Mapas Auto-organizáveis - Extensões
(*Self-Organizing Maps* - SOM)

Aluizio Fausto Ribeiro Araújo
Universidade Federal de Pernambuco
Centro de Informática - CIn
Departamento de Sistemas da Computação
aluizioa@cin.ufpe.br

Conteúdo

- Introdução
- *Grid* Diferente do SOM
- Ponderação no cálculo de distâncias
- Modelo estocástico
- *Kernel* SOM
- Modelo hierárquico
- Estrutura variante no tempo

Introdução

- Objetivos:
  - Apresentar diferenças e contribuições de alguns mapas derivados do SOM (*Self-Organizing Map*);
  - Atuar nas limitações encontradas na literatura;

Introdução

- Algumas limitações do SOM:
  - A pré-definição da estrutura limita o mapeamento resultante,
    - Número fixo de nodos;
    - Conexões entre nodos pré-definidas;
  - Tipos limitados de topologia dos mapas,
    - Poucos tipos de vizinhanças;
  - Emprego de distância euclideana,
    - Mesma importância das dimensões;
    - Distâncias que podem perder capacidade de discriminação;
  - Processamento restrito a uma camada,
    - Dificuldades em perceber diferenças sutis intracategóricas;

Grid Diferente

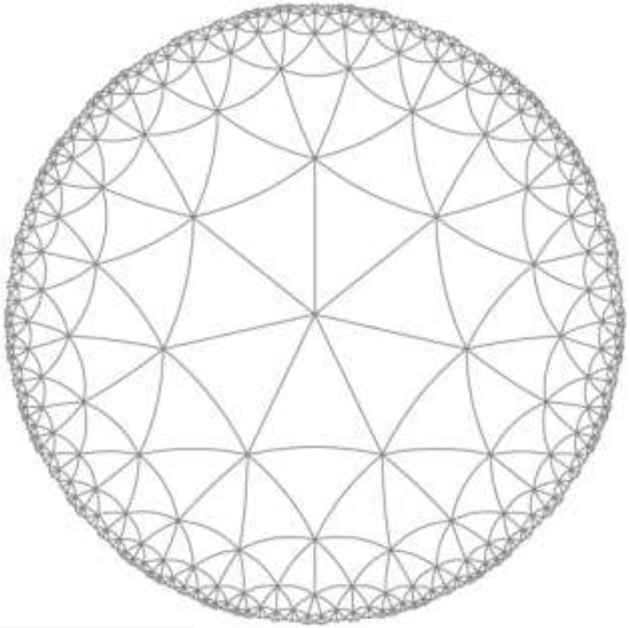
Hyperbolic Self-Organizing Maps – HSOM (2001)

- Contribuição:
  - O SOM possui uma vizinhança bastante restrita ajustada ao redor de um ponto sobre uma superfície euclidiana 2D;
  - Espaços hiperbólicos oferecem uma geometria cujo tamanho da vizinhança ao redor de um ponto cresce exponencialmente,
    - Permite novas configurações de vizinhança;

Grid Diferente

Hyperbolic Self-Organizing Maps (2001)

- Contribuição:
  - Plano hiperbólico:
  - Com triângulos:
    - Equiláteros;
    - Congruentes.



Grid Diferente

Hyperbolic Self-Organizing Maps (2001)

- Contribuição:
  - Competição:

$h_{r\,s}(n) = \exp\,(-d^2(r,s)/2\sigma^2)$

$r \in N(s,t)$, vizinhança do vencedor

$$d = 2\operatorname{arctanh}\left(\left|\frac{z_r - z_s}{1 - \bar{z}_s \cdot z_r}\right|\right)$$

onde para um *lattice* hiperbólico, cada nodo guarda um número complexo que identifica sua posição no modelo de Poincaré.

U F P E

Centro de Informática UFPE

Grid Diferente

Hyperbolic Self-Organizing Maps (2001)

- Aplicação: categorização de textos;
  - Criação de relação semântica entre textos armazenados em uma base muito grande;
  - Elaboração de visualização de baixa dimensão;
  - Agrupamento de textos por categorias.

U F P E

Centro de Informática
UFPE

Ponderação no Cálculo de Distâncias

Parameterized Self-organizing Maps – PSOM (1998)

- Contribuição:
  - Muitos modelos de redes neurais necessitam de centenas ou milhares de instâncias e iterações para treinamento;
  - Há casos nos quais a criação de uma base de dados requer muito esforço;
  - PSOM possui amplia capacidade de generalização em base de dados pequenas, além de ser rápido para treinar.

Ponderação no Cálculo de Distâncias

Parameterized Self-organizing Maps – PSOM (1998)

- Competição:
  - Pode selecionar subespaços dos dados de entrada;
  - **p** seleciona diferente subespaços;
  - Pondera os componentes das distâncias;

$$dist(x,x') = \sum_{k=1}^{d} p_k (x_k - x'_k)^2$$

Ponderação no Cálculo de Distâncias

Parameterized Self-organizing Maps – PSOM (1998)

- Aplicação:
  - Controle de robôs articulados, por exemplo, mão robótica.

Modelo Estocástico

Bayesian Self-Organising Map – BSOM (2001)

- Teorema de Bayes:
  - Modelagem por inferência estatística;
  - Pr(A) e Pr(B) são as probabilidades a priori de A e B;
  - Pr($B|A$) e Pr($A|B$) são as probabilidades a posteriori de B condicional a A e de A condicional a B respectivamente;

$$\Pr(A|B) = \frac{\Pr(B|A)\Pr(A)}{\Pr(B)}$$

Modelo Estocástico

Bayesian Self-Organising Map – BSOM (2001)

- Contribuição:
  - Em reconhecimento de padrões é importante que haja uma estimativa precisa da distribuição dos dados;
  - Mistura de distribuições oferece uma abordagem mais flexível para estimar a densidade dos dados;
  - BSOM possibilita a mistura de gaussianas;

Modelo Estocástico

Bayesian Self-Organising Map – BSOM (2001)

- Competição:
  - A métrica para determinar o vencedor é substituída por uma regra baseada na probabilidade a posteriori do teorema de Bayes,
    - O vencedor tem maior probabilidade a posteriori;
- Adaptação:
  - É proporcional à probabilidade a posteriori estimada do nodo vencedor e de seus vizinhos;

Centro de Informática UFPE

Modelo Estocástico

Bayesian Self-Organising Map – BSOM (2001)

- Aplicações:
  - Aprendizagem de mistura de gaussianas;
  - Reconhecimento de padrões;
  - Agrupamento.

