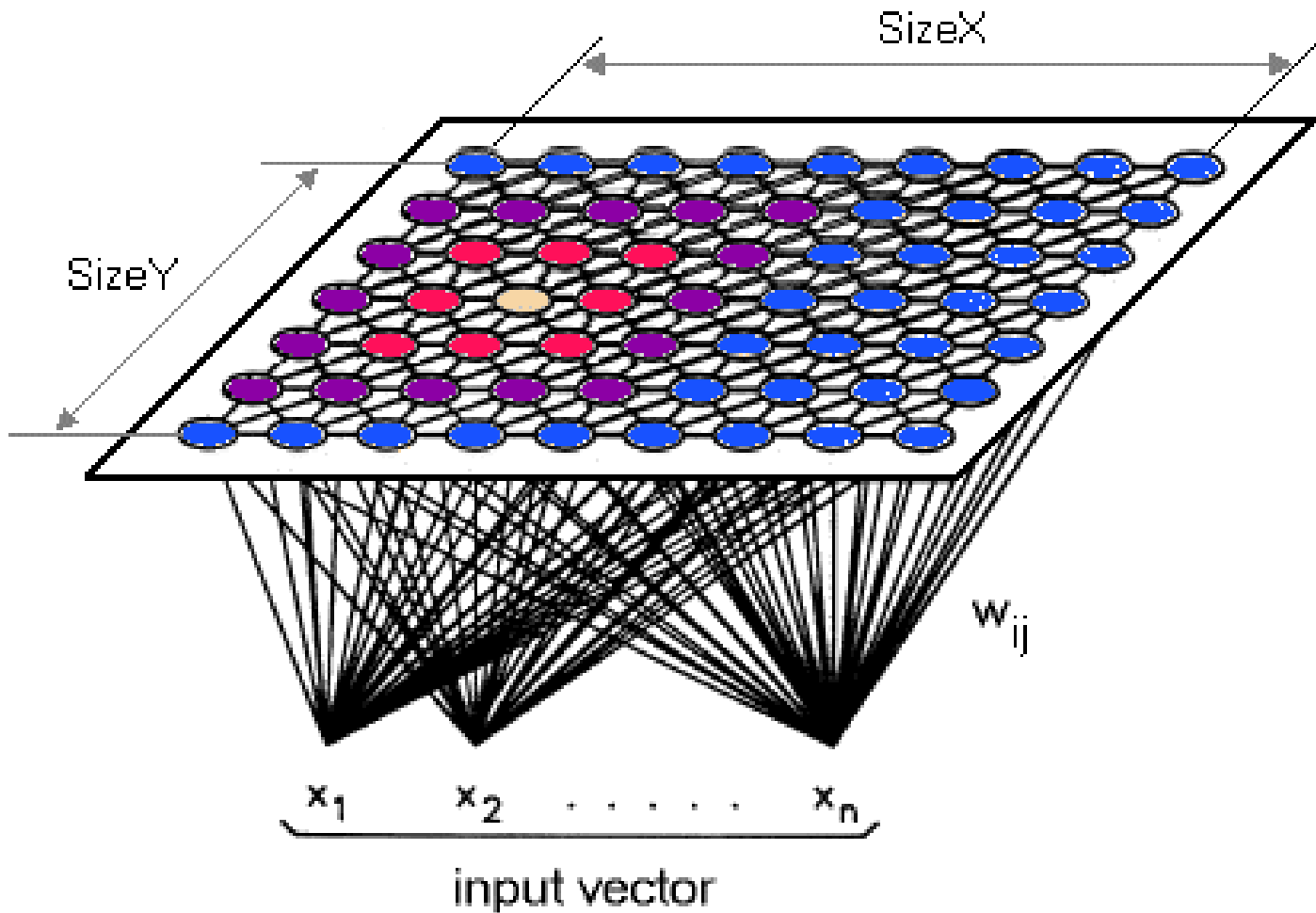


Self-Organizing Maps

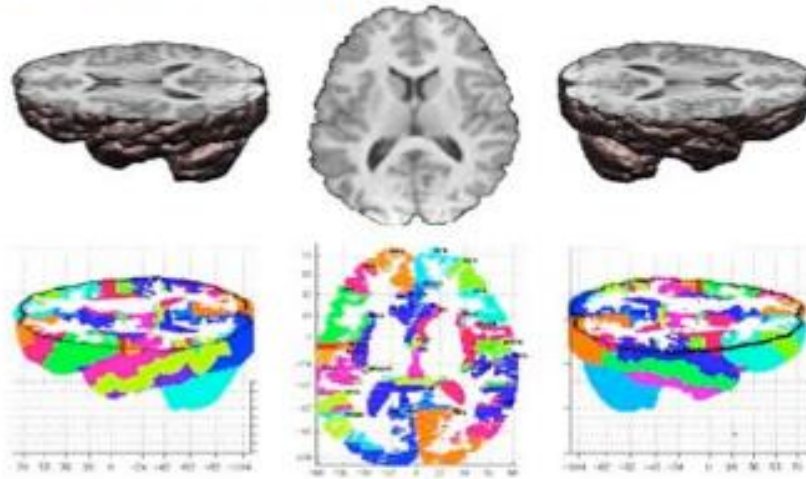


Tópicos

- *Introdução*
- *Algoritmo*
- *Exemplos de aplicação*
- *Conclusões*

Self-Organizing Maps

Self-Organizing Maps (SOM) were derived as a mathematical model of these configurations (T. Kohonen, 1981)



Inspired by patterns in cerebral cortex: the detailed topographical order of the neural connections (synapses) form localized maps.

Brain maps are determined both **genetically** and by **experience**

"experience" = some projections – growth of axons of neural cells – are developed or stunted with respect to others, different cells are recruited for different tasks

SOMs

- As redes SOMs foram desenvolvidos por T. Kohonen nos anos 80 (T. Kohonen, *Self-Organizing Maps*, Springer, 1995, 1997, 2006).
- SOMs são um tipo de rede neural cujos nodos / neurônios → mapear células → são sintonizados por um conjunto de sinais de entrada / dados / amostras de acordo com uma certa forma de adaptação (semelhante à uma regressão).
