástico e estável ?
- Como o sistema poderá reter as informações aprendidas anteriormente e continuar aprendendo dados novos ?

Em resposta a estas perguntas, a equipe de Grossberg, Carpenter desenvolveu a **Teoria da Ressonância Adaptativa (ART)** que pretende responder as perguntas antes formuladas. As Redes ART são uma variação dos mecanismos do aprendizado competitivo apresentado na Seção 5.2.

Quando a natureza dos dados não é estatisticamente estacionário ou quando surgem novos tipos de padrões podem surgir alguns problemas. tais como, dois neurônios respondendo ao mesmo tempo por padrões de classes diferentes [Grossberg 82].

Uma alternativa para resolver o problema da plasticidade e elasticidade consiste na adição à rede de um mecanismo de realimentação localizado entre a camada de entrada e a camada competitiva. Este mecanismo facilita o aprendizado de novos padrões sem esquecer a informação já armazenada, a alternância automática entre os modos plástico e estável.

Existem dois tipos de arquiteturas de redes ART adequadas para a classificação de padrões, a **ART1** e a **ART2**. A diferença está nos tipos de dados com que elas trabalham. a rede ART1 é apropriada para entradas binárias e a ART2 para entradas analógicas e escala de cinzas. A seguir serão apresentadas as características básicas das redes ART.

5.1.2 Arquitetura Básica de ART

A arquitetura básica de ART envolve duas camadas de neurônios. A camada de entrada, também conhecida como F_1 , para o processamento dos dados de entrada; e a camada de saída, também conhecida como F_2 , contendo os neurônios que representam os *clusters* (Figura 1).

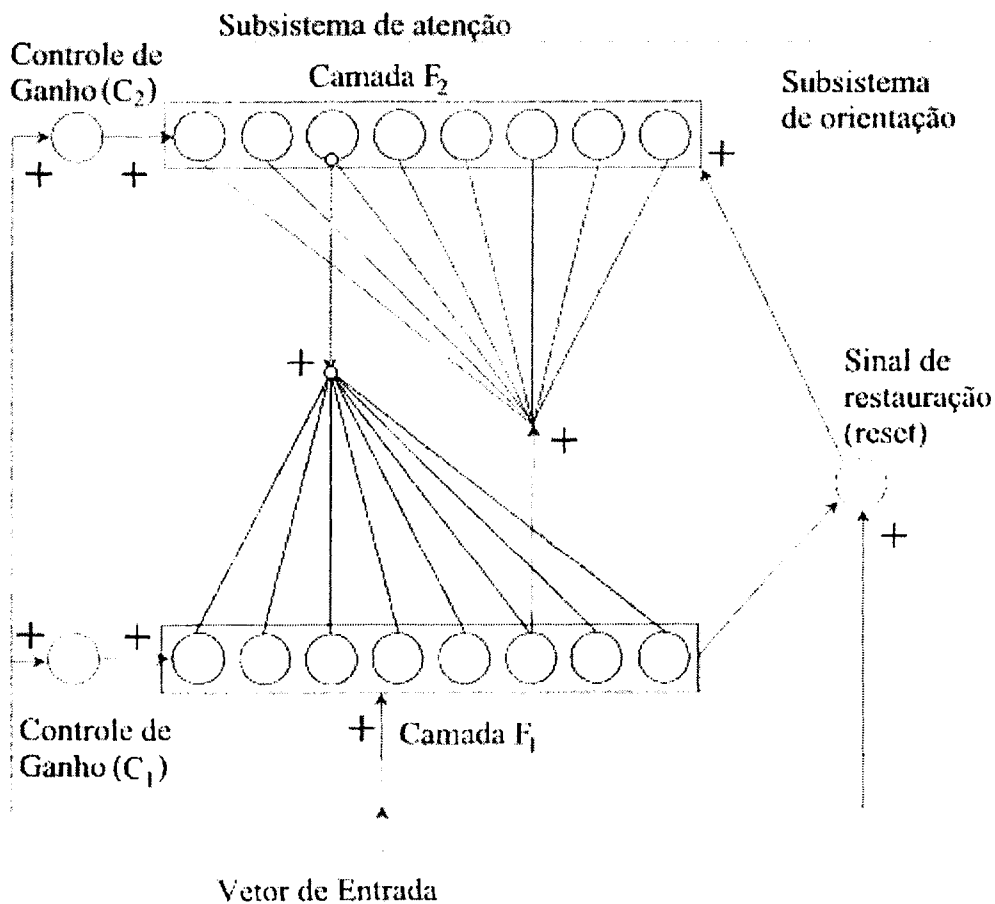


Figura 1: Arquitetura básica de ART.

Estas duas camadas estão conectadas através de dois vetores de pesos. Um vetor de pesos

feedforward, representado pela letra B , que assume valores reais e segue da camada de entrada para a camada de saída; e outro vetor de pesos *feedback*, representado pela letra T , que assume valores binários e vai da camada de saída para a camada de entrada. Na Figura 2, o neurônio X_i da camada de entrada está conectado ao neurônio Y_j da camada de saída através do peso b_{ij} . Igualmente, o neurônio Y_j da camada de saída está conectado ao neurônio X_i da camada de entrada através do peso t_{ji} . Existem também pesos entre os neurônios da camada de saída que atuam como inibidores laterais.

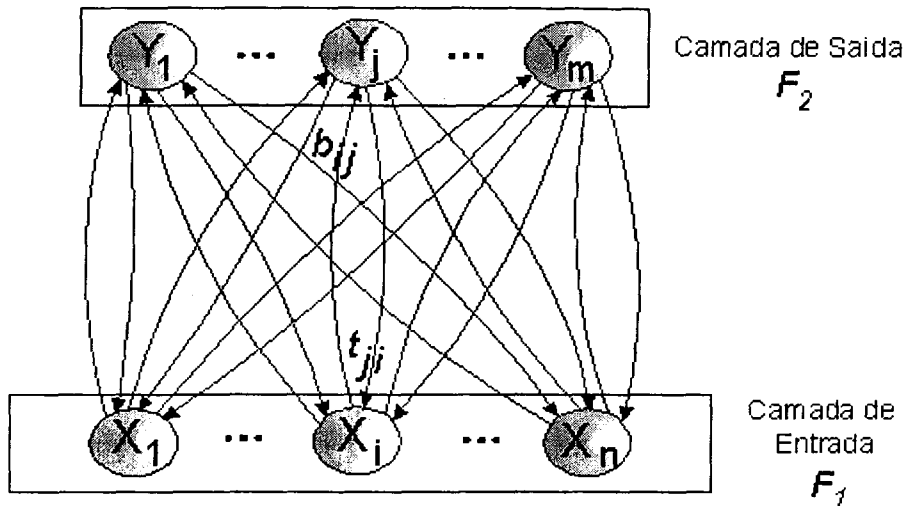


Figura 2: Os pesos na arquitetura de ART.

Os pesos *feedforward* e *feedback* são também conhecidos como *filtros adaptativos* porque mudam dinamicamente para possibilitar o aprendizado de novos padrões de entrada. O vetor de pesos *feedforward* associado a cada neurônio da camada de saída pode ser visto como um protótipo ou exemplar dos padrões que fazem parte do *cluster* representados por este neurônio.

Para cada camada da rede ART há uma unidade externa de controle. Estas unidades controlam o fluxo de dados através das camadas em cada fase de um ciclo de treinamento. Elas são representadas pelas letras C_1 e C_2 - ver Figura 1. C_1 determina o fluxo de dados para a camada de entrada. Assume valor 1 se uma entrada válida for apresentada à rede e valor 0 se algum neurônio da camada de saída está ativo. C_2 tem duas funções: determina o fluxo de dados para a camada de saída e habilita ou desabilita os neurônios da camada de saída. Assume valor 1 se uma entrada válida for apresentada à rede e 0 após desabilitar os neurônios da camada de saída e resetar seus estados de ativação para 0.

Entre a camada de entrada e a camada de saída há também uma unidade de *reset*, que é responsável por comparar as entradas da rede com um *threshold* de vigilância (ρ). Este *threshold* de vigilância determina se uma nova classe de padrões (*cluster*) poderá ser criada para um padrão de entrada.

5.1.3 Treinamento

O algoritmo de aprendizado da rede ART é não supervisionado e está apto a entrar no modo de treinamento a qualquer momento possibilitando que a rede incorpore conhecimento novo. Esta característica resolve o problema plasticidade-estabilidade comentado anteriormente.

Há dois tipos de treinamento para ART: aprendizado rápido e aprendizado lento. No aprendizado rápido os pesos *feedforward* são setados para seus valores ótimos em poucos ciclos, geralmente em apenas um ciclo de treinamento. No aprendizado lento os pesos são ajustados lentamente em vários ciclos de treinamento, significando que maior atenção será dada para encontrar as características relevantes dos padrões de entradas para determinar sua classificação.

A rede ART é muito sensível a variações em seus parâmetros durante o treinamento. O parâmetro mais crítico é o *threshold* de vigilância (ρ) que controla a resolução do processo de classificação. Se ρ assume um valor baixo (< 0.4), a resolução do processo de classificação será baixa, criando poucas classes. Se ρ for setado com um valor alto, pequenas variações nos padrões de entrada levarão a criação de novas classes.

5.1.4 Aprendizado em ART1

O processo de aprendizado em ART1 envolve algumas fases, tais como: inicialização dos parâmetros e pesos da rede, reconhecimento, comparação e busca. Cada uma destas fases será descrita a seguir.

- Inicialização

Na fase de inicialização, os vetores de pesos B e T devem ser inicializados. Todos os pesos *feedback* (T) são inicializados com o valor 1 binário, indicando que todo neurônio da camada de saída está inicialmente conectado a todo neurônio de entrada através de uma conexão *feedback*. Os pesos iniciais do vetor de pesos *feedforward* (B) são determinados pela equação:

$$b_{ij} = \frac{1}{1 + n}, \quad (1)$$

no qual n é o número de neurônios de entrada da rede.

O *threshold* de vigilância (ρ) é também inicializado, assumindo um valor dentro do intervalo $0 < \rho < 1$.