Kernel SOM

- Uma função *kernel* é definida como κ: $X \times X \in \Re$, onde X é o espaço de entrada;
- Mais especificamente, a função de *kernel* é definida por um produto interno: $\kappa(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x}) \bullet \phi(\mathbf{y})$, onde $\phi(.)$ é uma função mapeadora usualmente não-linear e desconhecida,
  - $\phi : X \rightarrow F$, onde este é um espaço de caraacterísticas de produto interno de alta dimensão;
- Todas as operações são definidas em termos da função de kernel ao invés da função de mapeamento

U F P E

Centro de Informática UFPE

Kernel SOM

- Função objetivo: entropia conjunta das saídas da função de *kernel*:

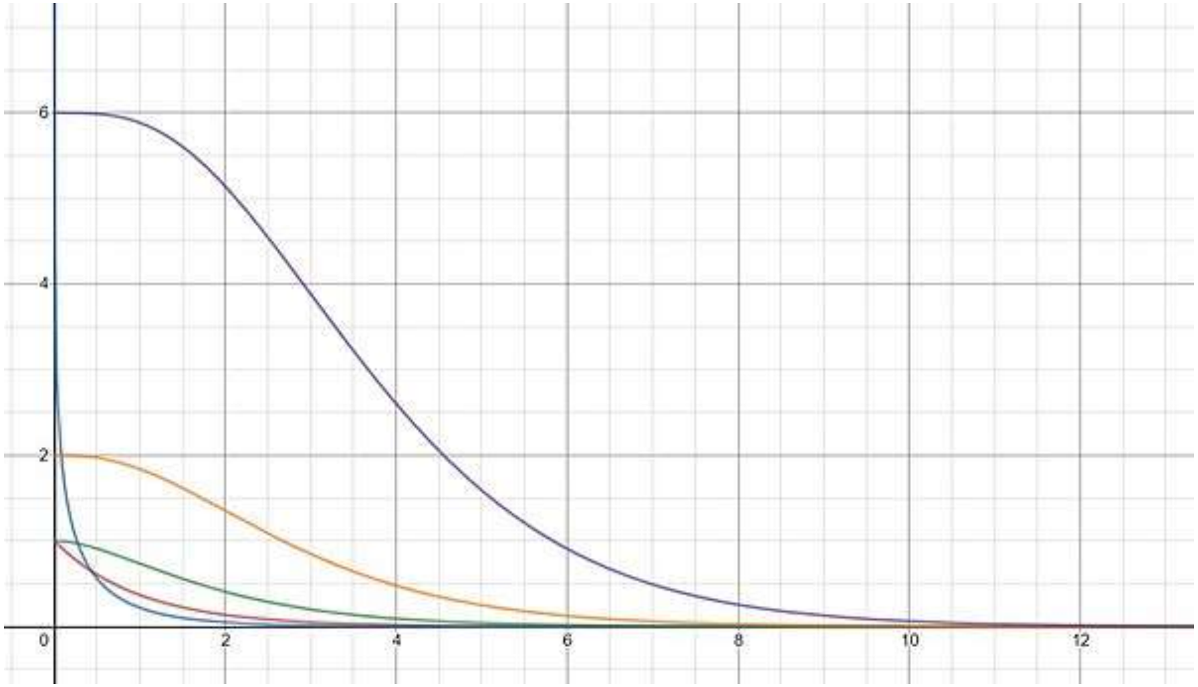
$$H(Y_i) = -\int_{-\infty} p_{Y_i}(y_i) \log p_{Y_i}(y_i) dy_i; \text{ onde } y_i = \kappa(\mathbf{x}, \mathbf{w}_i, \sigma_i)$$

- Função de *kernel* (distribuição gamma incompleta):

$$k(\mathbf{x}, \mathbf{w}_i, \sigma_i) = \frac{1}{\Gamma(\frac{m}{2})} \Gamma\left(\frac{m}{2}, \frac{\|\mathbf{x} - \mathbf{w}_i\|^2}{2\sigma_i^2}\right), \; i = 1, 2, \ldots, l$$

Kernel SOM

- Função de *kernel* (distribuição gamma incompleta):



Kernel SOM

- Determinação do vencedor:

$$i(\mathbf{x}) = \arg\min_{J} y_i = \arg\min_{j} k(\mathbf{x}, \mathbf{w}_i, \sigma_i)$$

- Função de vizinhança:

$$h_{j,i(\mathbf{x})} = \exp\left(-\frac{||\mathbf{x} - \mathbf{w}_j||^2}{2\sigma^2} \right), \quad j \in A$$

Kernel SOM

- Ajustes do peso e da dispersão:

$$\mathbf{w}_j(n+1) = \begin{cases} \mathbf{w}_j(n) + \dfrac{\eta_{\mathbf{w}} h_{j,i(\mathbf{x})}}{\sigma_j^2}(\mathbf{x}(n) - \mathbf{w}_j(n)), & j \in A \\ \mathbf{w}_j(n), & \text{de outra forma} \end{cases}$$

$$\sigma_j(n+1) = \begin{cases} \sigma_j(n) + \dfrac{\eta_\sigma h_{j,i(\mathbf{x})}}{\sigma_j(n)}\left[\dfrac{||(\mathbf{x}(n) - \mathbf{w}_j(n)||^2}{m\sigma_j^2(n)} - 1\right], & j \in A \\ \sigma_j(n), & \text{de outra forma} \end{cases}$$

Kernel SOM

Algoritmo de aprendizagem KSOM:

- Inicialize aleatoriamente os pesos $\mathbf{w}_j$ (0) com valores pequenos;
- Enquanto $H(Y_i)$ não convergir:
- Apresente a entrada $\mathbf{x}$;
- Encontre o vencedor $i(\mathbf{x})$;
- Atualize os pesos e as dispersões;
- Fim-do-enquanto;

Modelo Hierárquico

Growing Grid (1995)

- Contribuição:
  - Varia a tamanho da rede durante o treinamento;
- Arquitetura:
  - Semelhante ao SOM, mas com adição de linha ou coluna na grade;
- Aplicações:
  - Geração de mapas topológicos, visualização de dados.

