

Redes Auto-Organizáveis

Computação Natural

Gisele L. Pappa

Aprendizagem Supervisionada

- A classe a que os exemplos de entrada pertencem é conhecida
- Durante o treinamento, a saída da rede é comparada com a saída esperada (classe prevista pela rede *vs* a classe conhecida), e um erro é gerado
- Redes supervisionadas normalmente respondem a seguinte pergunta:
 - A que classe pertence esse dado de entrada?

Aprendizagem não-supervisionada

- A classe a que os exemplos de entrada pertencem não é conhecida
- Medidas de distância são utilizadas para treinar a rede
 - Os neurônios da rede são posicionados no espaço dos dados
 - O erro é dado pela distância entre a entrada e os neurônios

Aprendizagem não-supervisionada

- Redes não-supervisionadas normalmente respondem as seguintes perguntas:
 - Que grupos existem nesses dados ?
 - Como cada entrada de dados está relacionada a base de dados como um todo?
- Redes não-supervisionadas são utilizadas para resolver tarefas mais descritivas

Redes Não-supervisionadas

- ART
 - Resolve um dos maiores problemas de RNA: incapacidade de aprender material novo preservando o material antigo
 - Trabalha em dois estados:
 - De aprendizagem
 - De estabilidade
- Mapas auto-organizáveis
 - Redução de dimensionalidade
 - Útil para **visualização de dados contínuos n-dimensionais**

Redes Auto-Organizáveis (SOM)

- Também conhecidas como redes de Kohonen (1982)
- Aprendizagem não-supervisionada
- Seguem o **conceito** de redes feedforward de uma camada
 - Porém, para entendê-las melhor, esqueça a função de ativação, o feedforward e o backpropagation ☺

Redes Auto-Organizáveis (SOM)

- Tem como principal objetivo **transformar** um padrão de entrada de dimensão m em um mapa discreto de uma ou duas dimensões
- **Topologicamente ordenado**
 - Existe um mapeamento dos dados de entrada para o mapa de forma que cada ponto no mapa representa uma região do espaço de dados
 - Existe uma motivação biológica para isso