- Os vários nós formam um mapa topologicamente ordenado durante o processo de aprendizagem.
- O processo de aprendizagem não é supervisionado → não é necessário um vetor de referência de "resposta correta".
- Os nós são decodificadores dos sinais de entrada → podem ser usados para reconhecimento de padrões.
- Mapas bidimensionais são usados para agrupar / visualizar dados de elevada dimensão.

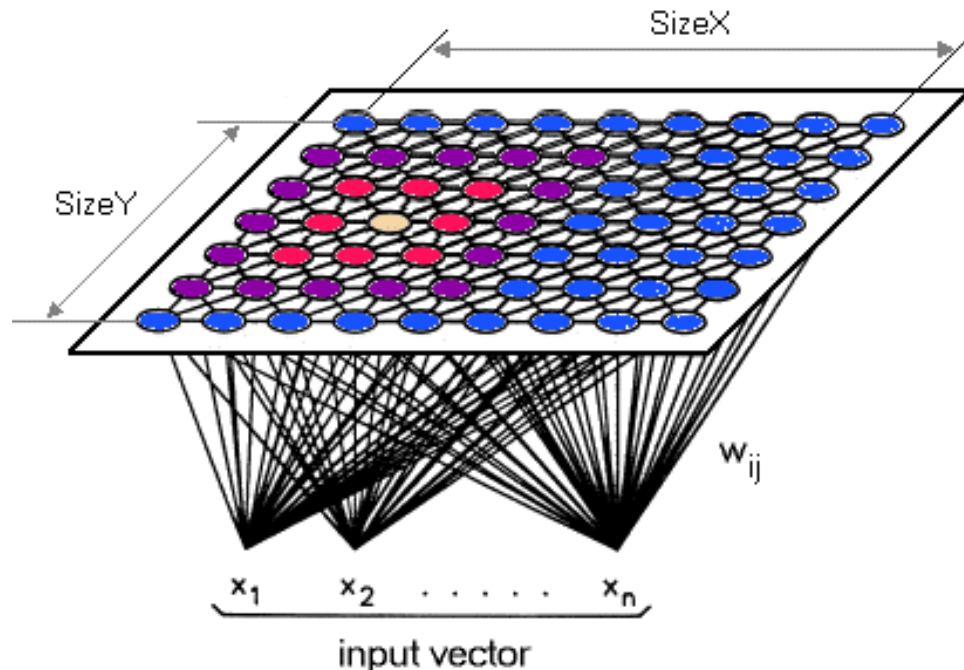
Self-Organizing Maps (Kohonen Maps)

- No córtex humano, espaços de entrada sensoriais multidimensionais (por exemplo, a entrada visual e a entrada sensorial tátil) são representados por mapas bidimensionais.
- A projeção das entradas sensoriais em tais mapas é feita de modo a haver conservação da topologia.
- ...Tal significa que as áreas vizinhas nesses mapas representam áreas vizinhas no espaço de entrada sensorial.
- Por exemplo, áreas do córtex sensorial responsáveis pelas regiões do braço e da mão são vizinhas.