UFPE

Centro de Informática UFPE

Modelo Hierárquico

Growing Hierarchical Self-Organizing Map – GHSOM (2000)

- Contribuição:
  - Um SOM hierárquico com topologia dinâmica;
  - O tamanho das sub-redes e a profundidade da hierarquia são determinados durante o processo de treinamento não-supervisionado.

Centro de Informática UFPE

Modelo Hierárquico

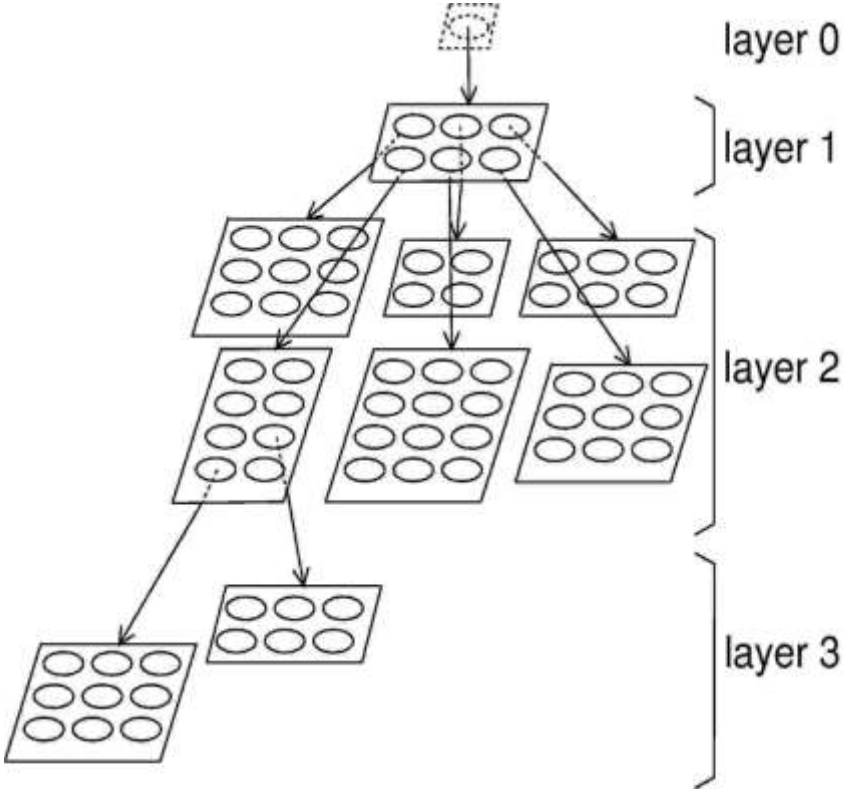
Growing Hierarchical Self-Organizing Map (2000)

- Estrutura:
  - Múltiplas camadas organizadas de forma hierárquica;
  - Cada camada possui uma quantidade independente de redes SOMs;
  - A primeira camada contem uma rede SOM apenas;
  - A rede SOM empregada é uma versão cujo crescimento acontece durante o processo de treinamento.

Modelo Hierárquico

Growing Hierarchical Self-Organizing Map (2000)

- Arquitetura:



Modelo Hierárquico

Growing Hierarchical Self-organizing Map (2000)

- Aplicação:
  - Recuperação da informação;
  - Organização de documentos;

Estrutura Variante no Tempo

Growing Neural Gas (GNG) (Fritzke, 1995)

- *Growing Neural Gas* (GNG) pode ser vista como:
  - Uma variante do *Growing Cell Structures* (GCS) sem preservação de topologia, ou
  - Uma variante incremental do *Topology Representing Networks* (TRN);
- Proposta:
  - Geração de grafo que represente a topologia da variedade (*manifold*) do espaço de entrada (aprendizagem de topologia);
  - A dimensionalidade do grafo depende da distribuição dos dados do espaço de entrada;

Estrutura Variante no Tempo

Growing Neural Gas (GNG)

- Regra de Aprendizagem:
  - As idades das conexões são usadas para remove-las quando se tornam inválidas durante o processo de aprendizagem (como na TRN);
  - Uma variável de erro associada a cada nodo é empregada para determinar onde inserir novos nodos (como na GCS);

Estrutura Variante no Tempo

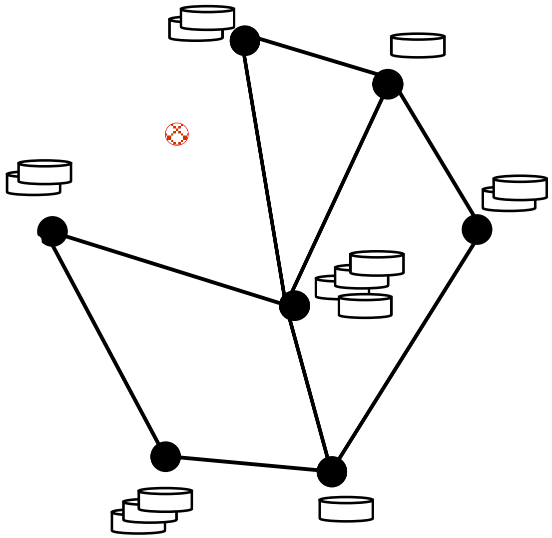
Learning Rule:

1. Start the map with two units s_a and s_b at random positions in $\boldsymbol{R}^n$

Centro de Informática UFPE

Estrutura Variante no Tempo

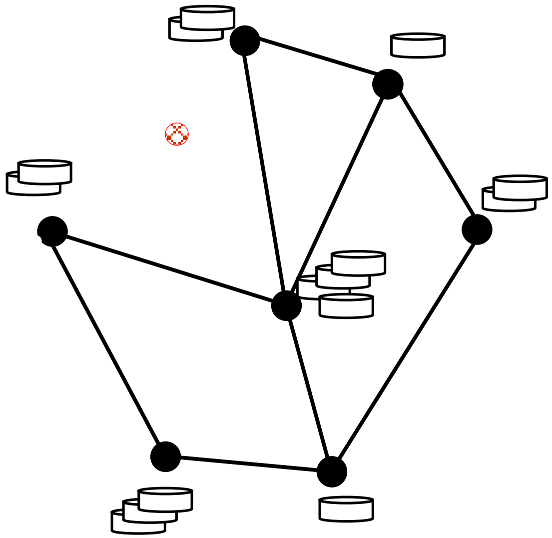
Learning Rule:



2. Generate an input signal ξ according to P(ξ)

Estrutura Variante no Tempo

Learning Rule:



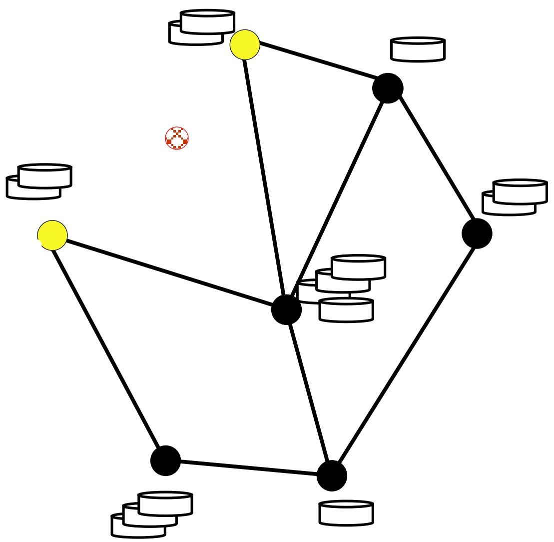
2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

$$\| \mathbf{w}_{s_1} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \quad \text{and}$$

$$\| \mathbf{w}_{s_2} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

Estrutura Variante no Tempo

Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

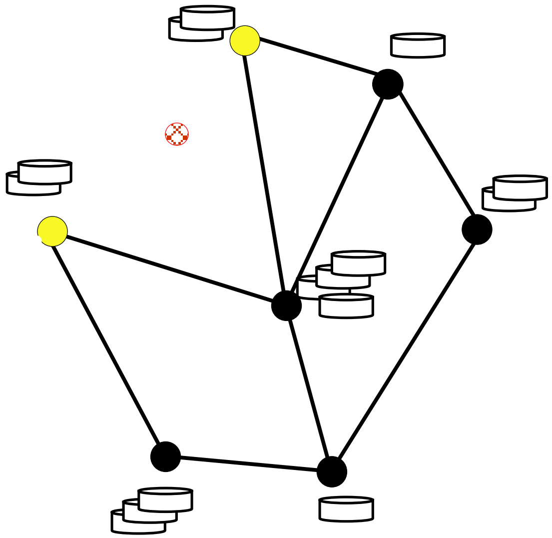
$$\| \mathbf{w}_{s_1} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \quad \text{and}$$

$$\| \mathbf{w}_{s_2} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

Competition

Estrutura Variante no Tempo

Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

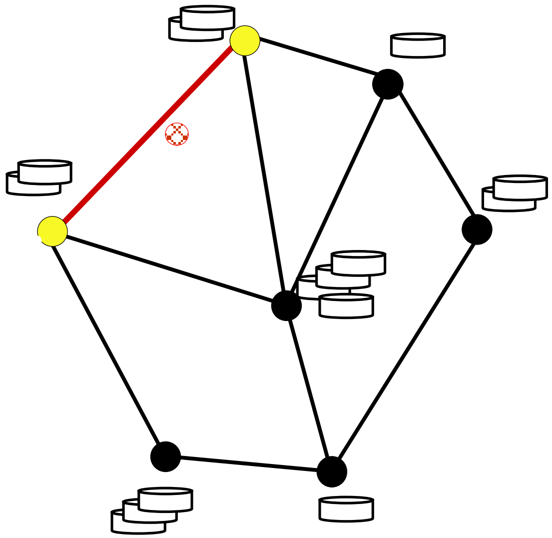
$$\|\mathbf{w}_{s_1} - \xi\| \leq \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A \qquad \text{and}$$

$$\|\mathbf{w}_{s_2} - \xi\| \leq \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero

Estrutura Variante no Tempo

Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

$$\| \mathbf{w}_{s_1} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A \qquad \text{and}$$

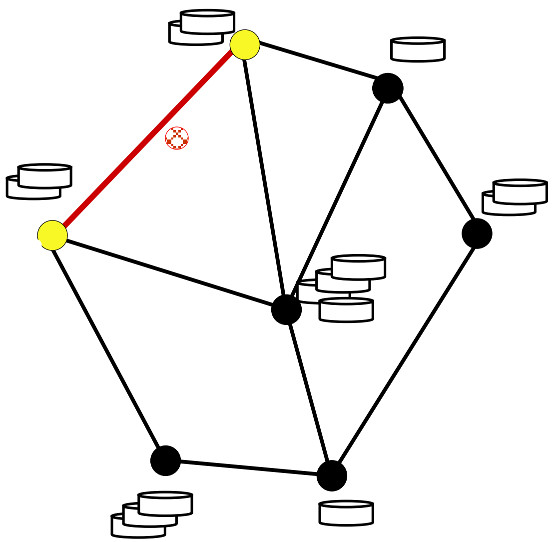
$$\| \mathbf{w}_{s_2} - \xi \| \leq \| \mathbf{w}_{s_i} - \xi \| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero

Topology Learning

Estrutura Variante no Tempo

Learning Rule:



2. Generate an input signal ξ according to $P(\xi)$
3. Determine the units s_1 e s_2 nearest to ξ

$$\|\mathbf{w}_{s_1} - \xi\| \le \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A \qquad \text{and}$$

$$\|\mathbf{w}_{s_2} - \xi\| \le \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero
5. Move s_1 and its direct topological neighbors towards ξ.

$$\Delta\mathbf{w}_{s_1} = \varepsilon_b(\xi - \mathbf{w}_{s1})$$

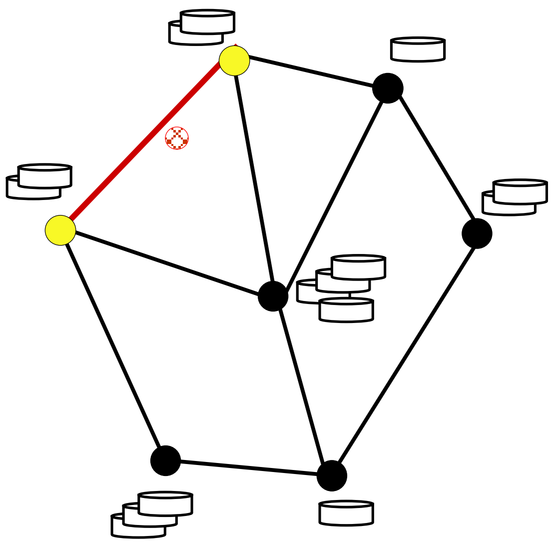
$$\Delta\mathbf{w}_{s_i} = \varepsilon_n(\xi - \mathbf{w}_{s_i}) \quad \forall s_i \in N_{s_1},$$

UFPE

Centro de Informática UFPE

Estrutura Variante no Tempo

Learning Rule:



2. Generate an input signal ξ according to P(ξ)
3. Determine the units s_1 e s_2 nearest to ξ

$$\|\mathbf{w}_{s_1} - \xi\| \le \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A \qquad \text{and}$$

$$\|\mathbf{w}_{s_2} - \xi\| \le \|\mathbf{w}_{s_i} - \xi\| \qquad \forall s_i \in A - \{s_1\}$$

4. If it does not already exist, insert a connection between s_1 and s_2. In any case, set the age of the connection between s_1 and s_2 to zero
5. Move s_1 and its direct topological neighbors towards ξ.

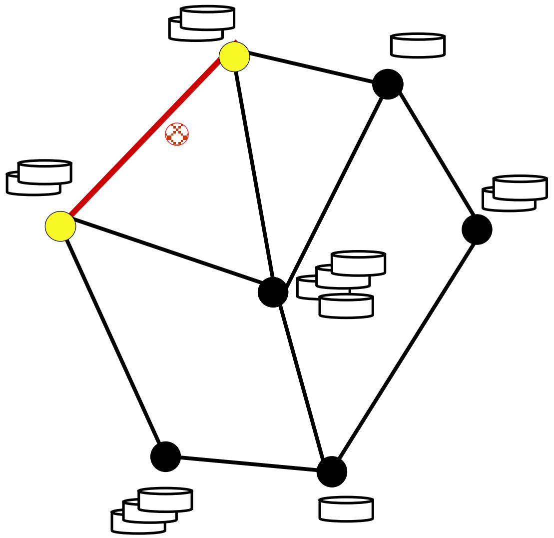
$$\Delta\mathbf{w}_{s_1} = \varepsilon_b(\xi - \mathbf{w}_{s1})$$

$$\Delta\mathbf{w}_{s_i} = \varepsilon_n(\xi - \mathbf{w}_{s_i}) \quad \forall s_i \in N_{s_1},$$

Competition and Collaboration

Estrutura Variante no Tempo

Learning Rule:

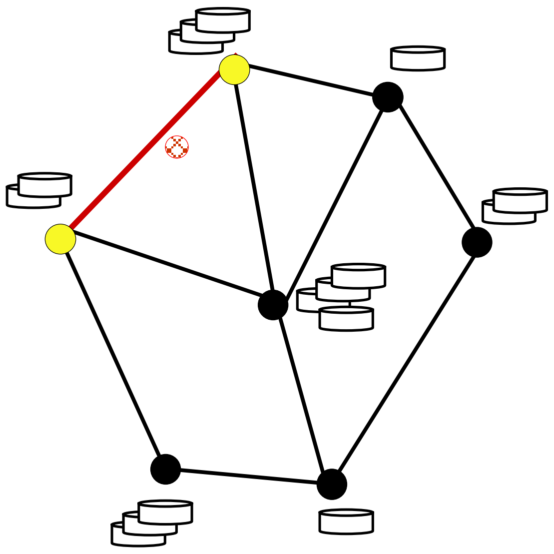


6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \boldsymbol{\xi} \|^2$$

Estrutura Variante no Tempo

Learning Rule:



6. Add ΔE_{s_1} to the s_1 local error variable

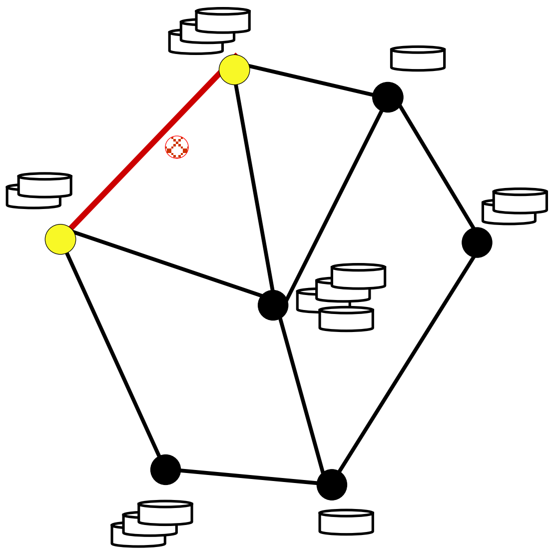
$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \boldsymbol{\xi} \|^2$$