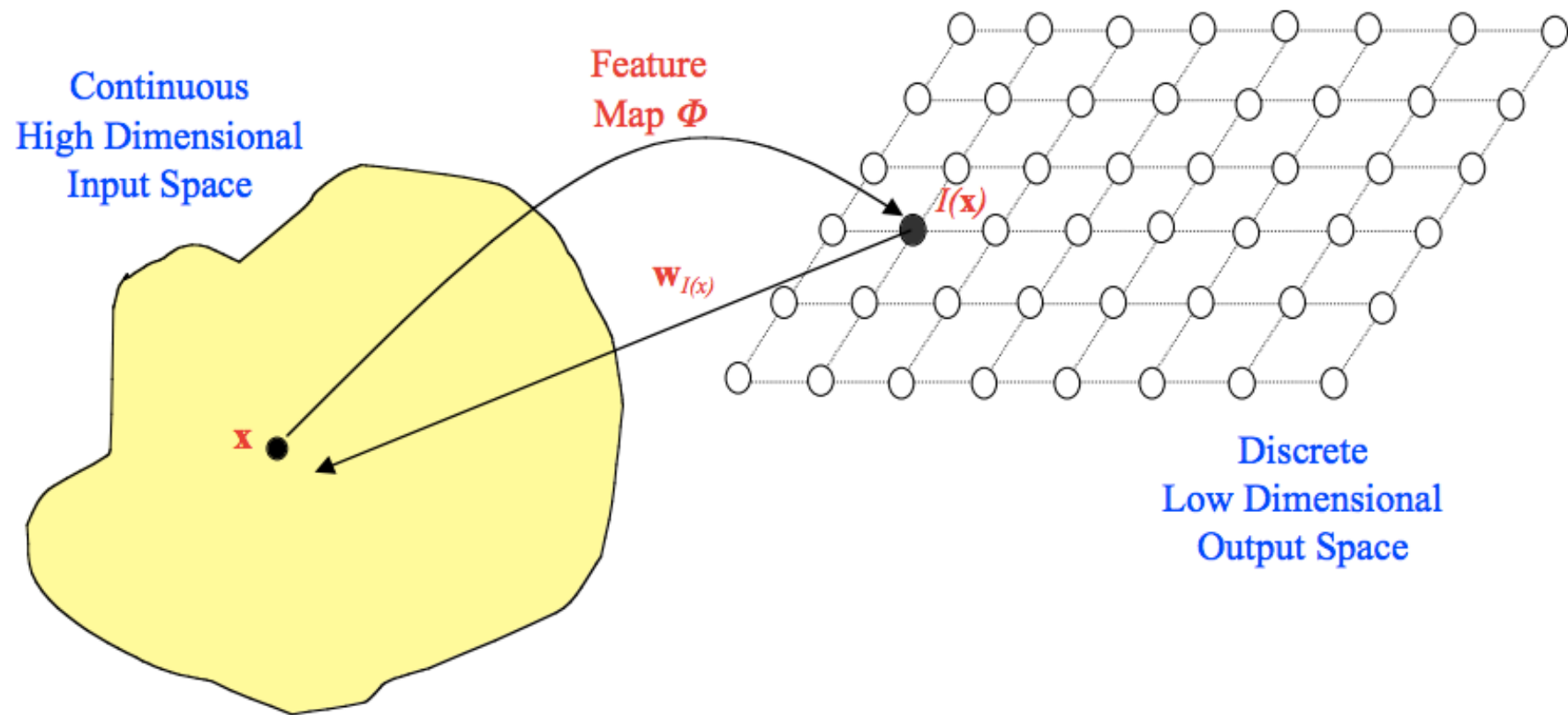
Motivação Biológica

- Córtex cerebral
 - Contém áreas especializadas na fala, visão, audição e outras funções motoras
 - Cada área exibe uma lógica relacionada a ordem de seu funcionamento
 - Ex: mapa tonotópico – Neurônios vizinhos respondem a frequências de som similares numa sequência ordenada da mais alta para o mais baixa
 - Quais são os mecanismos que fazem com que essa ordem apareça naturalmente?

Estrutura da Rede

- Cada nó está associado a uma posição topológica específica, representada por uma coordenada x,y
- A cada coordenada está associado um vetor de pesos, com o mesmo número de dimensões dos dados de entrada
- Vetor de entrada $V = (v_1, v_2, v_3, \dots, v_d) \Rightarrow$ vetor de pesos $W = (w_1, w_2, w_3, \dots, w_d)$

Mapeamento

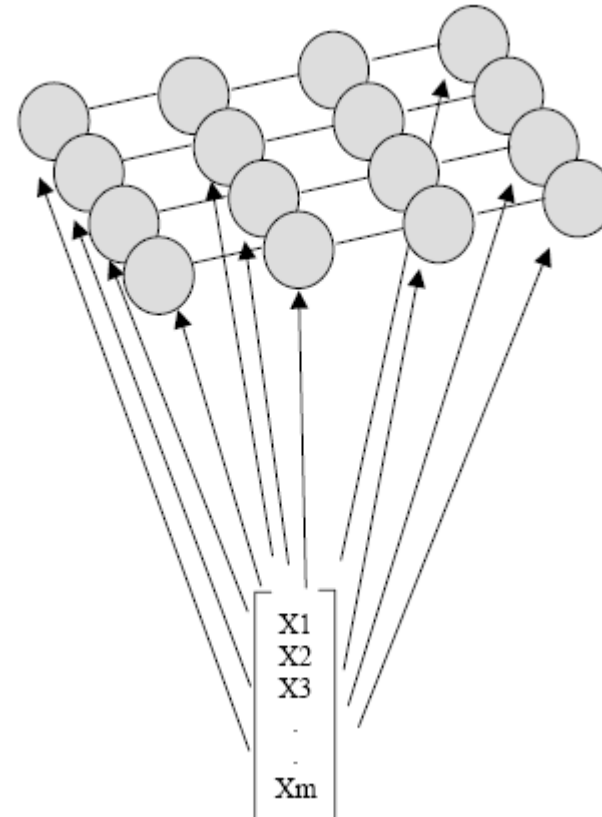
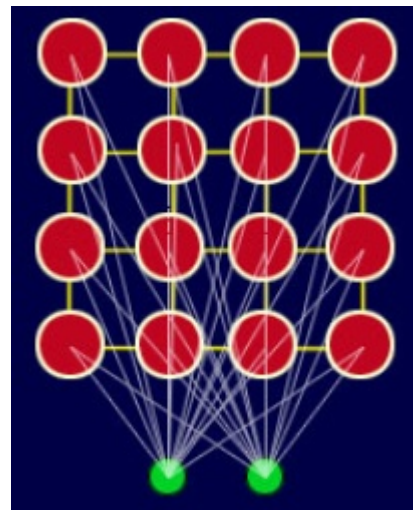


Estrutura da SOM

- Neurônios estão conectados em um *grid* 1D ou 2D
 - É esse grid que **vai representar os dados**
- Durante o treinamento, os neurônios se movem no *grid* tentando se ajustar aos dados
 - Dois espaços de dados
- Quando a posição de um neurônio muda, a posição dos neurônios vizinhos a este também são modificadas

Estrutura da Rede

- Todas as entradas estão conectadas por pesos a cada neurônio
- Existem conexões laterais *virtuais* entre nós vizinhos



Estrutura da Rede

- Cada nó está associado a uma posição topológica específica, representada por uma coordenada x,y
- A cada coordenada está associado um vetor de pesos, com o mesmo número de dimensões que os dados de entrada
- Vetor de entrada $V = (v_1, v_2, v_3, \dots, v_d) \Rightarrow$ vetor de pesos $W = (w_1, w_2, w_3, \dots, w_d)$

Algoritmo de Aprendizado

- Os pesos dos neurônios de entrada são inicializados aleatoriamente com valores pequenos
- A formação do mapa ocorre em 3 fases:
 - **Competição**
 - Para cada padrão de entrada, os neurônios vão competir entre si, e um (ou um grupo) será vencedor
 - **Cooperação**
 - O vencedor determina a região espacial (vizinhança) que será estimulada
 - **Adaptação**
 - Pesos do neurônio vencedor e seus vizinhos são atualizados