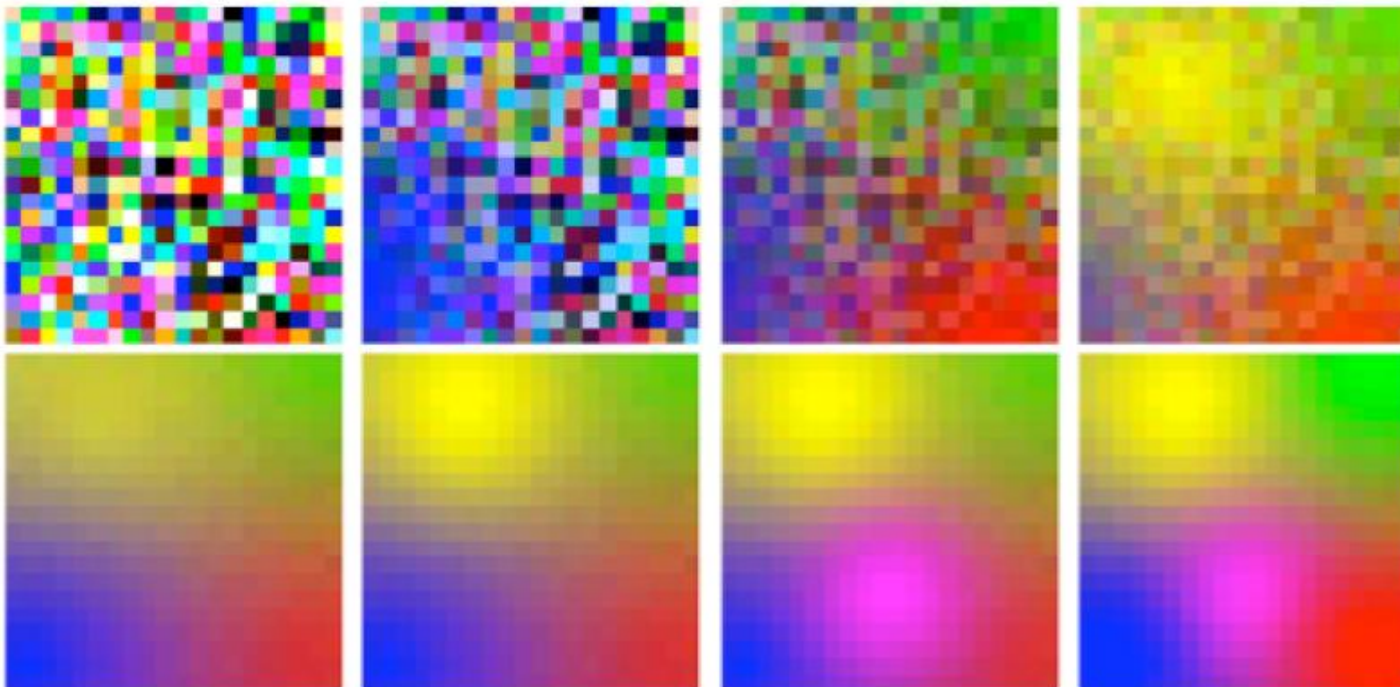
Self-Organizing Maps (Kohonen Maps)

SOMs é constituído por:

- Duas camadas: camada de entrada e camada de saída (mapa)
- Camadas de entrada e saída estão completamente ligadas.
- Os neurônios de saída estão ligados entre si dentro de uma mesma zona ou “bairro”.
- A topologia (relação de vizinhança) é definida na camada de saída.