Centro de Informática UFPE

UFPE

Estrutura Variante no Tempo

Learning Rule:



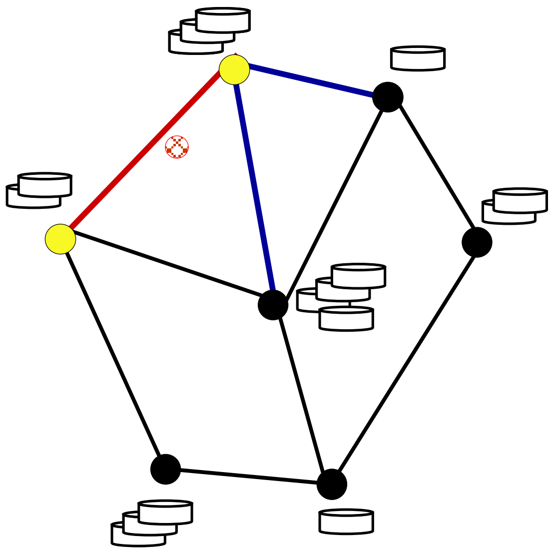
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \boldsymbol{\xi} \|^2$$

7. Increase by one the age of all edges emanating from s_1.

Estrutura Variante no Tempo

Learning Rule:



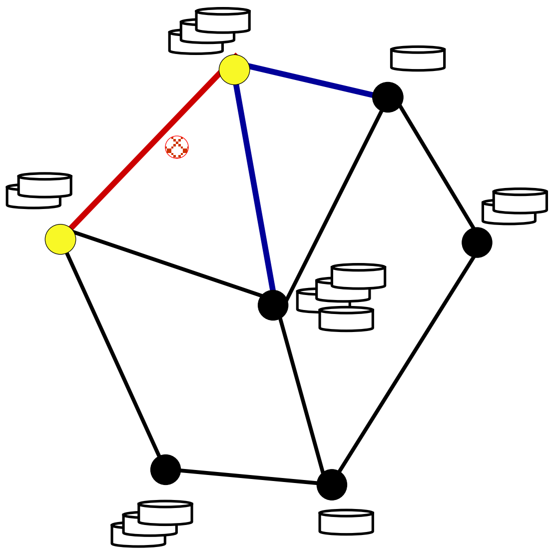
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

7. Increase by one the age of all edges emanating from s_1.

Estrutura Variante no Tempo

Learning Rule:



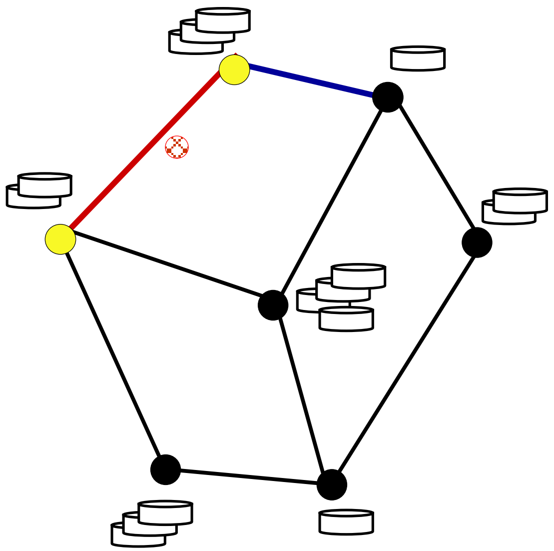
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

7. Increase by one the age of all edges emanating from s_1.
8. Remove the connections with an age larger then a threshold (a_{max})

Estrutura Variante no Tempo

Learning Rule:



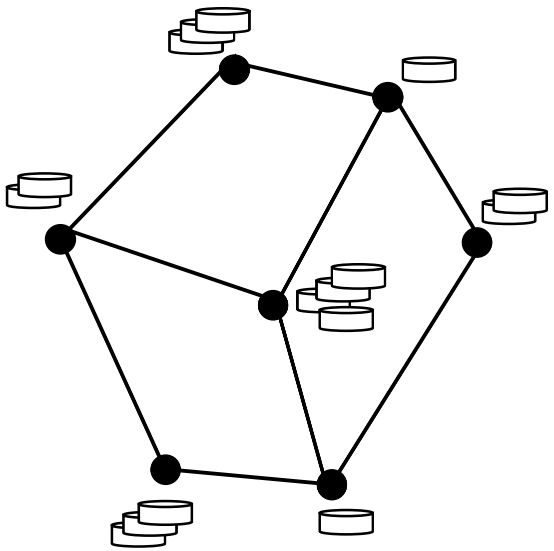
6. Add ΔE_{s_1} to the s_1 local error variable

$$\Delta E_{s_1} = \| \mathbf{w}_{s_1} - \xi \|^2$$

7. Increase by one the age of all edges emanating from s_1.
8. Remove the connections with an age larger then a threshold (a_{max})
   - If this results in units having no emanating edges, remove them as well.

Estrutura Variante no Tempo

Learning Rule:

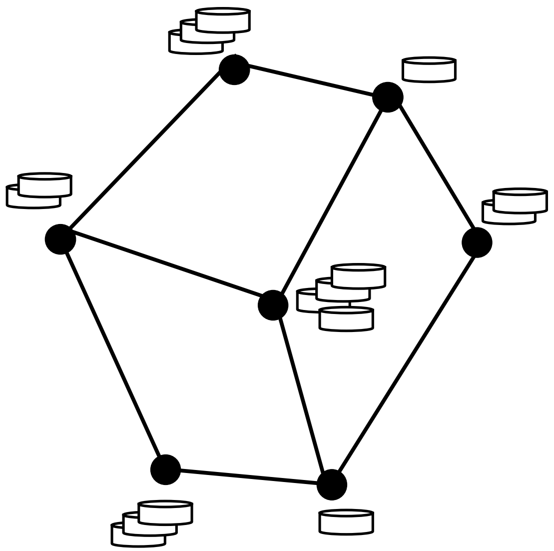


9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.