Competição

- Um padrão de entrada $X = \{x_1, x_2, \dots, x_n\}$ é apresentado para todos os neurônios da rede
- Para determinar o neurônio vencedor, calculamos a distância entre o vetor de pesos W de um neurônio e a entrada X

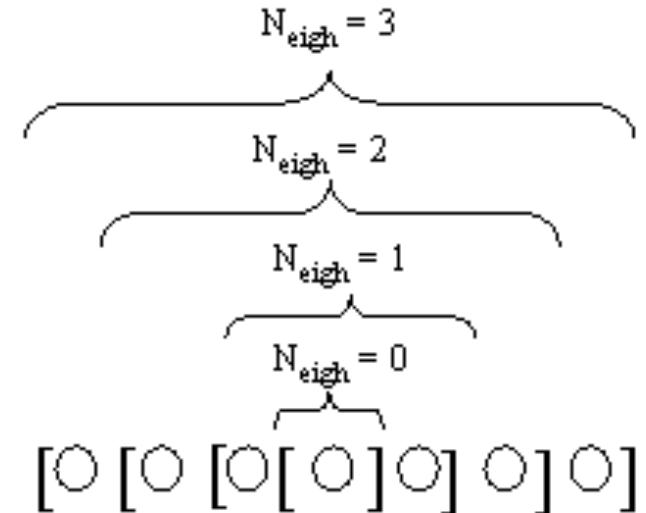
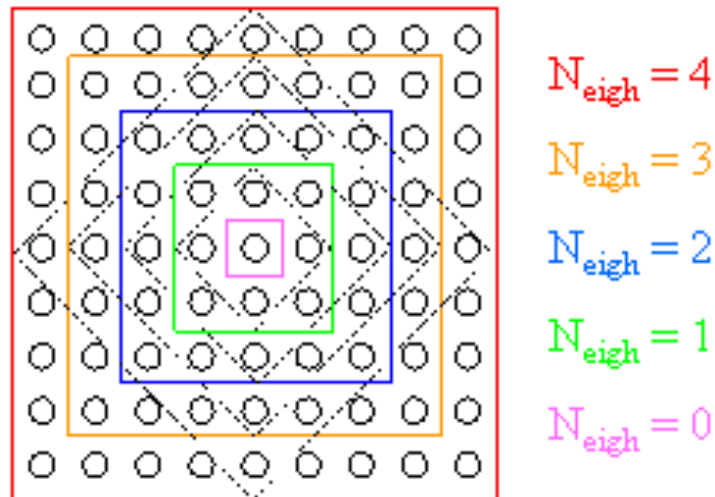
$$venc(\mathbf{x}) = \arg \min_j \|\mathbf{x} - \mathbf{w}_j\|, j = 1, \dots, n$$

Cooperação

- Uma vizinhança é definida ao redor do neurônio vencedor
 - A cada iteração, apenas os pesos (posição) dos neurônios dentro dessa vizinhança são alterados
 - A largura ou o raio de vizinhança é um parâmetro da rede, e varia com o tempo

Cooperação

- O neurônio vencedor (i) é considerado o centro da vizinhança topológica



Cooperação

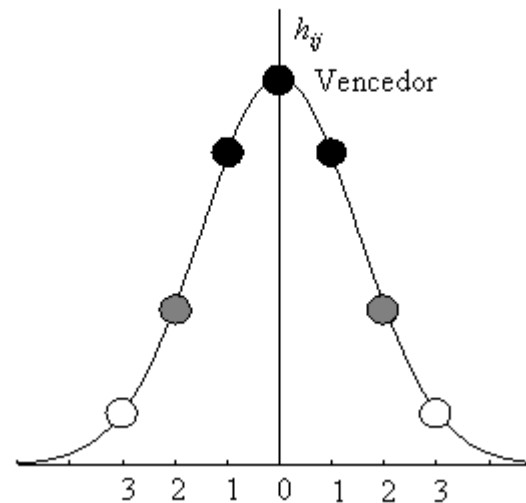
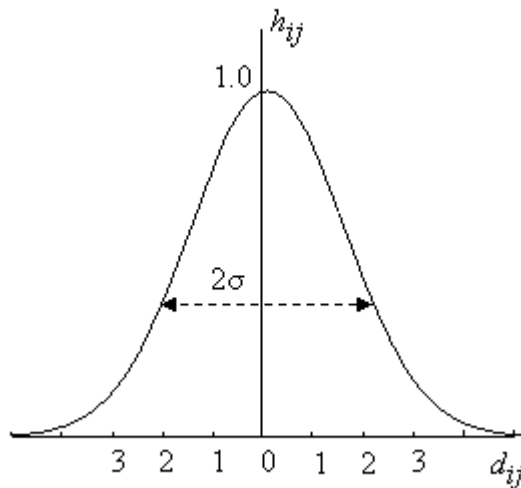
- Uma função de kernel k_{ij} define a influência de um neurônio sobre seus vizinhos

$$h_{ji} = \exp\left(-\frac{d_{ji}^2}{2\sigma^2}\right)$$

1-D, $d = ||j - i||$

2-D, $d = ||r_j - r_i||^2$, r é a posição do neurônio

onde σ é a largura da vizinhança



Cooperação

- A largura ou o raio de vizinhança varia com o tempo
 - Começa com uma largura grande e diminui exponencialmente com o tempo

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{t_1}\right)$$

onde t_1 é uma constante de tempo e n é a época.