Reconhecimento

Cada neurônio da camada de entrada recebe sinais de três fontes: um sinal do vetor de entrada, um sinal *feedback* da camada de saída e um sinal da unidade de Controle-1 (C_1). Um neurônio da camada de entrada somente será ativado se receber pelo menos dois

sinais excitatórios. caso contrário, a ativação não ocorre. Esta regra é denominada *regra dois-terços* (2/3).

Na fase de reconhecimento, o vetor de entrada é enviado à camada de entrada, onde será aplicada a regra 2/3. Em seguida, o vetor de entrada é comparado com todos os exemplares (protótipos) armazenados nos vetores de pesos T dos neurônios da camada de saída para encontrar o melhor *matching*. Esta comparação é realizada calculando-se o produto interno entre o vetor de entrada resultante da aplicação da regra 2/3 e o vetor de peso *feedforward* de cada neurônio da camada de saída, dada por:

$$y_j = \sum_i b_{ij}x_i, \quad (2)$$

onde y_j é o somatório do produto correspondente ao neurônio de saída j , entre cada componente b_{ij} do vetor de pesos B e cada componente x_i do vetor entrada.

O neurônio da camada de saída cujo produto apresenta o maior valor é selecionado como provável *cluster* para armazenar o novo padrão. Este neurônio envia à camada de entrada seu protótipo ou exemplar, ou seja, seu vetor de peso T . A inibição lateral faz com que somente o neurônio vencedor da camada de saída fique ativo.

Comparação

Como foi descrito anteriormente, cada neurônio da camada de entrada recebe três sinais: do vetor de entrada; do vetor do exemplar do neurônio de saída selecionado, vindo da fase de reconhecimento; e da unidade de Controle-1 (C_1), que neste instante é *zero* porque um neurônio foi ativado na fase de reconhecimento. A regra 2/3 é aplicada novamente resultando na ativação ou não de cada neurônio da camada de entrada.

Como o sinal da unidade de Controle-1 é *zero*, há uma operação *AND* entre o vetor do exemplar do neurônio selecionado e o vetor de entrada para produzir um novo vetor, chamado de *vetor de comparação*. Este vetor é enviado para o mecanismo de *reset* juntamente com o vetor de entrada.

O mecanismo de *reset* é responsável por testar a similaridade entre o vetor de entrada e o vetor de comparação. O teste é feito calculando a razão P , através da Equação (3), entre o número de *uns* de ambos os vetores, de entrada e de comparação, e comparando o resultado desta razão com o *threshold* de vigilância (ρ) utilizando as Equações (4) e (5).

$$P = \frac{\|\sum x_i t_{ij}\|}{\|\sum x_i\|} \quad (3)$$

$$P > \rho \quad (4)$$

$$P < \rho \quad (5)$$

Se P for maior que ρ , então o neurônio da camada de saída ativado incluirá o vetor de entrada em seu *cluster*, caso contrário, outro neurônio da camada de saída deverá ser encontrado para representar o vetor de entrada e a rede entra na fase de busca.

5.1.5 Recuperação de Informação

Durante esta fase, a rede tenta encontrar um outro neurônio da camada de saída para representar o vetor de entrada corrente.

O neurônio da camada de saída rejeitado na fase de comparação anterior é impossibilitado de competir para representar o padrão de entrada corrente. O neurônio vencedor é desabilitado, setando seu valor de saída para 0, indicando que ele não poderá mais ser selecionado para o padrão de entrada corrente. O vetor de entrada é representado e a rede entra novamente na fase de comparação, que termina com o teste no mecanismo de *reset* para determinar a similaridade entre o novo exemplar escolhido e o vetor de entrada corrente. Este processo é repetido, desabilitando neurônios da camada de saída, até encontrar um neurônio de saída que melhor “case” com o vetor de entrada corrente, dentro dos limites do *threshold* de vigilância (ρ). Se nenhum neurônio de saída for encontrado, o vetor de entrada é então considerado de uma classe desconhecida e será alocado um novo neurônio de saída “livre” (que não esteja associado ainda a nenhum *cluster*) para representá-lo.

5.1.6 Exemplo

Um exemplo será descrito para melhor compreensão do funcionamento do aprendizado da rede ART1. Neste exemplo, há três neurônios na camada de entrada e inicialmente um neurônio na camada de saída. C_1 e C_2 são inicializados com zero, o parâmetro de vigilância é inicializado para 0.8, os pesos *feedback* com 1 e os pesos *feedforward* de acordo com a Equação (1).

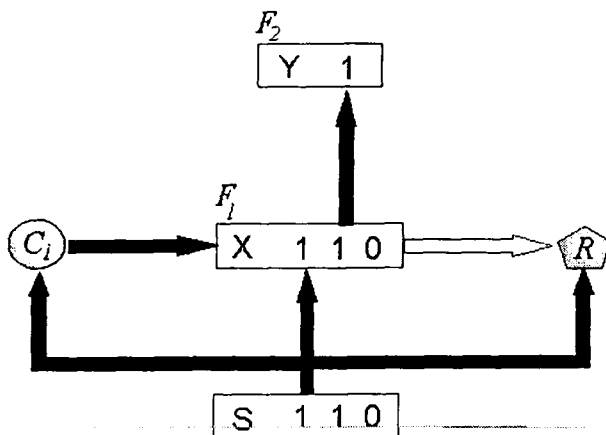


Figura 3: Passo 1.

O vetor de entrada S é aplicado à camada de entrada (F_1) (Figura 3). A regra 2/3 determina a resposta da camada entrada para o padrão de entrada. Neste ponto, a camada de entrada recebe dois sinais excitatórios; um sinal do vetor de entrada S e um sinal da unidade de Controle-1 (C_1), significando que há uma entrada válida para a rede. Os sinais excitatórios correspondem a 1 e os sinais inibitórios correspondem a 0. Na Tabela 1 é mostrado o resultado, vetor X , do efeito destes sinais após aplicar a regra 2/3.

Entrada (S)	C_1	Resultado (X)
1	1	1
1	1	1
0	1	0

Tabela 1: Resultado da primeira aplicação da regra 2/3.

O vetor X é comparado com os pesos *feedforward* de cada neurônio da camada de saída. Isto é feito calculando o produto interno entre estes vetores pela Equação (2).

O neurônio selecionado como vencedor (neurônio com maior produto interno) envia seu exemplar armazenado para a camada de entrada (F_1) e inibe C_1 , setando seu valor para 0. A camada de entrada recebe três sinais: do vetor de entrada, do neurônio vencedor da camada de saída e de C_1 (Figura 4). O resultado depois de aplicada a regra 2/3 pode ser visto na Tabela 2.

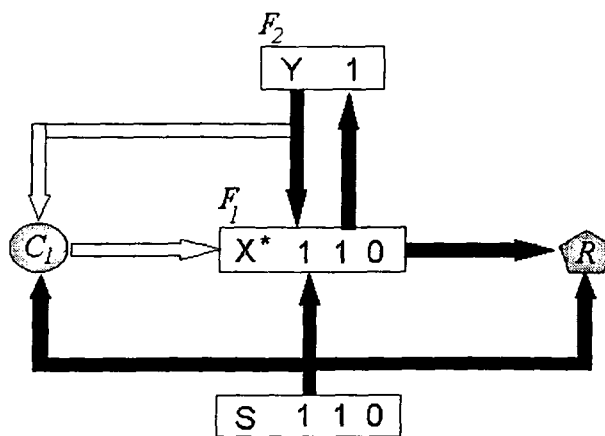


Figura 4: Passo 2.

Entrada (S)	Pesos Feedback	C_1	Resultado (X^*)
1	1	0	1
1	1	0	1
0	1	0	0

Tabela 2: Resultado da segunda aplicação da regra 2/3.

O vetor de comparação X^* e o vetor de entrada S são ambos enviados ao mecanismo de *reset* para o teste de vigilância. A razão (P) entre estes dois vetores é calculada pela

Equação (3), sendo neste caso 1 porque os vetores são idênticos. Esta razão é comparada ao *threshold* de vigilância (ρ), adotado aqui como sendo maior que 0.8, então o vetor de entrada foi classificado corretamente. Em seguida, os vetores de pesos do neurônio vencedor são ajustados para incorporar as características do vetor de entrada corrente. Isto é feito pelas seguintes equações:

$$T_{ij(novo)} = T_{ij(velho)} \wedge X_i \quad (6)$$

$$B_{ij(novo)} = \frac{L * X_i}{L - 1 + \sum T_{ij(velho)} \wedge X_i} \quad (7)$$

onde \wedge é o operador lógico AND e L é um parâmetro ($L > 1$) utilizado na atualização dos pesos *bottom-up*.

Quando a razão (P) é menor que ρ , R dispara e inibe o neurônio ativado da camada de saída (Figura 5). O padrão de entrada será atribuído a outro neurônio da camada de saída. Após a aplicação do próximo vetor de entrada à rede, a unidade de Controle-2 (C_2) habilita todos os neurônios inibidos da camada de saída para que eles voltem a competir entre si.

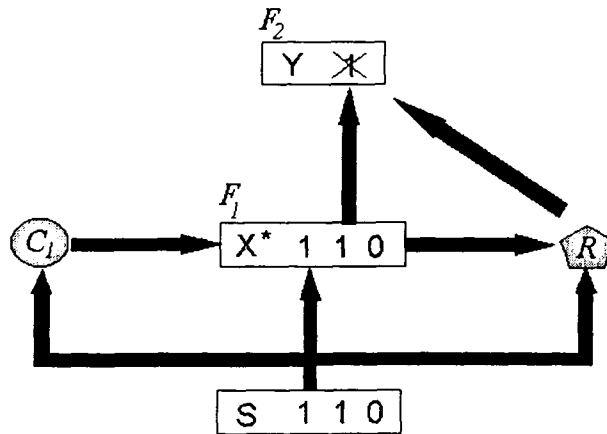


Figura 5: Passo 3.

