

## *Neural Gas*

- O método de auto-organização conhecido como *neural gas network* (NG), proposto por MARTINETZ & SCHULTEN (1991), difere dos mapas auto-organizáveis de Kohonen em **três** aspectos fundamentais:
  - ✓ Em vez de usar um arranjo topológico fixo de unidades, a NG automaticamente cria e atualiza uma matriz de conectividade entre as unidades, a qual espelha, até certo ponto, as potenciais similaridades existentes no espaço dos dados de entrada – o termo *gas* vem justamente desta propriedade;
  - ✓ Cada padrão de entrada apresentado à rede faz com que todos os vetores de pesos sejam ajustados;

- ✓ A extensão de tais modificações não é determinada por uma geometria de arranjo, mas pelas distâncias relativas entre as unidades neurais no espaço de entrada.
- Seja  $\mathbf{p}(t)$  o padrão atualmente apresentado na entrada da rede. Para cada unidade  $i$ , o número  $k_i$  de unidades cujos vetores de pesos estão mais próximos do padrão de entrada que seu vetor de pesos (i.e.,  $\mathbf{w}_i$ ) é determinado.
- Em termos matemáticos,  $k_i = |A|$ , onde  $|\cdot|$  é a cardinalidade de um conjunto e  $A = \{m \in \mathbb{N}: |\mathbf{p}(t) - \mathbf{w}_m(t)| < |\mathbf{p}(t) - \mathbf{w}_i(t)|\}$ .
- De maneira similar,  $i_r$  identifica a unidade da rede para a qual existem  $r$  unidades cujos vetores de pesos estão mais próximos de  $\mathbf{p}(t)$  quando comparados a  $\mathbf{w}_{i_r}$ . Consequentemente,  $k_{i_r} = r$ .
- Então, cada vetor de pesos é adaptado de acordo com a seguinte expressão:

$$\mathbf{w}_i(t + 1) = \mathbf{w}_i(t) + \mu e^{-\frac{k_i}{\lambda}} (\mathbf{p}(t) - \mathbf{w}_i(t)),$$

onde  $\mu$  é o tamanho do passo e  $\lambda$  determina o número de unidades cujos vetores de pesos são significativamente modificados em cada iteração.

- A NG é capaz de aprender possíveis relações de vizinhança entre vetores de pesos através da atualização de uma matriz de conectividade  $\mathbf{C} \in \mathbb{R}^{n \times n}$ : assim que um padrão de entrada é apresentado e todos os vetores de pesos são apropriadamente modificados, uma conexão é criada entre a unidade  $i_0$ , cujo vetor de pesos tem a menor distância em relação ao padrão de entrada, e a unidade  $i_1$ , cujo vetor de pesos é o segundo mais próximo a  $\mathbf{p}(t)$ , trocando-se o valor de  $\mathbf{C}_{i_0, i_1}$  de zero para um.
- Além disso, cada conexão  $(i, j)$  possui uma idade  $t_{(i, j)}$  que pode ser alterada de três formas diferentes:

- ✓ Quando a idade de uma conexão  $(i,j)$  excede um tempo de vida máximo  $T$ , a conexão é removida, i.e.,  $\mathbf{C}_{i,j}$  recebe o valor zero;
  - ✓ Se a conexão  $\mathbf{C}_{i_0,i_1}$  que o algoritmo está tentando criar já existe, a idade correspondente  $t_{(i_0,i_1)}$  é reinicializada para zero;
  - ✓ A idade de todas as conexões que a unidade  $i_0$  possui é incrementada após a apresentação de cada padrão.
- 
- É conveniente reduzir de maneira progressiva o parâmetro  $\lambda$  e o tamanho do passo  $\mu$  à medida que as épocas de treinamento são realizadas a fim de promover um ajuste fino, envolvendo apenas um pequeno conjunto de vetores de pesos mais próximos ao padrão de entrada, no final do processo.
  - Em contrapartida, aumentando progressivamente o valor de  $T$ , o algoritmo esquece com maior rapidez as conexões no início do processo, enquanto, mais próximo do

final, as conexões são mantidas por um período mais longo uma vez que os vetores de pesos já devem ter atingido uma configuração adequada no espaço de entrada.