- O valor σ_0 é definido pelo usuário, e depende da abrangência da vizinhança definida.

Adaptação

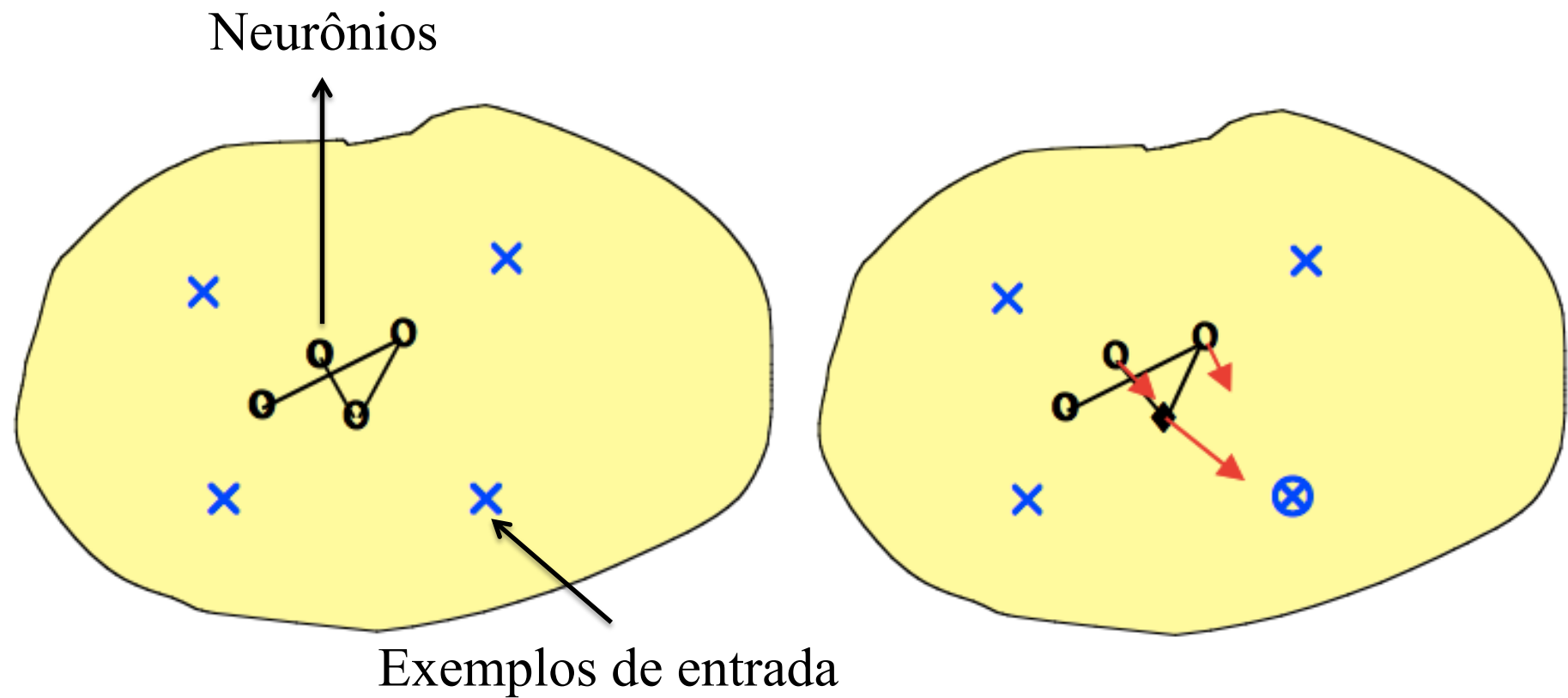
- Atualizar os pesos do vencedor e sua vizinhança de forma que eles fiquem mais parecidos com os dados

$$w_j(n+1) = w_j(n) + \overset{\text{Taxa de aprendizado}}{\eta(n)} \underset{\text{Função de kernel}}{h_{ji}(n)} (x - w_j(n))$$