Growing Step

Estrutura Variante no Tempo

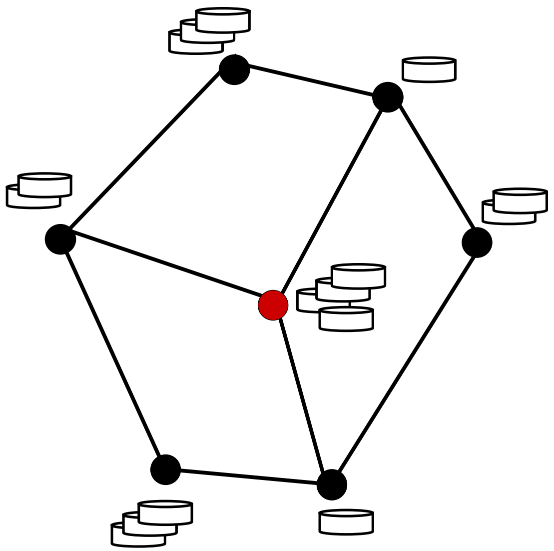
Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.

Estrutura Variante no Tempo

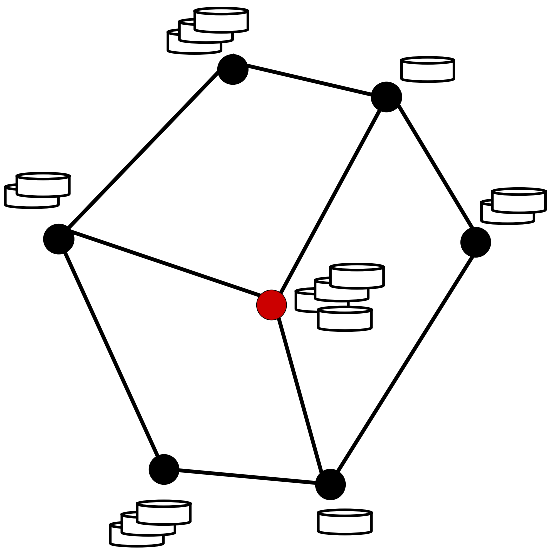
Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.

Estrutura Variante no Tempo

Learning Rule:

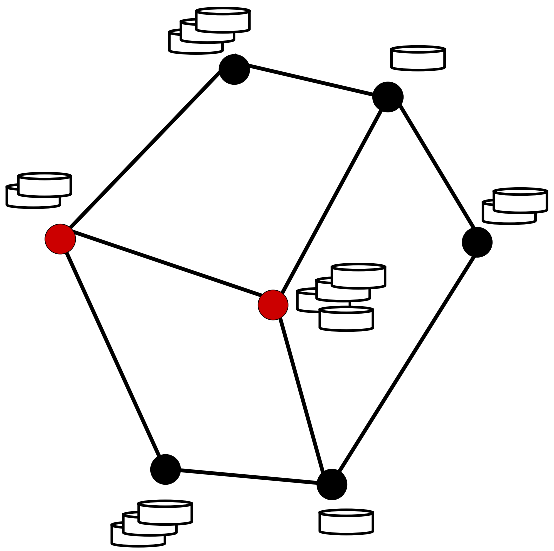


9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f , topological neighbor of $s_{q,}$ with the largest error variable.

Centro de Informática UFPE

Estrutura Variante no Tempo

Learning Rule:

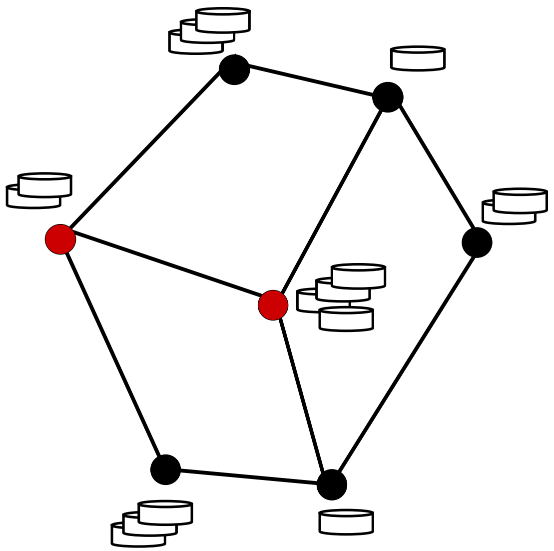


9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f , topological neighbor of $s_{q,}$ with the largest error variable.

Centro de Informática UFPE

Estrutura Variante no Tempo

Learning Rule:



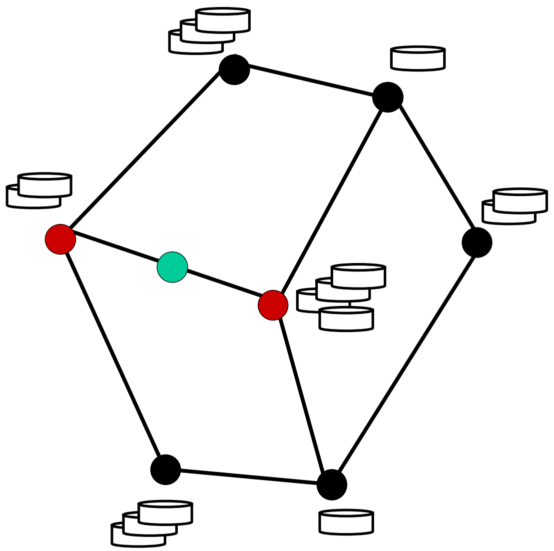
9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f , topological neighbor of s_q, with the largest error variable.
   - Insert a new unit (s_r) between s_q and s_f.