- Por isso, após todos os padrões de entrada serem apresentados para a rede, estes parâmetros são ajustados de acordo com a seguinte expressão:

$$q(t) = q_i \left( \frac{q_f}{q_i} \right)^{t/N_T},$$

onde  $q \in \{\mu, \lambda, T\}$ ,  $q_i$  e  $q_f$  são os valores inicial e final do parâmetro, respectivamente, e  $N_T$  representa o número de épocas de treinamento.

## Exemplos:

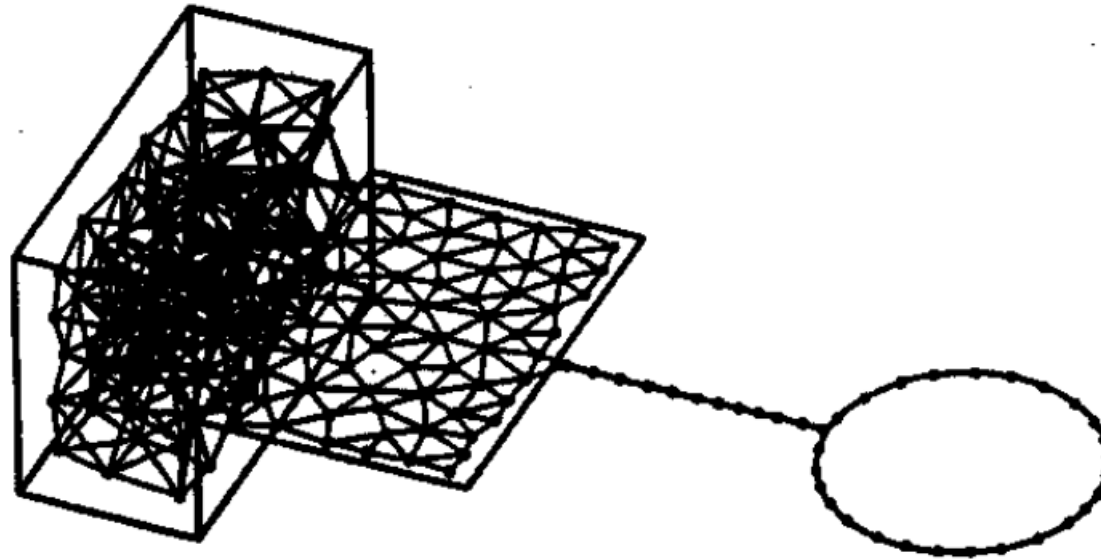
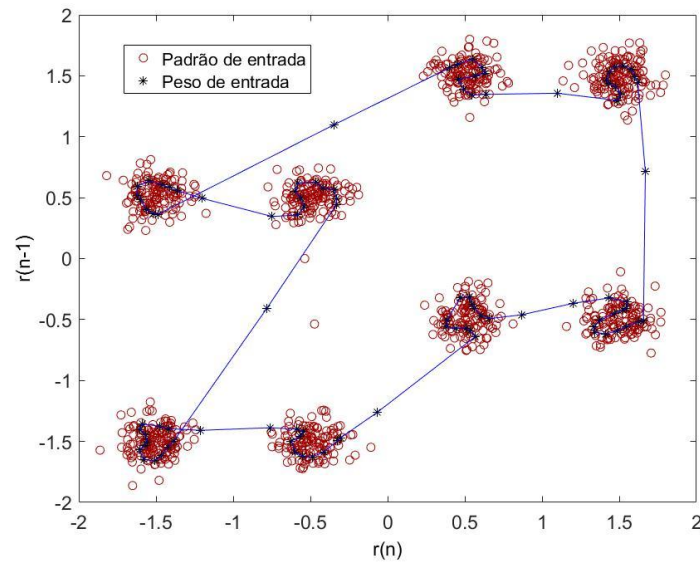
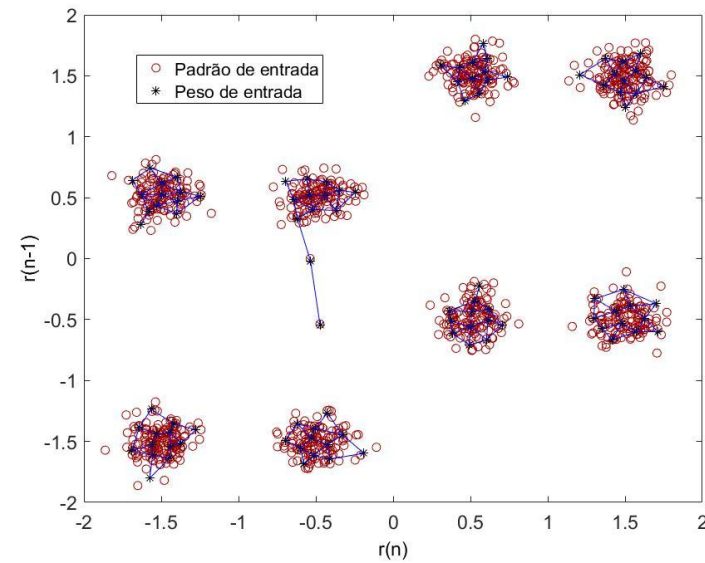