5.1.7 Algoritmo de Aprendizado de ART1

O algoritmo de aprendizado do modelo ART1 é sumarizado abaixo:

1. Inicializar pesos e parâmetros
2. Repita
 - 2.1 Para cada padrão de treinamento S
 - 2.2 Definir neurônio vencedor
 - 2.3 Comparar prototipo do vencedor com a entrada

2.4 Se comparação $> threshold$ de vigilância

Então atualizar os pesos do neurônio vencedor

Senão desabilitar neurônio

Se ainda existir neurônio não analisado

Então voltar a 2.2

Senão alocar novo neurônio ao padrão de entrada

Ate conjunto de prototipos não mudar

5.1.8 Modelos ART

Desde o surgimento da Teoria da Ressonância Adaptativa várias outras redes baseadas em ART foram desenvolvidas. O primeiro modelo da família ART a ser desenvolvida foi a rede ART1 [Carpenter 87a]. Esta rede responde à sequências arbitrárias de padrões de entrada binários.

Entretanto, vários problemas do mundo real utilizam dados analógicos. Em vista disto, foi desenvolvida a rede ART2 [Carpenter 87b]. Esta rede reconhece categorias (classes) em resposta a sequências arbitrárias de padrões de entrada analógicos (valores contínuos), tão bem como para sequências de padrões binários. A diferença entre ART1 e ART2 está na camada F_1 , que precisa ser mais complexa para o pré-processamento necessário para acomodar os padrões de entrada de valores contínuos.

A rede ARTMAP [Carpenter 91a] é outra rede baseada na rede ART que realiza aprendizado incremental supervisionado para reconhecer categorias em resposta a vetores de entrada apresentados à rede. Esta classe é composta por duas redes ART1 (ART_a e ART_b). Durante o treinamento, o módulo ART_a recebe um padrão de entrada $a^{(p)}$ e o módulo ART_b recebe um padrão de entrada $b^{(p)}$ que é a predição correta de $a^{(p)}$, ou seja, $b^{(p)}$ é a resposta desejada do padrão $a^{(p)}$. Estes dois módulos são conectados através de um mapeamento entre as camadas de saída (F_2) de ART_a e ART_b .

A primeira rede ARTMAP foi utilizada para classificar entradas que eram vetores de valores binários representando a presença ou ausência de cada característica. Uma rede ARTMAP mais geral foi então projetada para aprender a classificar entradas que eram vetores de valores fuzzy entre 0 e 1 indicando o grau de cada característica apresentada. Esta generalização é realizada por substituir os módulos de ART1 de ARTMAP por módulos de outra rede baseada em ART, a rede Fuzzy ART [Carpenter 91b]. A nova rede é conhecida como rede Fuzzy ARTMAP [Carpenter 92].

Ao longo do tempo foram surgindo mais variações, não tão conhecidas ainda, baseadas na rede ART: CALM [Murre 89], ART3 [Carpenter 90], ART2a [Carpenter 91c], SMART [Bartfai 94] e HART [Bartfai 95].

O presente trabalho propõe a utilização do modelo ART2 (por aceitar entradas contínuas) como será descrito na seção 6.

5.2 Mapas auto-organizáveis

5.2.1 Introdução

O cérebro humano talvez seja a estrutura mais interessante da fisiologia humana. Mesmo sendo muito complexo do ponto de vista microscópico, ele tem uma estrutura uniforme a escala macroscópica. Os centros responsáveis por atividades tais como o pensamento, visão, audição, etc. sempre se desenvolvem na mesma posição em todos os seres humanos. As áreas individuais apresentam uma ordenação lógica em relação a sua funcionalidade. Um exemplo conhecido desta ordenação é o mapa tonotópico das regiões auditivas. Neste mapa neurônios próximos entre si respondem a frequências similares de sons. Também é conhecida a existência do mapa somatotópico que é o mapa dos nervos motores responsáveis por cada parte do corpo humano. Neste tipo de mapa regiões fisicamente próximas serão responsáveis por membros fisicamente próximos do corpo.

Regiões tais como o mapa Tonotópico e o mapa Somatotópico são também conhecidas como: **correspondências ordenadas de características**. Como definição simplificada pode-se dizer que numa correspondência que respeite a topologia, aquelas unidades que estejam fisicamente próximas umas das outras responderão analogamente a vetores de entrada similares. Mesmo sendo fácil visualizar esta similaridade ou proximidade em forma de uma matriz bidimensional, fica difícil determinar padrões similares em espaços multi-dimensionais. Nesta seção, serão apresentadas as técnicas para poder representar de forma natural esses mapas de características.

Este tipo de redes também conhecidas como Mapas Auto-organizáveis (SOM³) foram estudadas por Kohonen em [Kohonen 84]. Eles não precisam que lhes seja apresentada uma saída desejada para realizar a correção dos erros, ou seja, todas elas utilizam o Paradigma Não Supervisionado. Assim mesmo, o algoritmo utilizado para o processo de aprendizado da rede é o **Competitivo** que será explicado a seguir com mais detalhes.

5.2.2 Processo de Aprendizado em um SOM

Um SOM possui uma arquitetura muito simples. Apenas duas camadas compõem a rede. uma de entrada (que na verdade não é uma camada mas sim um filtro que é utilizado para apresentar os padrões) e uma de saída de forma similar à rede apresentada na Figura 6.

³As siglas correspondem ao nome em inglês *Self-Organizing Maps*.

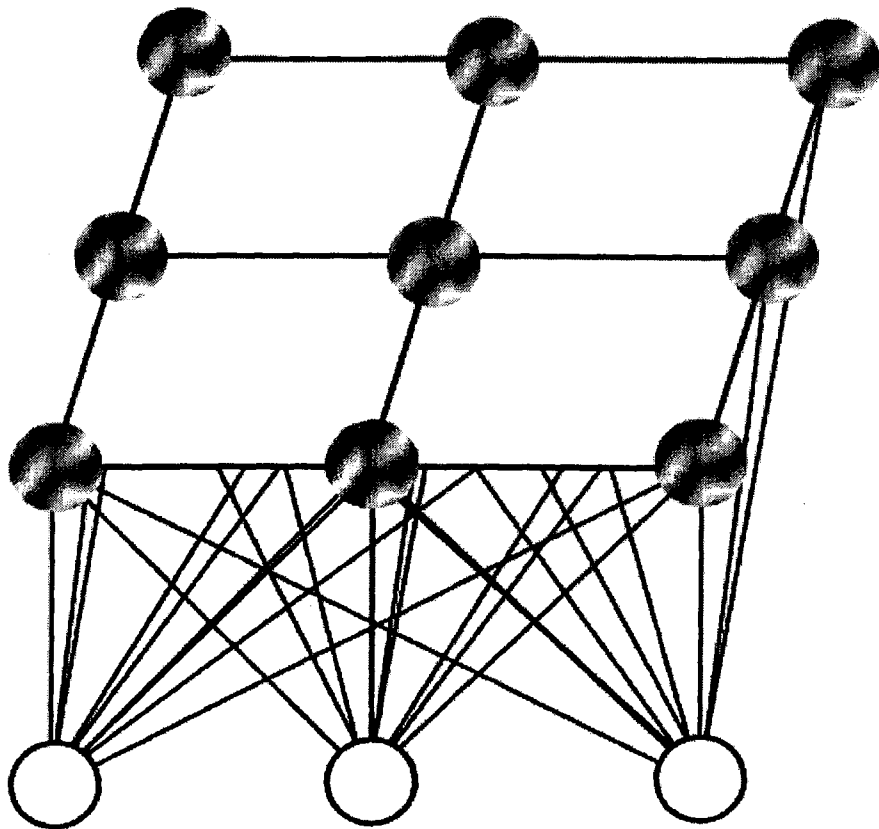


Figura 6: Arquitetura Básica de um Mapa Auto-Organizável.

Cada neurônio da camada de saída possui conexões com cada uma das entradas e existe um peso associado a cada uma dessas conexões. Quando um padrão é apresentado à rede, cada neurônio calcula a diferença do seu vetor de pesos em relação ao padrão que está sendo apresentado de acordo com a equação:

$$\|x - w\| \quad (8)$$

onde x é o vetor de entrada e w é o vetor de pesos. Depois de que cada unidade tem calculado essa diferença, o neurônio ganhador é aquele que cumpra com a equação:

$$\|x - w_c\| = \min\{\|x - w_i\|\} \quad (9)$$

onde c é o índice do neurônio ganhador e w_c é o seu vetor de pesos. Neste contexto, uma vez escolhida a unidade ganhadora, só ela tem o direito de atualizar suas conexões e as dos seus vizinhos. Um critério de vizinhança pode ser, por exemplo, escolher regiões retangulares em volta da unidade ganhadora como observado na Figura 7.

Depois de realizar a atualização dos pesos em volta do neurônio ganhador, a rede passa a ter vários grupos ou *clusters*. Cada *cluster* corresponde a um grupo de padrões similares.

sendo que o centro corresponde ao padrão que melhor representa aquela classe. Na Figura 8 pode-se observar um *cluster* junto com três das imagens que foram classificadas.

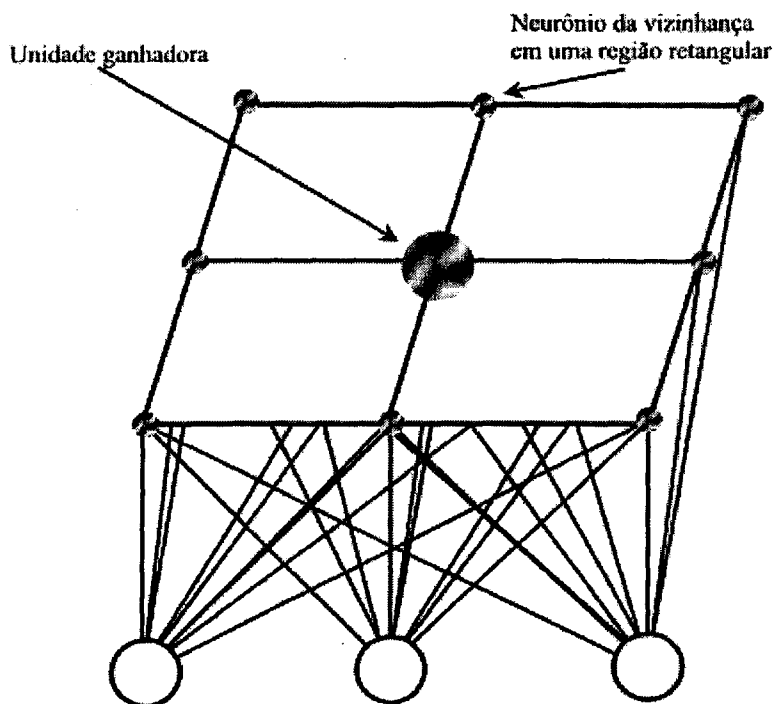


Figura 7: Vizinhança do neurônio ganhador em um mapa Auto-Organizável.