Centro de Informática UFPE

U F P E

Estrutura Variante no Tempo

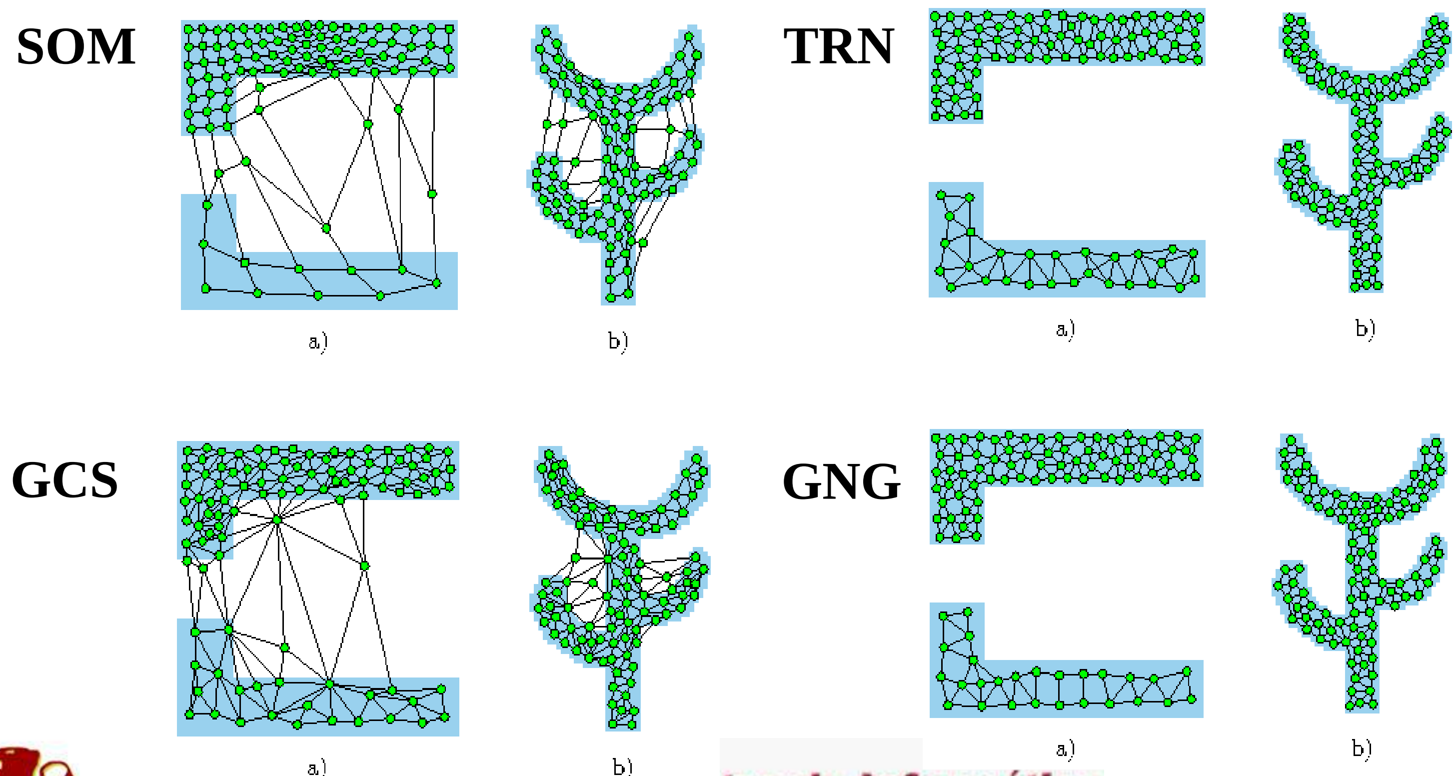
Learning Rule:



9. If the number of input signals generated so far is an integer multiple of a parameter λ, insert a new unit as follows.
   - Determine the node s_q with the maximum accumulated error.
   - Determine the node s_f, topological neighbor of s_q, with the largest error variable.
   - Insert a new node (s_r) between s_q and s_f
   - Insert edges connecting the new node s_r with nodes s_q and s_f , and remove the original edge between sq and s_f.

Comparação de Resultados

Mapeamentos criados pelo SOM e suas variantes:



Referências

Dittenbach, M., Merkl, D., & Rauber, A. (2000). The growing hierarchical self-organizing map. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 6, 15-19.

Fritzke, B. (1995a). A growing neural gas network learns topologies. *Advances in Neural Information Processing Systems*, 7: 625-632.

Fritzke, B. (1996). Unsupervised ontogenetic networks. *Handbook of Neural Computation*, IOP Publishing and Oxford University Press.

López-Rubio, E., Muñoz-Pérez, J., & Gómez-Ruiz, J. A. (2004). A principal components analysis self-organizing map. *Neural networks*, 17(2), 261-70.

Ontrup, J., & Ritter, H. (2001). Hyperbolic Self-Organizing Maps for Semantic Navigation. *Proceedings of the 14th International Conference on Neural Information Processing Systems*, 1417–1424.

Referências

Villa, N., & Rossi, F. (2007). A comparison between dissimilarity SOM and kernel SOM for clustering the vertices of a graph. *Proceedings of the International Workshop on Self-Organizing Maps*.

Walter, J. A. (1998). PSOM Network : Learning with few examples. *PSOM network: learning with few examples. Proceedings. 1998 IEEE International Conference on Robotics and Automation*, 1-6.

Yin, H., & Allinson, N. M. (2001). Bayesian self-organising map for Gaussian mixtures. *IEE Proceedings-Vision, Image and Signal Processing*, 148(4), 234-240.

https://ni.www.techfak.uni-bielefeld.de/files/Ritter1999-SOM.pdf