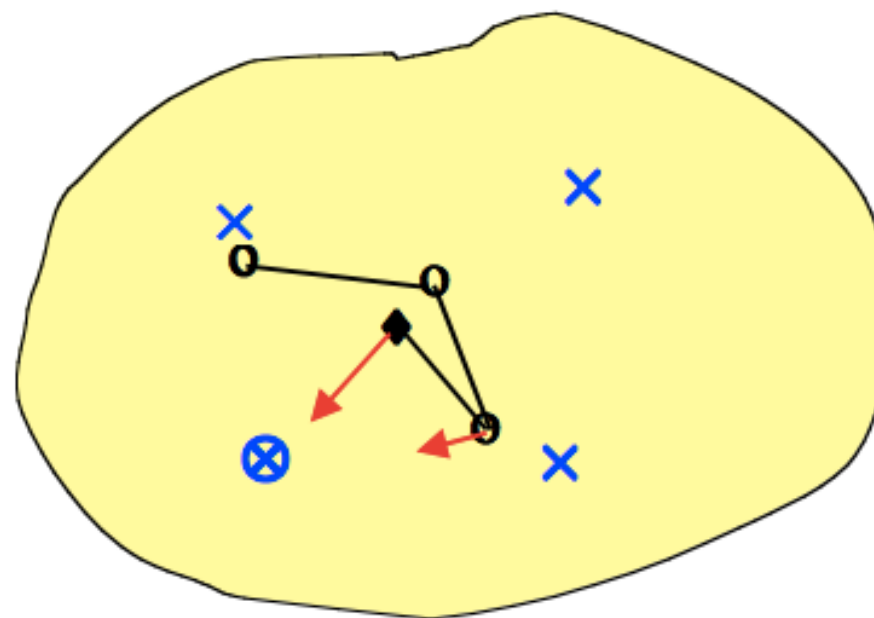
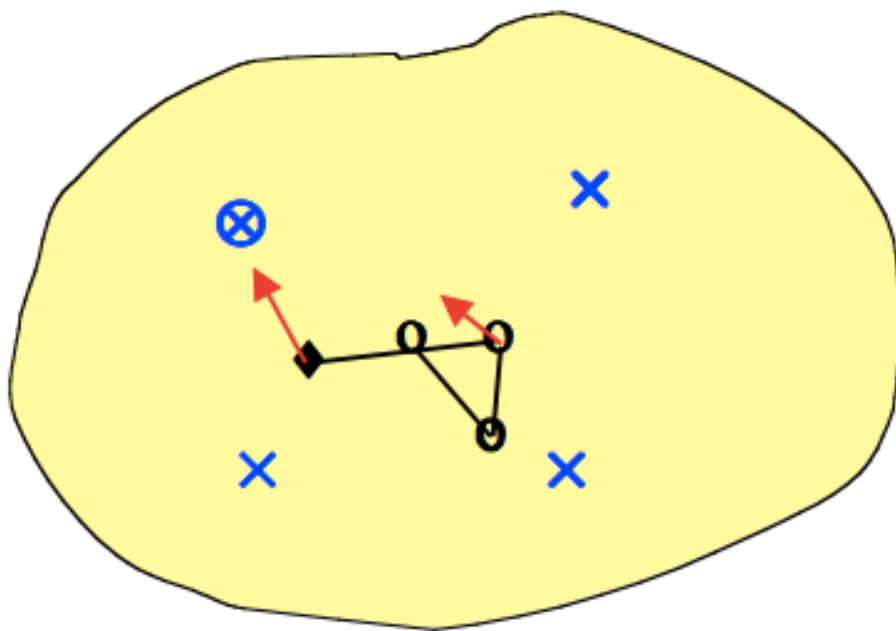
- A taxa de aprendizado varia de acordo com a fórmula

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{t_2}\right)$$

Como funciona?



Como funciona?



Treinamento da Rede

- Normalmente acontece em duas fases:

1. Fase de ordenação

- Ordenação topológica da rede – leva em torno de 1000 iterações
- $\eta(n)$ deve ser um valor próximo de 0.1 e permanecer acima de 0.01 e t_2 igual a 1000
- Inicialmente, $h(n)$ deveria incluir a maioria dos neurônios. Assim, σ_0 deve ser igual ao raio da rede e $t_1 = 1000/\log \sigma_0$

Treinamento da Rede

2. Fase de convergência

- Diferenciação entre neurônios vizinhos
- O número de interações deve ser pelo menos 500 vezes o número de neurônios
- O parâmetro de aprendizagem η deve ser mantido em torno de 0.01
- Função de vizinhança deve incluir 0 ou 1 neurônios