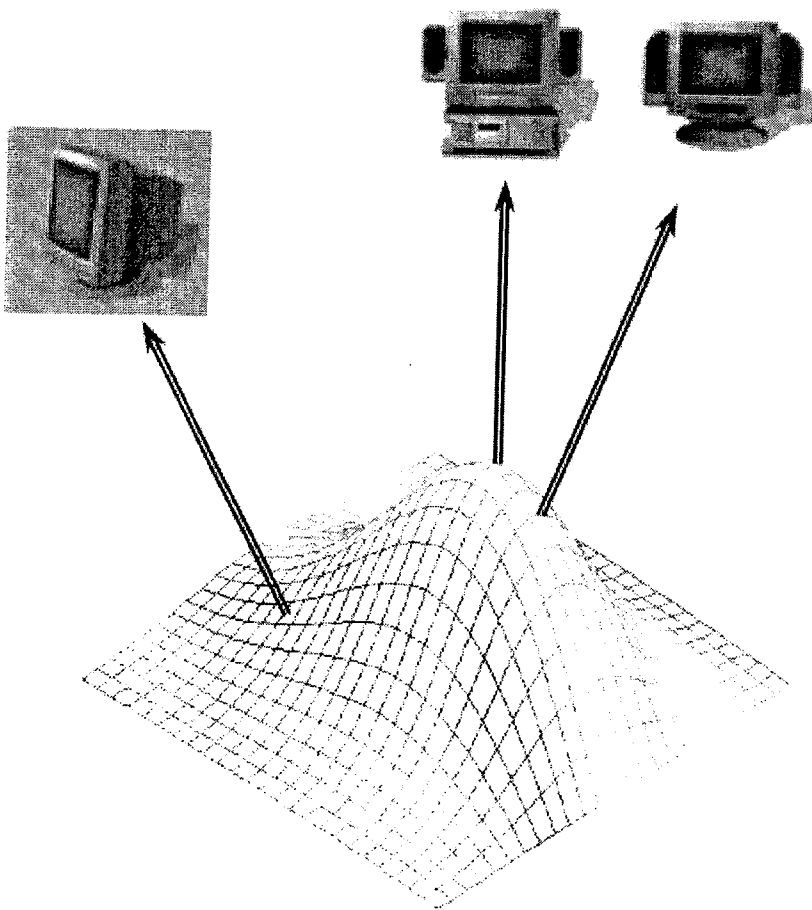


Figura 8: Correspondência entre a distância física dos neurônios de uma rede de Kohonen e o grau de semelhança dos padrões por eles reconhecidos.

A rede de Kohonen é composta por muitos *clusters*, reduzindo assim, os vetores multi-dimensionais a pontos numa superfície tri-dimensional como pode ser observado na Figura 9.

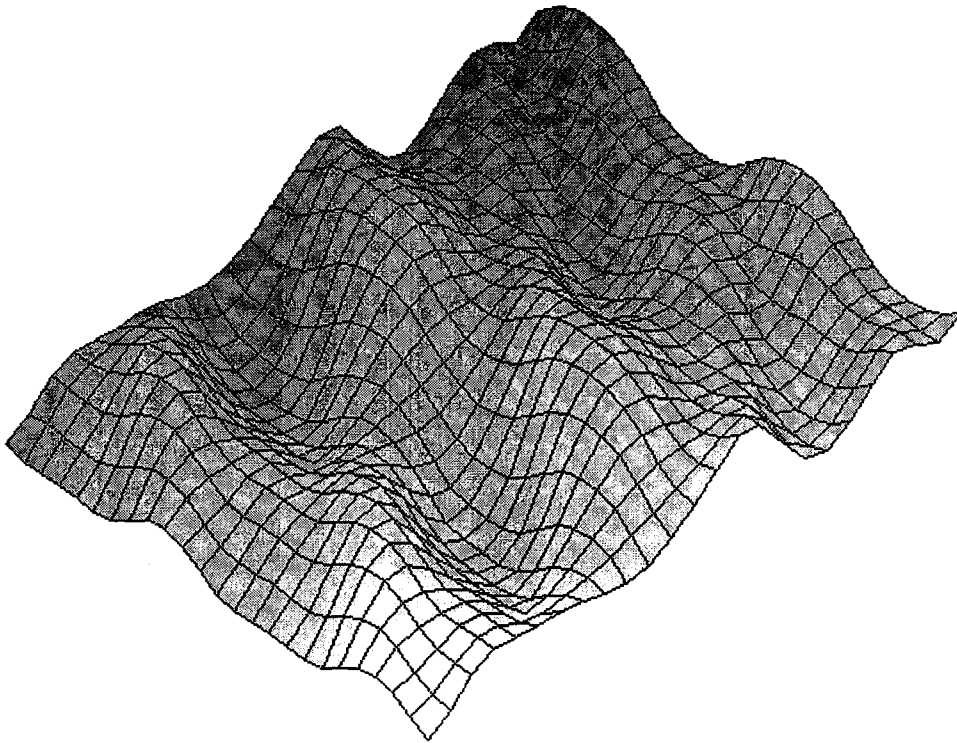


Figura 9: Clusters gerados por uma rede de Kohonen.

A distância do neurônio ganhador até o centro do *cluster* é diretamente proporcional ao grau de diferença do padrão sendo apresentado com o padrão representado no centro do *cluster*.

Como já foi mencionado anteriormente, quando da indexação e recuperação de dados multimídia em Bancos de Dados, supõe-se que ocorrerão inserções e remoções a qualquer momento. Essas condições fazem supor que a RNA utilizada deve ter a característica de crescer durante o treinamento ou inclusive remover aquelas unidades que não sejam mais úteis depois de uma remoção de padrões. Um tipo de Rede Neural Construtiva que utiliza Aprendizado Não Supervisionado, a *Growing Neural Gas* será apresentada na próxima seção.

5.3 Growing Neural Gas

5.3.1 Introdução

O método *Growing Neural Gas* (GNG) foi proposto por Fritzke em [Fritzke 1995a]. Este modelo surgiu principalmente com o objetivo de melhorar algumas limitações do modelo básico de Kohonen. Enquanto o modelo de Kohonen parte de uma topologia fixa, uma

rede GNG inicia o seu treinamento apenas com 2 neurônios e novas unidades são criadas ao longo do processo de treinamento com o intuito de melhorar o desempenho da rede. Portanto, o modelo GNG, além de trabalhar com o Paradigma Não Supervisionado, também é construtivo, sendo capaz de gerar uma topologia diferente para cada tipo de problema.

Uma outra diferença em relação ao modelo de Kohonen é a forma de conectar as unidades. O mapa de Kohonen cria conexões laterais formando uma cruz em cada unidade como já foi visto na Figura 6. Já no modelo GNG, uma unidade pode ter muito mais de quatro vizinhos gerando diversas figuras geométricas⁴ e uma rede com maior capacidade de aprendizado.

O mecanismo para o crescimento de estruturas celulares [Fritzke 1994a], [Fritzke 1994b] junto com a geração de topologias utilizando Aprendizado Hebbiano ou Competitivo [Martinetz 91] foram combinados para criar este novo modelo. Uma rede GNG inicia o processo de treinamento com dois neurônios só e novas unidades vão sendo inseridas sucessivamente. Para determinar em que posição deverá ser inserida uma nova unidade, informações relacionadas ao erro são recoletadas durante o processo de adaptação. Cada nova unidade deverá ser inserida perto da unidade com maior erro. O algoritmo *Growing Neural Gas* pode ser analisado a seguir⁵.

1. Inicializar a rede A com duas unidades c_1 e c_2

$$A = \{c_1, c_2\} \quad (10)$$

Os pesos devem ser inicializados com valores aleatórios. Inicializar o conjunto de conexões C. $C \subset A \times A$ por:

$$C = \emptyset \quad (11)$$

2. Apresentar um padrão ε ⁶ à rede de acordo com uma distribuição uniforme $p(\varepsilon)$
3. Determinar o neurônio ganhador s_1 e o mais próximo dele s_2 de acordo com as equações (12) e (13) onde:

$$s_1 = \min \| \varepsilon - w_c \| \quad \forall c \in A \quad (12)$$

$$s_2 = \min \| \varepsilon - w_c \| \quad \forall c \in A - \{s_1\} \quad (13)$$

onde $\| \varepsilon - w_c \|$ representa a distância euclideana entre os vetores ε e w_c .

⁴Em duas dimensões, se a rede estiver muito carregada, as unidades tendem a formar triângulos como visto na Figura 20.

⁵A nomenclatura utilizada corresponde ao artigo original escrito por Fritzke em [Fritzke 97]

⁶ $\varepsilon \in R^n$, onde n é a dimensão dos padrões que estão sendo indexados.

4. Se não existe uma conexão entre s_1 e s_2 , então criá-la.

$$C = C \cup \{s_1, s_2\} \quad (14)$$

Inicializar a idade desta nova conexão com 0

$$idade_{(s_1, s_2)} = 0 \quad (15)$$

5. Adicionar o quadrado da distância entre o neurônio ganhador e o padrão apresentado a uma variable de erro local:

$$\Delta E_{s_1} = \|\varepsilon - w_{s_1}\|^2 \quad (16)$$

6. Sendo ε_b e ε_n as taxas de aprendizado para o neurônio ganhador e para os seus vizinhos respectivamente. O vetor de pesos relacionados às conexões da unidade ganhadora e os vetores de pesos dos seus vizinhos são atualizados da seguinte forma.:

$$\Delta w_{s_1} = \varepsilon_b(\varepsilon - w_{s_1}) \quad (17)$$

$$\Delta w_i = \varepsilon_n(\varepsilon - w_i) \quad (\forall i \in N_{s_1}) \quad (18)$$

onde N_{s_1} é o conjunto de vizinhos topológicos diretos de s_1 .