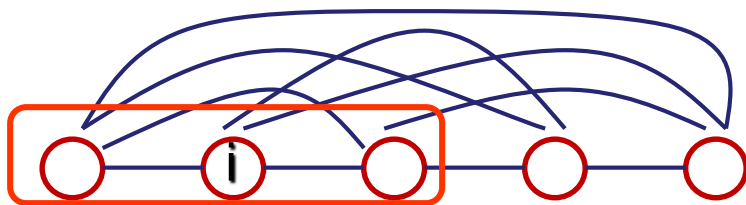
“Colors” Example



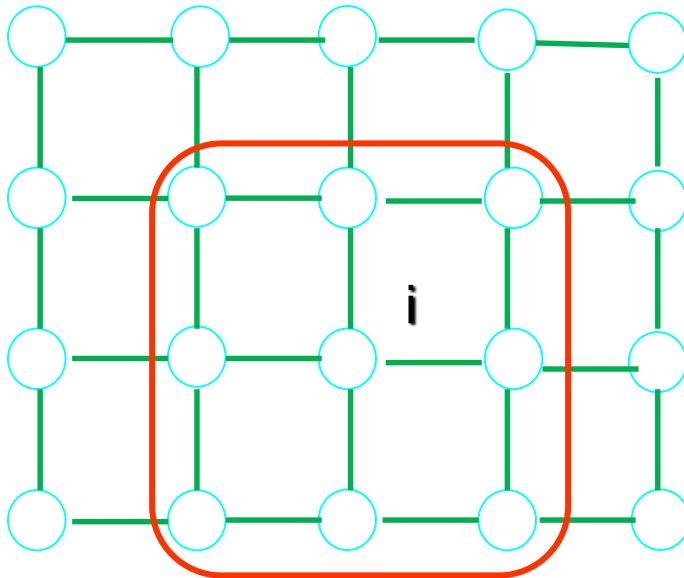
Representação do mapa de 5 amostras iniciais:
azul, amarelo, vermelho, verde, magenta

Self-Organizing Maps (Kohonen Maps)

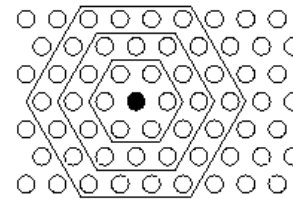
Exemplos de estrutura na camada de saída (mapa) :



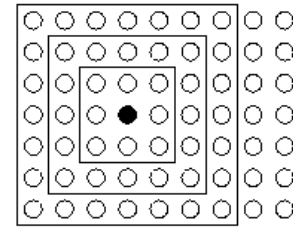
Topologia Unidimensional
(nº de ligações completas)



Bi-dimensional



(a) Hexagonal grid



(b) Rectangular grid



Vizinhança do neurónio i

Aprendizagem:

As células, i , que estão próximas do "neurônio vencedor" k , ativam-se para "aprender" o valor de x

$$W_i(n+1) = W_i(n) + \varphi_{ki}(n) [x(n) - W_i(n)]$$

Iteração do treino

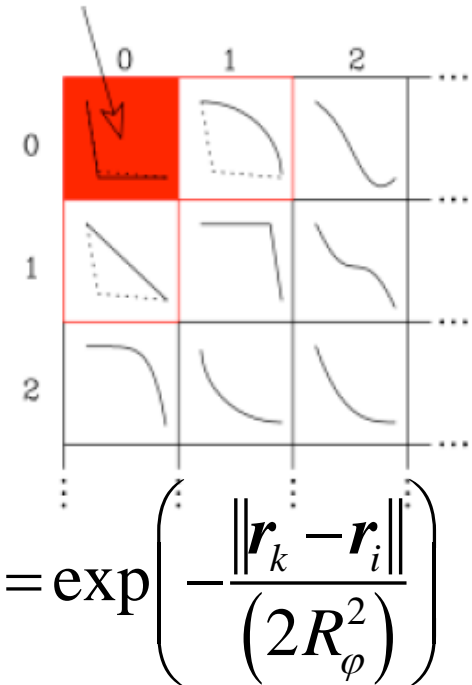
$$\varphi_{ki}(n) = \underbrace{\left(\frac{n_{train} - n}{n_{train}} \right)}_{R_\varphi(n)} \left(\frac{1}{1 + \|\mathbf{r}_k - \mathbf{r}_i\|} \right) \quad \text{ou} \quad \varphi_{ki}(n) = \exp \left(-\frac{\|\mathbf{r}_k - \mathbf{r}_i\|}{(2R_\varphi^2)} \right)$$

Função vizinhança \Rightarrow decresce com "n" e com a "distancia"

(Aqui estão apresentadas duas versões possíveis para φ)

$$R_\varphi(n) = 1.5 \left(\frac{n_{train} - n}{n_{train}} \right) R_m \quad \leftarrow \text{Tamanho do mapa}$$