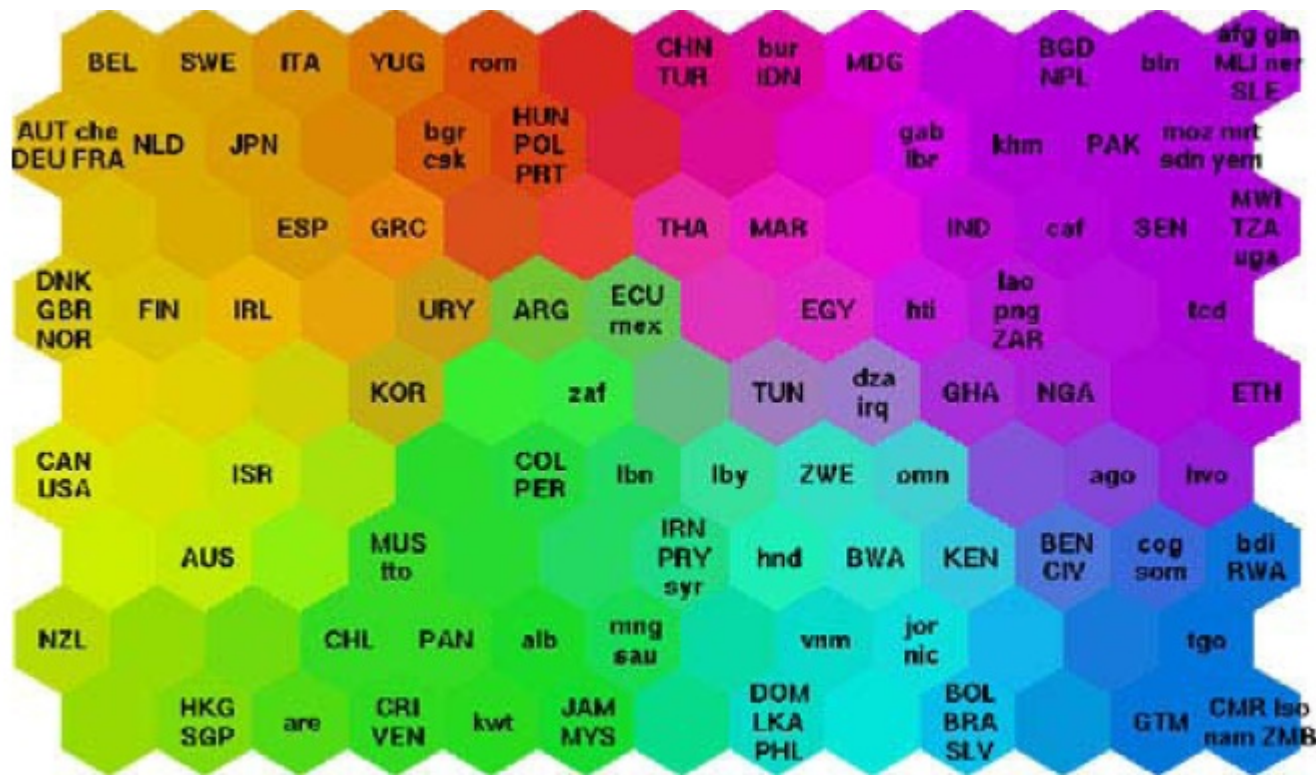
Aplicações

- Reconhecimento de voz
- Monitoramento de plantas e processos industriais
- Classificação de nuvens em imagens de satélites
- Organização de coleções de documentos
 - WEBSOM
- Análise e visualização de coleções de dados estatísticos
 - Ex: dados financeiros

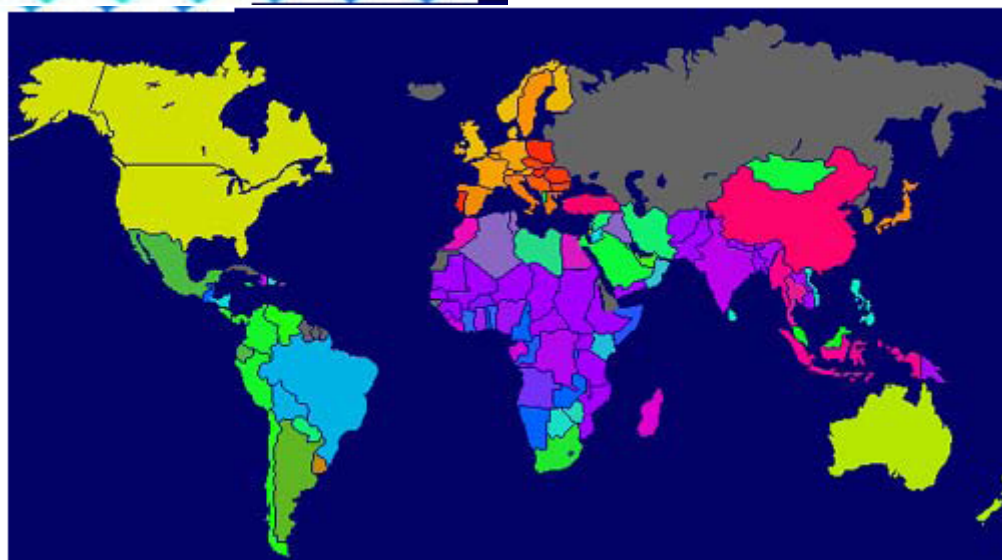
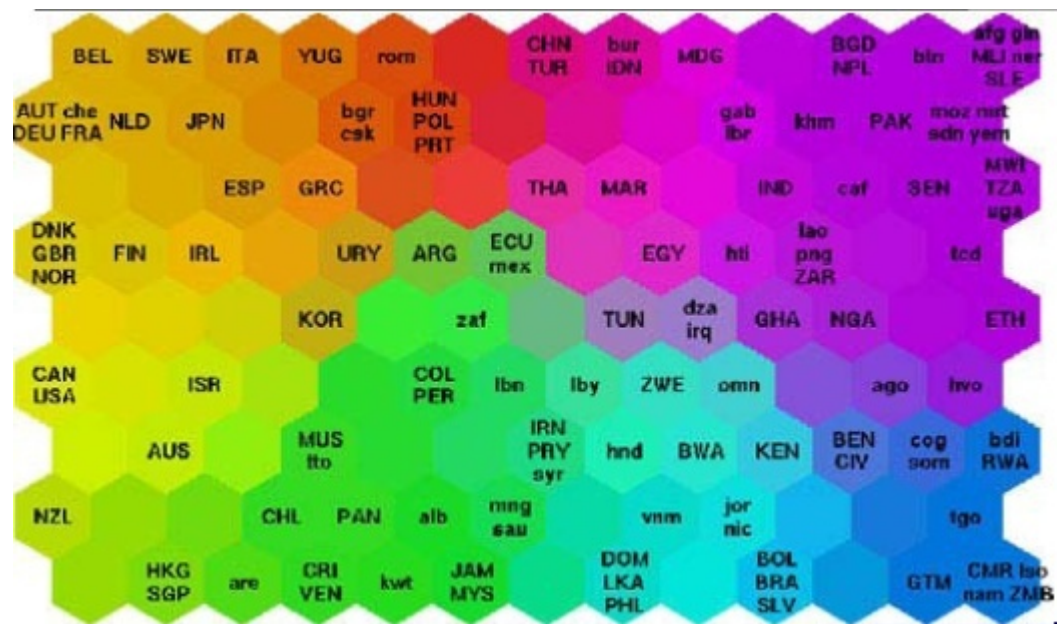
Mapa de Características