7. Incrementar a idade de todas as conexões de s_1 :

$$idade(s_i, i) = idade(s_i, i) + 1 \quad \forall i \in N_{s_1}. \quad (19)$$

8. Remover as conexões com idade maior de a_{max} ⁷. Se depois da remoção existirem unidades sem conexões, elas devem ser removidas da rede.

9. Se o número de padrões apresentados até o momento for múltiplo de um certo número λ , uma nova unidade deve ser inserida assim:

- Determinar a unidade \mathbf{q} com o maior erro acumulado de toda a rede.

$$q = \max\{E_c \mid \forall c \in A\} \quad (20)$$

- Determinar entre os vizinhos of \mathbf{q} a unidade \mathbf{f} com maior erro acumulado.

$$f = \max\{E_c\} \quad \forall c \in N_q \quad (21)$$

- Adicionar uma nova unidade \mathbf{r} à rede e interpolar seu vetor de pesos a partir de \mathbf{q} e \mathbf{f} de acordo com a equação 22.

$$A = A \cup r, \quad w_r = \frac{w_q + w_f}{2} \quad (22)$$

⁷ a_{max} é a idade máxima permitida para uma conexão.

- Inserir conexões de r até q e de r até f e remover a conexão original entre q e f :

$$C = C \cup \{(r, q), (r, f)\}, \quad C = C - \{(q, f)\} \quad (23)$$

- Diminuir as variáveis de erro das unidades q e f em uma fração α :

$$\Delta E_q = -\alpha E_q, \quad \Delta E_f = -\alpha E_f \quad (24)$$

- Interpolar a variável de erro de r a partir de q e f :

$$E_r = \frac{E_q + E_f}{2} \quad (25)$$

10. Diminuir a variável de erro de todas as unidades:

$$\Delta E_c = -\beta E_c, \quad \forall c \in A \quad (26)$$

onde β é a taxa de correção de erros

11. Se o critério de parada ainda não foi alcançado (isto é, tamanho máximo da rede ou alguma outra medida de desempenho) voltar ao passo 2.

5.3.2 Exemplos de reconhecimento com uma rede GNG

Para realizar alguns exemplos com este modelo, foi implementada uma rede *Growing Neural Gas*⁸. Além disso, diversos objetos no espaço 3D foram criados para serem reconhecidos. A dimensão adotada é ainda uma dimensão baixa em relação às dimensões de dados multimídia. Inicialmente, foi escolhida esta dimensão para poder observar o comportamento deste tipo de redes e entender melhor o método que se deseja utilizar para a futura indexação de dados multimídia⁹.

Neste caso, diversos objetos volumétricos serão objeto de estudo. Para realizar o treinamento da rede, são gerados pontos aleatórios da forma (x,y,z) e a rede deve reconhecer quantos objetos existem (isto é, reconhecer os *clusters*) assim como as formas deles. Na Figura 10 e na Figura 11 são apresentados os diversos objetos básicos que foram utilizados para realizar o reconhecimento de volumes e, na Figura 16 podem ser observados alguns objetos compostos. Com esta filosofia básica é possível também desejar que a rede reconheça objetos mais complexos como mesas, cadeiras, casas, etc. Também acredita-se na possibilidade generalizar o conceito para reconhecer e classificar corretamente dados multimídia.

⁸A linguagem utilizada para a implementação foi C++. O compilador foi o Visual C++ 6.0.

⁹Para a visualização foi utilizada a hierarquia VTK (*Visualization Toolkit*) versão 3.1.

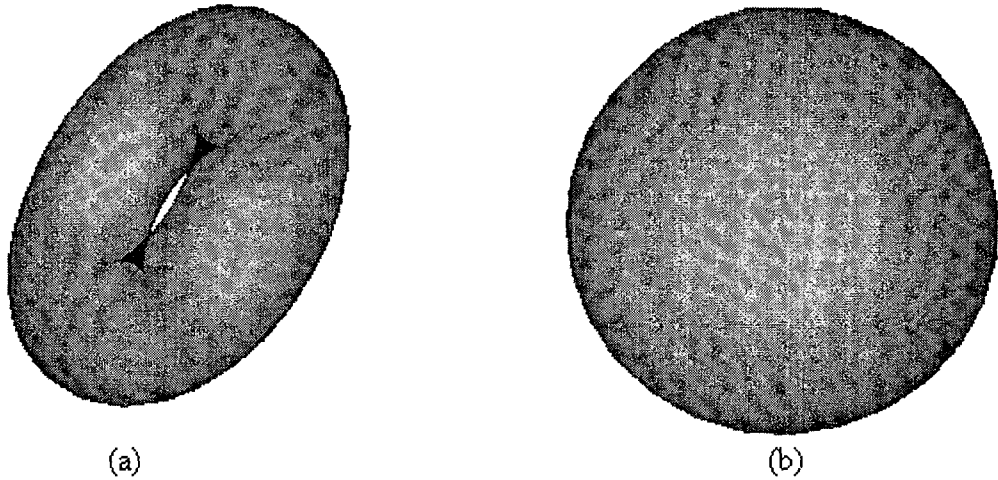


Figura 10: Objetos volumétricos utilizados para os experimentos (a) Toroide (b) Esfera.

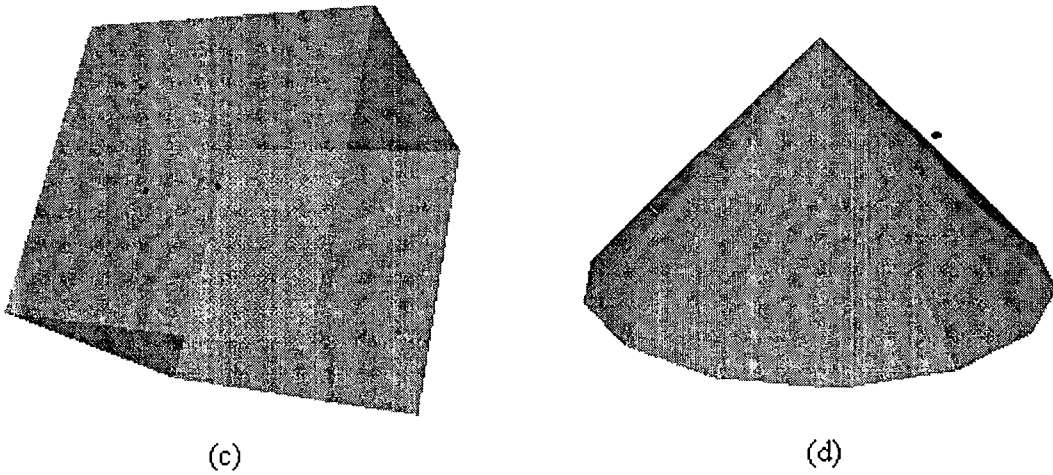


Figura 11: Objetos volumétricos utilizados para os experimentos (c) Cubo (d) Cone.

Foram realizados diversos experimentos para observar o comportamento da rede GNG com os objetos apresentados. Na Figura 12 pode se observar a evolução do treino quando o objeto é um toroide. Os parâmetros utilizados para esta simulação foram: $\lambda = 400$, $\varepsilon_b = 0.05$, $\varepsilon_n = 0.0006$, $\alpha = 0.5$, $\beta = 0.0005$, $a_{max} = 100$ e um número máximo de neurônios igual a 100.

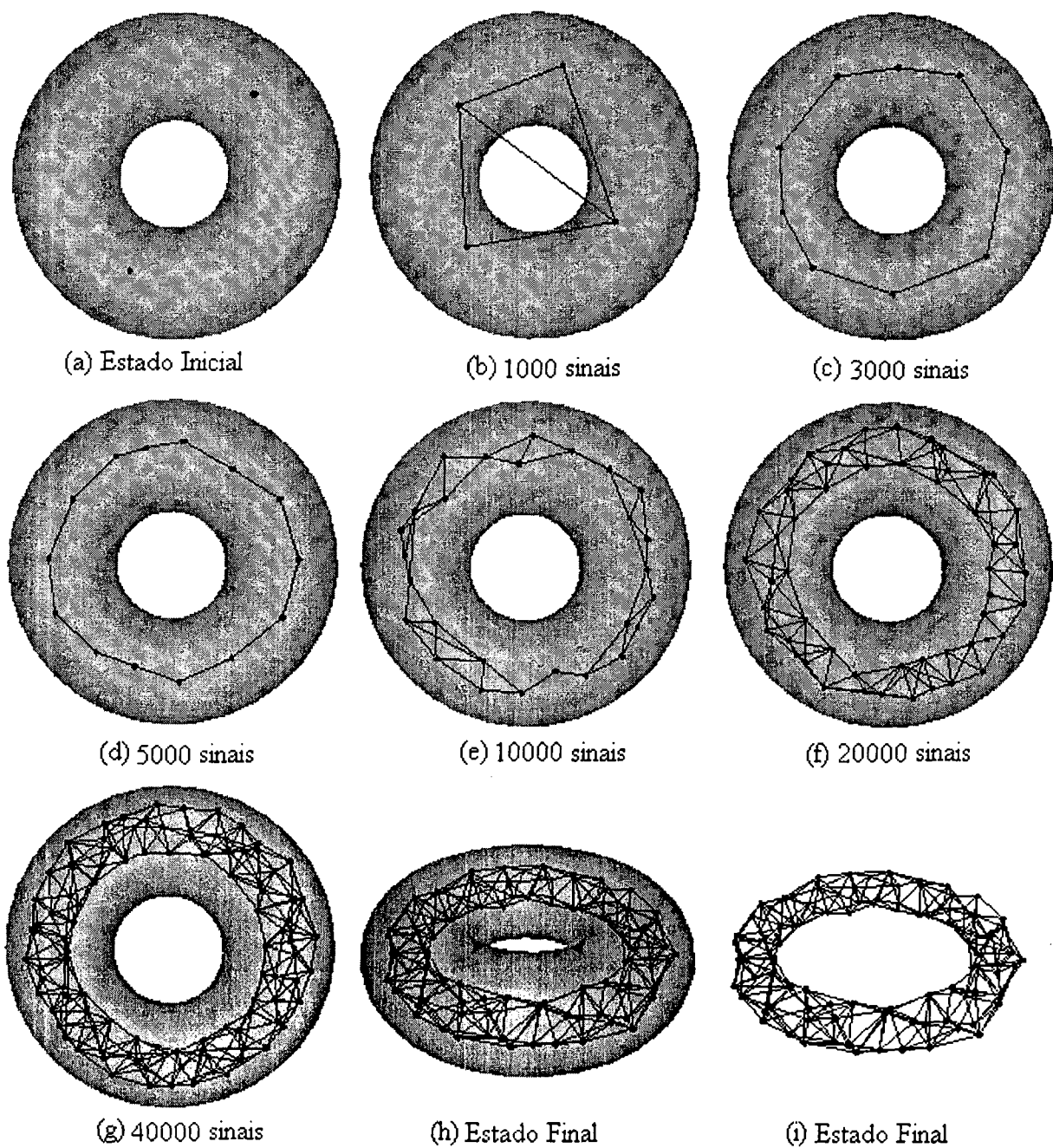


Figura 12: Evolução do treino de uma rede GNG durante o reconhecimento de um toroide.