Figura extraída de MARTINETZ & SCHULTEN (1991), mostrando como as conexões entre as unidades refletem a estrutura topológica e a dimensionalidade do espaço (*manifold*) dos dados.



(a) SOM



(b) NG

Figuras extraídas de BOCCATO (2013).

### *Growing neural gas*

- Em 1995, Fritzke propôs uma estratégia inspirada na NG capaz de automaticamente ajustar o número de unidades.

- A ideia principal do método, conhecido como *growing neural gas*, é acrescentar novas unidades a uma rede inicialmente pequena ao avaliar medidas estatísticas locais durante os passos de adaptação.
- Agora, cada unidade  $i$  está associada a um erro acumulado, denotado por  $E_i$ , o qual fornece um indicativo do erro cometido ao utilizarmos  $i$  para representar os padrões de entrada mais próximos a ela.
- Abaixo, segue uma descrição do algoritmo associado ao *growing neural gas*.

**Passo 0:** comece com duas unidades  $a$  e  $b$  em posições aleatórias do espaço de entrada ( $\mathbf{w}_a$  e  $\mathbf{w}_b$ , respectivamente).

**Passo 1:** Seja  $\mathbf{p}(t)$  o padrão apresentado na iteração  $t$ . Encontre as duas unidades mais próximas de  $\mathbf{p}(t)$  – i.e.,  $i_0$  e  $i_1$ .

**Passo 2:** incremente a idade de todas as arestas emanando de  $i_0$ .

**Passo 3:** acrescente a distância entre o padrão de entrada ( $\mathbf{p}(t)$ ) e a unidade mais próxima ( $i_0$ ) ao erro acumulado correspondente:

$$E_{i_0} = E_{i_0} + \|\mathbf{p}(t) - \mathbf{w}_{i_0}(t)\|^2.$$



**Passo 4:** mova  $i_0$  e seus vizinhos topológicos imediatos na direção do padrão  $\mathbf{p}(t)$ :

$$\Delta w_{i_0}(t) = \mu_a(\mathbf{p}(t) - \mathbf{w}_{i_0}(t))$$

$$\Delta w_j(t) = \mu_b(\mathbf{p}(t) - \mathbf{w}_j(t)),$$

para os vizinhos  $j$  de  $i_0$ .

**Passo 5:** se a conexão entre  $i_0$  e  $i_1$  já existe, faça  $t_{(i_0, i_1)} = 0$ . Caso contrário, crie a conexão.

**Passo 6:** remova as arestas com idade superior a  $T$ . Se isto der origem a pontos a partir dos quais não saem arestas, remova-os da rede.

**Passo 7:** se o número de padrões já apresentados for um múltiplo inteiro de um parâmetro  $\kappa$ , insira uma nova unidade da seguinte forma:

- Determine a unidade  $q$  com o maior erro acumulado.
- Insira uma nova unidade  $r$  no ponto médio entre  $q$  e seu vizinho  $f$  com maior erro acumulado:

$$\mathbf{w}_r = 0,5(\mathbf{w}_q + \mathbf{w}_f).$$

- Insira arestas conectando a nova unidade  $r$  com as unidades  $q$  e  $f$ ; remova a aresta original entre  $q$  e  $f$ .
- Reduza os erros acumulados de  $q$  e  $f$  multiplicando-os por uma constante  $\alpha$ .

➤ Inicialize o erro associado à unidade  $r$  com uma cópia do novo valor do erro acumulado  $E_q$ .

**Passo 8:** reduza todos os erros acumulados por um fator multiplicativo  $d$ ;

**Passo 9:** se o critério de parada (e.g., tamanho da rede ou alguma medida de desempenho) não tiver sido atingido, retorne ao passo 1.