- O resultado da rede é um mapa de características em 1D ou 2D
- Os neurônios estão agrupados no espaço de dados, seguindo as posições dos grupos de dados
- Pode-se projetar dados nesse mapa com o objetivo de visualizá-los melhor
- Observando a quais neurônios os dados estão mapeados, podemos encontrar relações nos dados

Agrupamento de Países de Acordo com seu grau de Pobreza



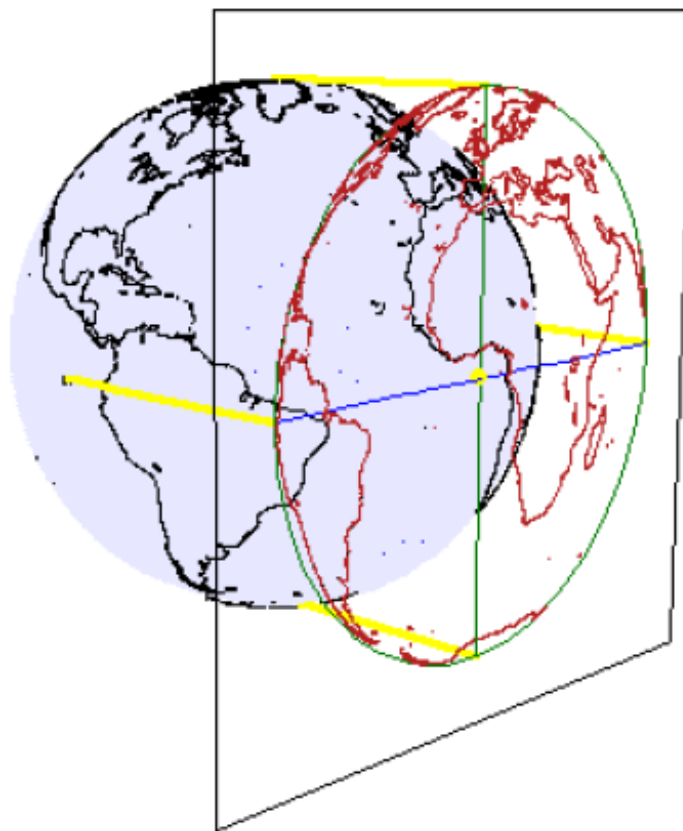
(Poverty map based on 39 indicators from World Bank Statistics, 1992)



Visualização

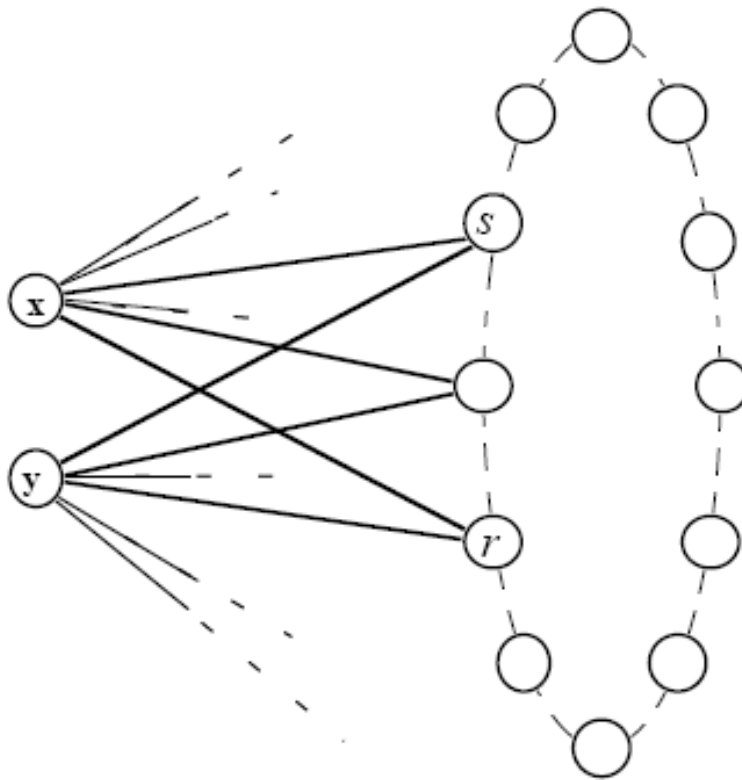
- É difícil visualizar dados contínuos representados em n -dimensões
- Redução da dimensionalidade facilita a visualização quando a dimensão é reduzida de nD para $2D$