A Figura 12(h) corresponde ao estado final da rede depois de 40000 padrões vista lateralmente e a Figura 12(i) também é a rede final sem o toroide que está sendo reconhecido.

Na Figura 13 é apresentado o treino da rede GNG para o reconhecimento de um volume com forma de cubo simétrico. Os parâmetros utilizados para esta simulação foram: $\lambda = 400$, $\varepsilon_b = 0.05$, $\varepsilon_n = 0.0006$, $\alpha = 0.5$, $\beta = 0.0005$, $a_{max} = 100$ o número de sinais apresentados foi de 10000 e um número máximo de neurônios foi igual a 27 e 64 em 13(a) e 13(b), respectivamente. Escolheu-se 27 e 64 por ser 3^3 e 4^3 respectivamente espe-

rando que a rede gerada adotasse também uma forma simétrica. O resultado foi muito positivo, em ambos os casos aconteceu o mesmo. A rede gerada tendeu a formar malha tridimensionais de 3 e 4 neurônios por lado respectivamente.

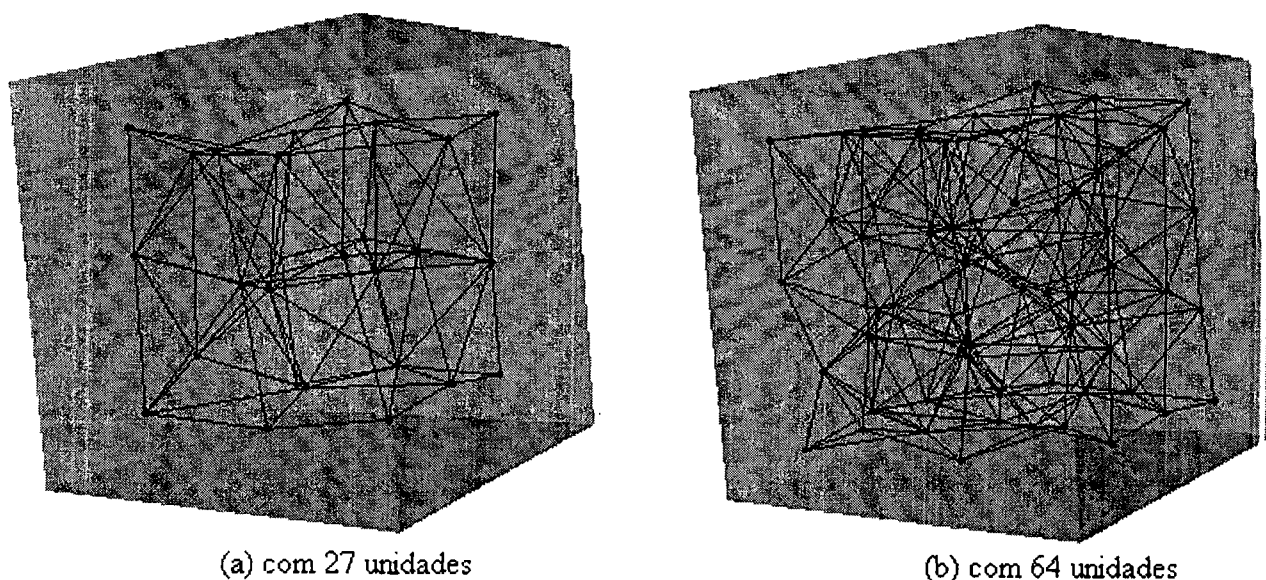


Figura 13: Evolução do treino de uma rede GNG durante o reconhecimento de um cubo.

A partir dos objetos básicos podem ser criados outros objetos compostos com maior grau de complexidade. Na Figura 14 pode-se observar como a rede percebe que os sinais gerados não pertencem a um único objeto e inicia-se o processo de espalhamento. Na Figura 15 pode-se observar que a rede já foi dividida formando 3 sub-redes sem relações diretas entre elas. Vale a pena ressaltar que inclusive as sub-redes geradas dentro de cada um dos 3 cubos, também formam têm a tendência de se organizar em forma de um cubo. Os parâmetros utilizados para esta simulação foram: $\lambda = 600$, $\varepsilon_b = 0.5$, $\varepsilon_n = 0.0006$. $\alpha = 0.5$, $\beta = 0.0005$, $a_{max} = 100$

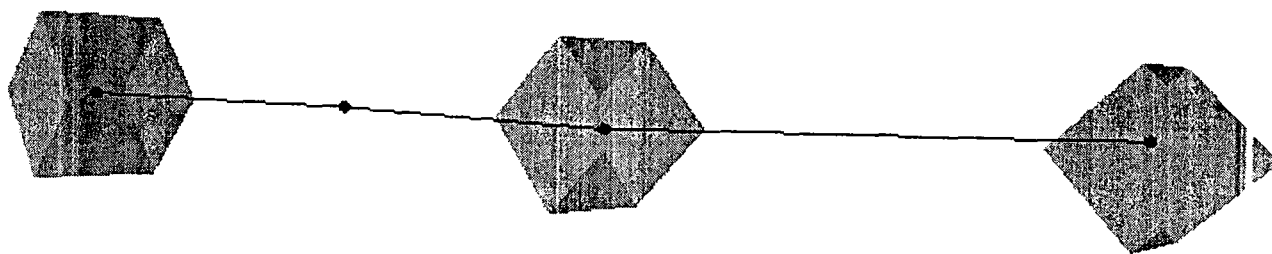


Figura 14: Treinamento de uma rede GNG para o reconhecimento de volumes com forma de cubo depois de 3000 sinais apresentados.

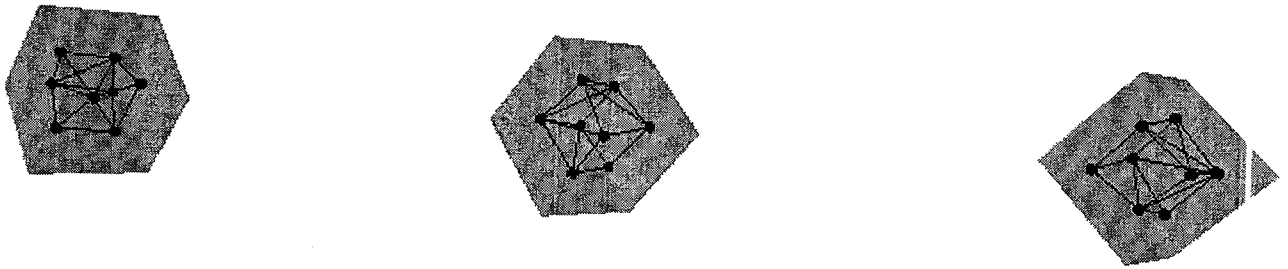


Figura 15: Treinamento de uma rede GNG para o reconhecimento de volumes com forma de cubo depois de 3000 sinais apresentados.

Na Figura 16 a rede só recebeu 5000 sinais e pode-se observar que a distribuição dos objetos já foi percebida por ela. Depois de 20000 sinais, a rede consegue separar muito bem os objetos que estão fisicamente longe. Observe-se também que, quando existe uma intersecção entre objetos ou eles estão muito perto a rede não se sub-divide.

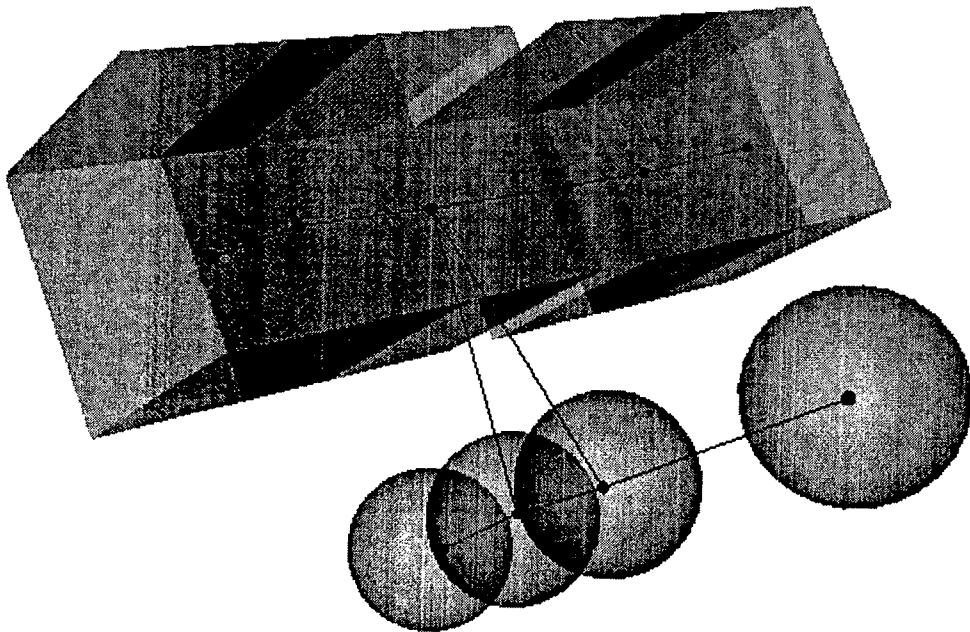


Figura 16: Treinamento de uma rede GNG para o reconhecimento de volumes com intersecções depois de 3000 sinais apresentados.