- O acúmulo das distâncias quadráticas durante a adaptação (Passo 3) ajuda a identificar unidades situadas em regiões do espaço de entrada nas quais o mapeamento dos padrões para as unidades causa bastante erro. A fim de reduzir este erro, novas unidades são inseridas exatamente nestas regiões.

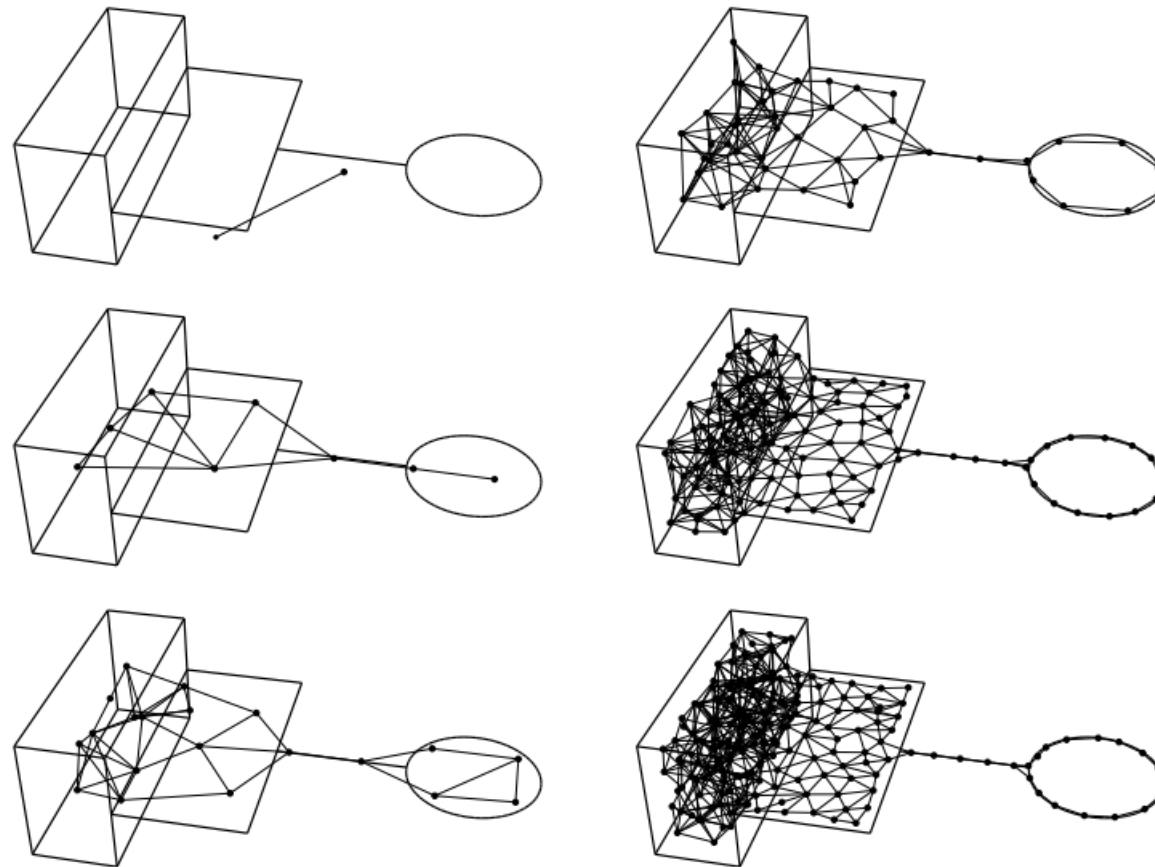


Figura extraída de FRITZKE (1995), mostrando a evolução da rede. Os parâmetros utilizados foram  $\lambda = 100$ ,  $\mu_a = 0,2$ ,  $\mu_b = 0,006$ ,  $\alpha = 0,5$ ,  $T = 50$  e  $d = 0,995$ .

## Referências

- BOCCATO, L. “Novas propostas e aplicações de redes neurais com estados de eco”, Tese de Doutorado, Faculdade de Engenharia Elétrica e de Computação – Universidade Estadual de Campinas, 2013.
- MARTINETZ, T. & SCHULTEN, K. “A neural gas network learns topologies”. Em *Artificial Neural Networks*, vol. I, pp. 397-402, Elsevier, 1991.
- FRITZKE, B. “A growing neural gas network learns topologies”, Em *Advances in Neural Information Processing Systems*, vol. 7, pp. 625-632, MIT Press, 1995.