Redução de Dimensionalidade



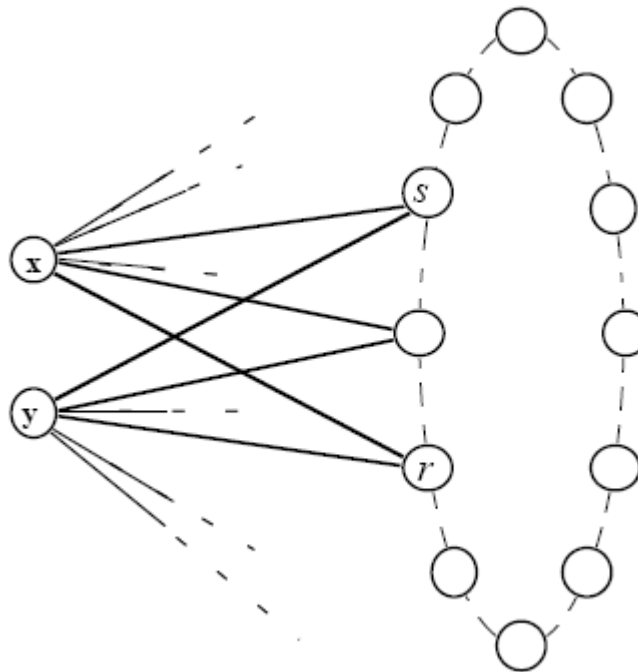
Caxeiro Viajante e SOM

- Entrada da rede ???
- Neurônios ??



Caxeiro Viajante e SOM

- Entrada da rede
 - Coordenadas (x,y) de uma cidade
- Cada neurônio possui 2 pesos, $(w_x, w_y) = \vec{w}$



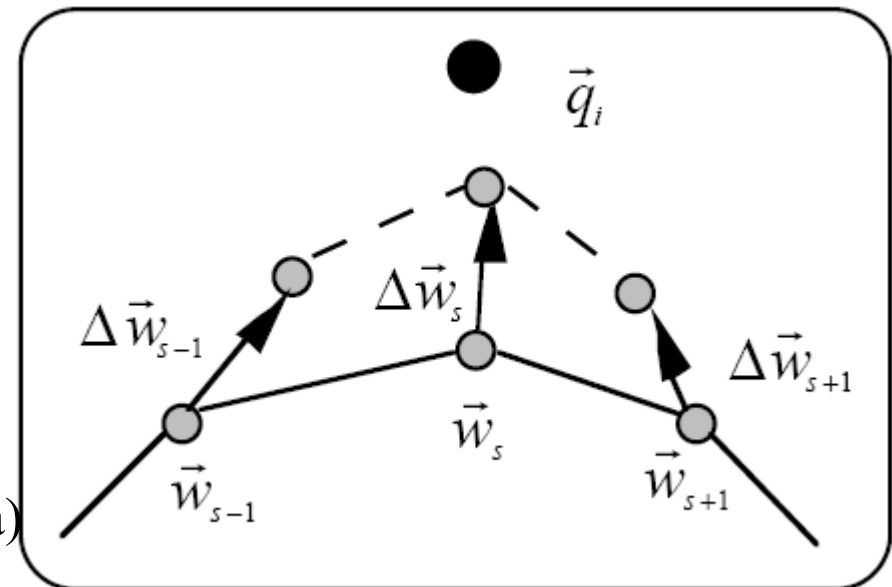
Caxeiro e SOM

- Cada neurônio e cada entrada podem ser representados por um ponto no espaço 2D
- Cada vez que uma entrada (cidade) é apresentada para a rede, o neurônio mais próximo da entrada é selecionado.
- O peso dos outros neurônios da rede são atualizados de acordo com:

$$\Delta \vec{w}_r = \epsilon h_{rs} (\vec{q}_i - \vec{w}_r)$$

taxa de aprendizagem

distância (Gaussiana)



Demos

- <http://www.patol.com/java/TSP/index.html>
- <http://www.eee.metu.edu.tr/~alatan/Courses/Demo/Kohonen.htm>
- <http://www.cis.hut.fi/research/javasomdemo/demo2.html>