Esta mesma rede depois de 20000 sinais se divide em 3 sub-redes como pode ser observado na Figura 16

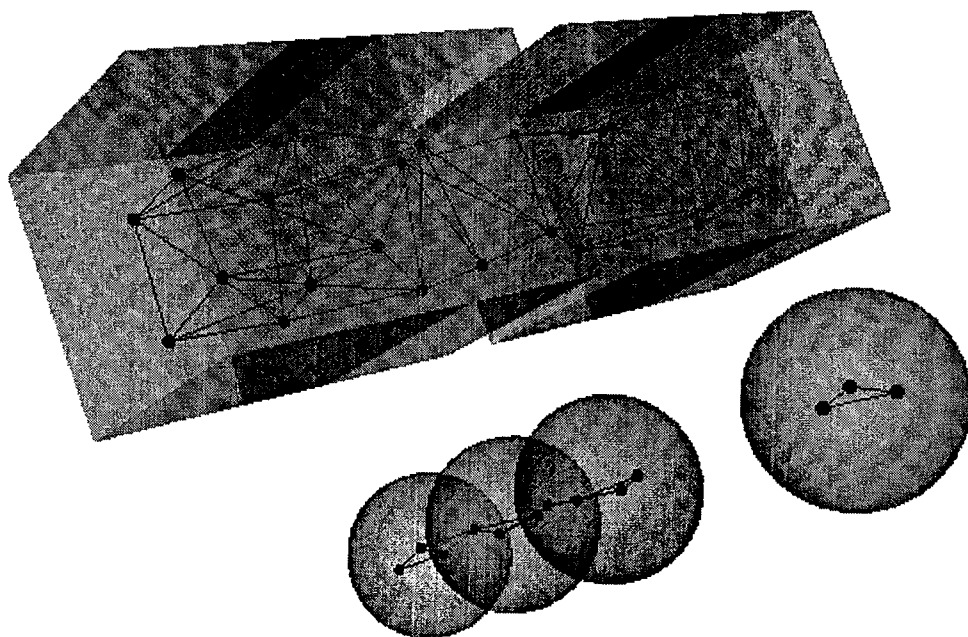


Figura 17: Treinamento de uma rede GNG para o reconhecimento de volumes com intersecções depois de 20000 sinais apresentados.

Outros experimentos também foram realizados com objetos gerados em posições e com dimensões aleatórias. O resultado continuou sendo positivo, em todos os casos a rede se dividiu e quando da intersecção de objetos a rede se manteve unida como observado nas Figura 18, e 19. Os parâmetros utilizados para esta simulação foram: $\lambda = 600$, $\varepsilon_b = 0.5$, $\varepsilon_n = 0.0006$, $\alpha = 0.5$, $\beta = 0.0005$, $a_{max} = 100$ e foram gerados 40000 sinais.

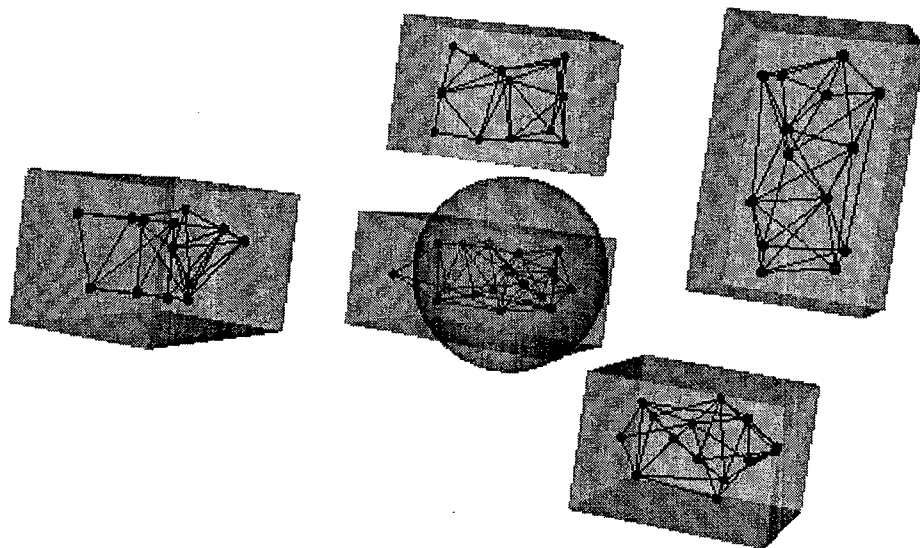


Figura 18: Treinamento de uma rede GNG para o reconhecimento de volumes gerados de forma aleatória.

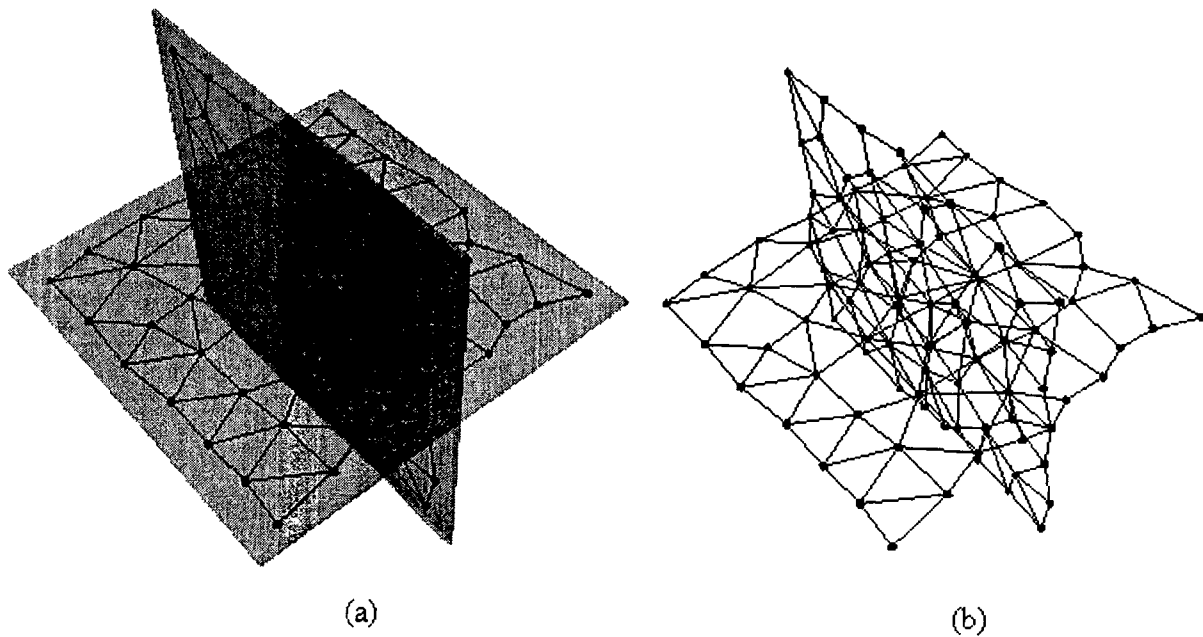


Figura 19: Treinamento de uma rede GNG para o reconhecimento de dois planos com intersecção (a) Com os Planos (b) Sem os planos.

6 Proposta

Com base no estudo realizado, uma proposta inicial para resolver o problema colocado é: utilizar uma primeira RNA para a detecção dos *clusters* e, depois de conhecer o *cluster* com que o padrão tem maior grau de similaridade, poder-se-á utilizar uma segunda RNA para indexar o padrão. Desta forma é possível reduzir a dimensionalidade do padrão para uma superfície tridimensional. O esquema geral do modelo proposto pode ser observado a seguir na Figura 20.

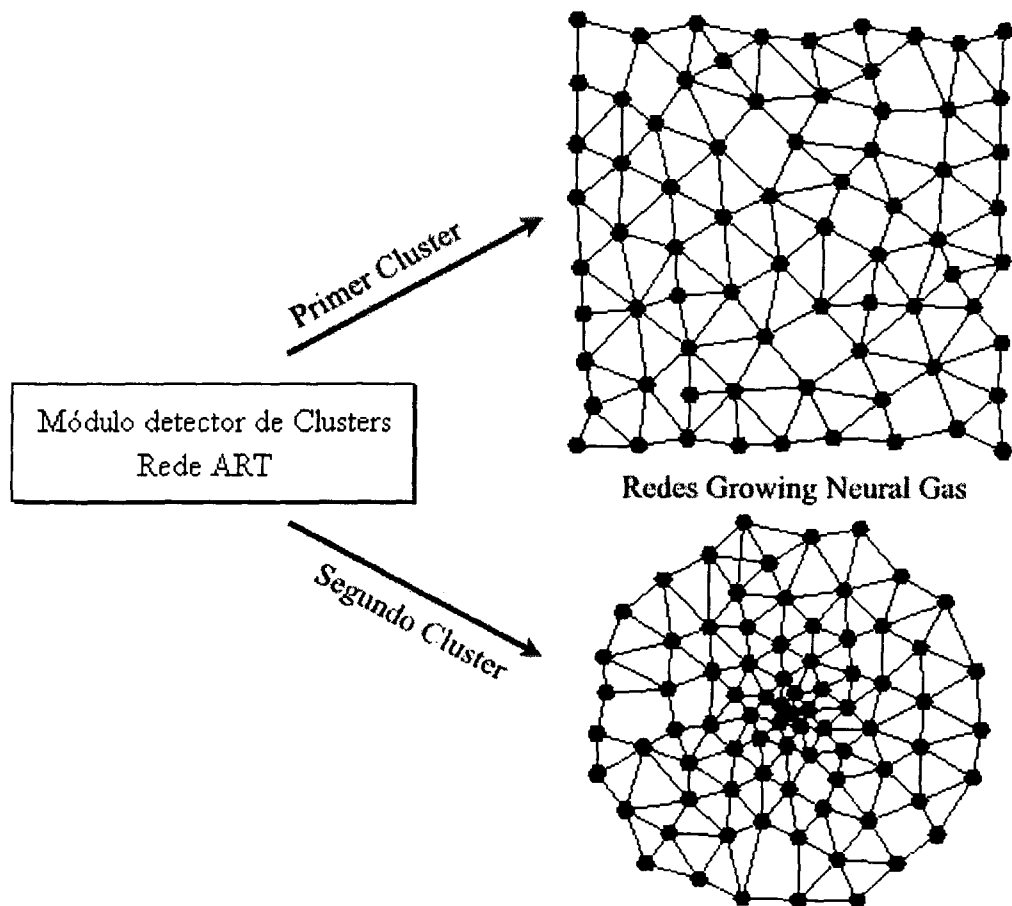


Figura 20: Modelo Proposto

Baseados neste modelo básico, e considerando os requisitos que cada uma das redes deve cumprir (ser **Construtiva** e trabalhar com o **Paradigma Não Supervisionado**), propõe-se três alternativas que serão apresentadas a seguir.

6.1 Proposta 1

Considerando o caso onde não se conhece o número de *clusters* existentes nos dados. A primeira das redes poderia ser da família ART já que só se deve detectar aglomerados.

A partir de cada *cluster* detectado deve ser criada uma segunda rede para classificar os padrões dentro desse grupo, já que, o grande volume de informação, realmente, encontra-se no segundo nível. Uma rede mais adequada para esta tarefa é a rede *Growing Neural Gas*.

6.2 Proposta 2

Considerando o caso onde se conhece a priori o número de aglomerados (n) que irão existir nos dados, então no primeiro nível do modelo poderia-se utilizar uma rede GNG com n . A partir de cada neurônio poderia existir um segundo nível de rede GNG “pindurada”. criando, desta forma uma estrutura hierárquica de redes GNG poderia ser criada. Esta estrutura poderia ser chamada de *Multi-Level Growing Neural Gas*.

O principal motivo para a criação da rede do primeiro nível é a redução do processamento envolvido. O neurônio que representa o *cluster* ganhador terá o direito de propagar o sinal a sua sub-rede. Isto diminuirá o tempo envolvido. A diminuição de tempo é muito importante porque a dimensionalidade de dados multimídia é realmente alta.

6.3 Proposta 3

Uma alternativa para a redução de processamento é a utilização de algum algoritmo para a redução da dimensão do padrão de entrada. Uma técnica que se tem mostrado eficiente é a Análise da Componente Principal ou PCA¹⁰. Com esta técnica é possível reduzir de forma significativa a dimensão dos dados e, assim, o processamento também seria reduzido. Neste caso, inclusive, a rede do primeiro nível teria o seu trabalho reduzido consideravelmente.

7 Conclusões e Trabalhos Futuros

Devido às características das Redes Neurais antes analisadas, elas poderiam ser uma alternativa viável para a solução do problema de similaridade em um Sistema Gerenciador de Bases de Dados Multimídia (SGBDM).

Para um melhor desempenho, o módulo deve ser dividido em dois níveis de redes. No primeiro nível apenas se relaciona o padrão de entrada com um *cluster*. Depois de escolher um único agrupamento, o padrão é classificado por uma segunda RNA pertencente ao segundo nível, que é onde realmente se armazena a informação.

Uma técnica para diminuir a alta dimensionalidade de dados multimídia é utilizar a técnica PCA. A incorporação deste módulo, diminuiria o processamento requerido para a indexação.

¹⁰As sigla corresponde ao nome em inglês *Principal Component Analysis*.

As três propostas apresentadas por este trabalho devem ser testadas em breve com dados com mais dimensões. Só assim será possível comparar o desempenho e obter resultados mais concretos em relação a dados multimídia.

Referências

- [Baeza-Yates 94] R.A. Baeza-Yates, W. Cunto, U. Manber, S. Wu. *Proximity Matching Using Fixed-Queries Tree*. pp.198-212. CPM 1994.
- [Bartfai 94] G. Bartfai. *Hierarchical Clustering with ART Neural Network*. Technical Report CS-TR-94/1. Department of Computer Science, Victoria. University of Wellington, New Zealand. Janeiro 1994.
- [Bartfai 95] G. Bartfai. *An ART-based Modular Architecture for Learning Hierarchical Clustering*. Technical Report CS-TR-95/3. Department of Computer Science, Victoria. University of Wellington, New Zealand. February, 1995.
- [Beckmann 90] N. Beckmann, H.P.Kriegel, R. Schneider, B. Seeger. *The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles*. ACM SIGMOD. pp. 322-331. 1990.
- [Beker 97] M. F. Beker. *Representação de Áudio em Bancos de Dados*. Dissertação de Mestrado, ICMC-USP. Setembro 1997.
- [Bozcaya 97] T. Bozcaya, M. Ozsoyoglu. *Distance-Based Indexing for High-Dimensional Metric Spaces*. ACM SIGMOD. pp. 357-368. 1997.
- [Brin 95] S. Brin. *Near Neighbor search in large metric spaces*. VLDB. pp. 574-584. 1995.
- [Carpenter 87a] G. Carpenter, S. Grossberg. *A massively parallel architecture for a Self-Organization Neural Pattern Recognition Machine*. Computer Vision, Graphics and Image Proc. 37. pp. 54-115. 1987.
- [Carpenter 87b] G. Carpenter, S. Grossberg. *ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns*. Applied Optics. Vol. 26. Nro. 23. pp. 4919-4930. December 1987.
- [Carpenter 88] G. Carpenter, S. Grossberg. *The ART of adaptive pattern recognition by a self-organizing neural network*. Computer, 21(3). pp. 77-88. Março 1988.
- [Carpenter 90] G. Carpenter, S. Grossberg. *ART3: Hierarchical Search Using Chemical Transmitters in Self-Organizing Pattern Recognition Architectures*. Neural Networks. Vol. 3. pp. 129-152. 1990.
- [Carpenter 91a] G. Carpenter, S. Grossberg, J. H. Reynolds. *Supervised Real-Time Learning and Classification of Nonstationary Data by Self-Organizing Neural Networks*. Neural Networks. Vol. 4. pp. 565-588. 1991.
- [Carpenter 91b] G. Carpenter, S. Grossberg, D. B. Rosen. *Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System*. Neural Networks. Vol. 4. pp. 759-771. 1991.

- [Carpenter 91c] G. Carpenter, S. Grossberg, D. B. Rosen. *ART2a: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition*. Neural Networks. Vol. 4. pp. 493-504. 1991.
- [Carpenter 92] G. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, D. B. Rosen. *Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps*. IEEE Transactions on Neural Networks. Vol. 3. Nro. 5. pp. 698-713. Setembro 1992.
- [Chiueh 94] T. Chiueh. *Content-Based Image Indexing*. International Conference on Very Large Databases. pp.582-593. 1994.
- [Ciaccia 97] P. Ciaccia, M. Patella, P. Zezula. *M-Tree: An efficient access method for similarity search in metric spaces*. VLDB. pp. 426-435. 1997.
- [Faloutsos 95] C. Faloutsos, K. Lin. *FastMap: A Fast Algorithm for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets*. SIGMOD. pp. 163-174. 1995.
- [Fritzke 1994a] B. Fritzke. *Growing Cell Structures - a self-organizing networks for unsupervised and supervised learning*. Neural Networks, 7(9). pp. 1441-1460. 1994.
- [Fritzke 1994b] B. Fritzke. *Fast Learning with incremental RBF networks*. Neural Processing Letters, 1(1). pp. 2-5. 1994.
- [Fritzke 1995a] B. Fritzke. *A growing neural gas networks learns topologies*. Advances in Neural Information Processing Systems 7. pp. 625-632. MIT Press, Cambridge MA. 1995.
- [Fritzke 97] B. Fritzke. *Some Competitive Learning Methods*. Institute of Neural Computation. Ruhr-Universität Bochum. Abril 1997.
- [Gaede 98] V. Gaede, O. Gunther. *Multidimensional Access Method*. ACM Computing Surveys. Vol. 30. N 2. pp. 170-231, 1998.
- [Grossberg 82] S. Grossberg. *Adaptive pattern classification and universal recoding, I: Parallel development and coding of neural feature detectors*. Editado por Stephen Grossberg. Studies of Man and Brain. D. Reidel Publishing. pp. 448-497. Boston. 1982.
- [Guttman 84] A. Guttman. *R-Tree: A dynamic Index Structure for Spatial Searching*. ACM SIGMOD. pp. 47-57. 1984.
- [Haykin 99] S. Haykin. *Neural Network, A Comprehensive Foundation*. Prentice Hall. 1999.
- [Hellerstein 95] J.M. Hellerstein, J.F. Naughton, A. Pfeffer. *Generalized Search Trees for Database Systems*. VLDB. pp. 562-573. 1995.

- [Kohonen 84] T. Kohonen. *Self Organization and Associative Memory*. Vol. 8 de Springer Series of Information Science. Springer-Verlag. 1984.
- [Martinetz 91] T. M. Martinetz, K. J. Schulten. *A "neural gas" network learns topologies*. In T. Kohonen, K. Mäkiä, O. Simula, and J. Kangas, editors. *Artificial Neural Networks*. pp. 397-402. North-Holland, Amsterdã. 1991.
- [Mitchell 97] T. Mitchell. *Machine Learning*. WCB/McGraw-Hill. 1997.
- [Murre 89] J. M. J. Murre, R. H. Phaf, G. Wolters. *CALM Networks: A Modular Approach to Supervised and Unsupervised Learning*. Proceedings of the International Joint Conference on Neural Networks. Washington DC, 1. New York: IEEE Press pp. 649-665. 1989.
- [Parekh 95] R. Parekh., J. Yang, V. Honavar. *Constructive Neural Network Learning Algorithms for Multi-Category Pattern Classification*. Technical Report TR95-15a. 1995.
- [Papadias 95] D. Papadias, N. Mamoulis, Y. Manalopoulos. *Processing and Optimization of Multiway Spatial Joins Using R-Trees*. PODS. pp.44-55. 1999.
- [Rumelhart 86] D.E. Rumelhart, J. L. Mc Clelland. *Parallel Distributed Processing*. MIT Bradford Press. 1986.
- [Santos 97] R. R. Santos. *Incorporação do Tipo de Dado Imagem Imagem em um Banco de Dados*. Tese de Doutorado apresentada ao IFSC-USP. Novembro 1997.
- [Sellis 87] T. Sellis, N. Roussopoulos, C. Faloutsos. *The R+tree: A Dynamic Index for Multidimensional Objects*. VLDB. pp. 507-518. 1987.