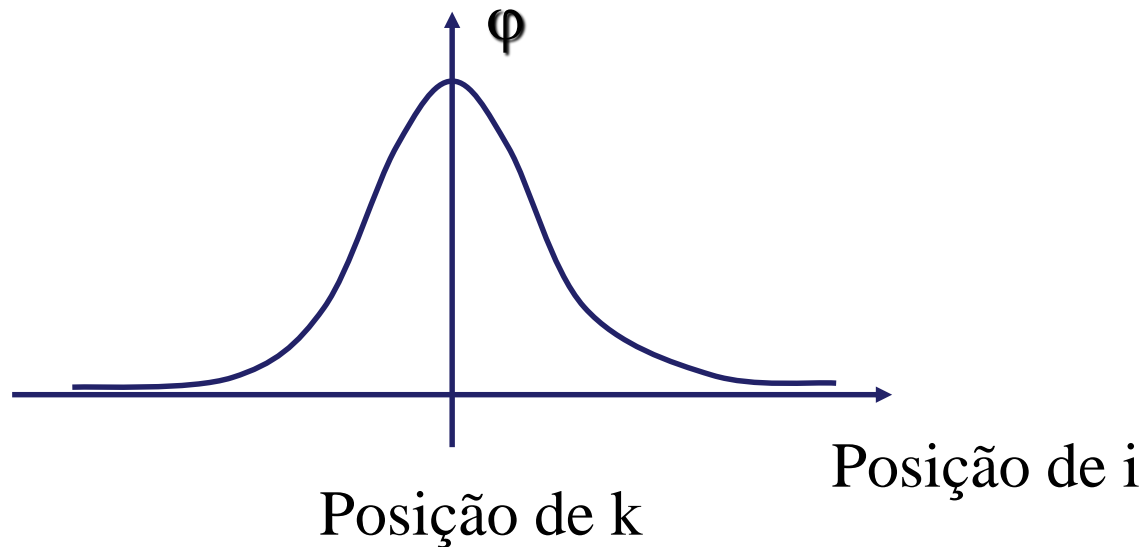
Tamanho da vizinhança \Rightarrow decresce com "n"



Nota:

Uma função de vizinhança $\varphi(i, k)$ mede a proximidade entre os neurônios i e k da camada de saída que estão ligados entre si.

A função gaussiana é frequentemente usada para medir a distância entre os dois neurônios da camada:



Aprendizagem não supervisionada

Seja o espaço de entrada n -dimensional e m o número de neuróns no mapa de saída

(1) Inicializar o neurónio i , $i = 1, \dots, m$, com um vector peso w_i aleatório.

(2) Escolher aleatoriamente uma entrada x

(3) Calcular o neurónio vencedor, k :

$$\|w_k - x\| = \min_i \|w_i - x\| \quad (\text{Distância Euclideana})$$

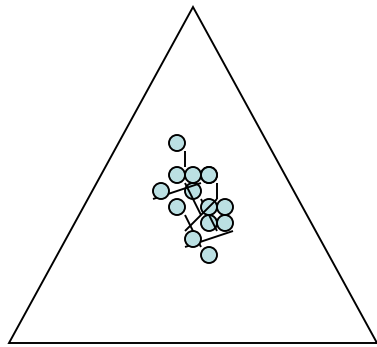
(4) Actualizar todos os vectores pesos i na vizinhança do neurónio k : $w_i := w_i + \eta \cdot \varphi(i, k) \cdot (x - w_i)$

(w_i aproxima-se de x)

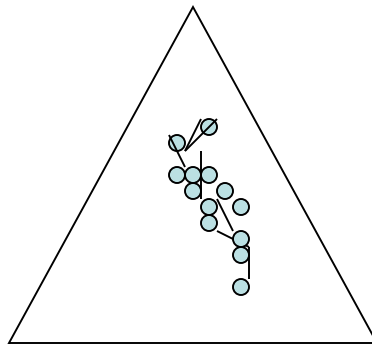
(5) Se o critério de paragem é satisfeito, então STOP.

Senão, estreitar a função de vizinhança φ e ajustar o parâmetro de aprendizagem η e ir de novo para o passo (2).

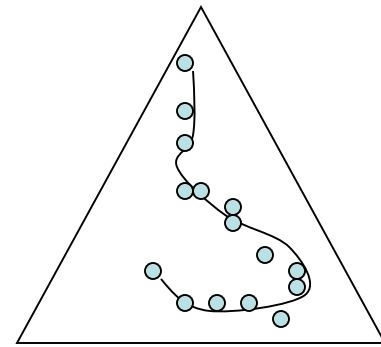
Exemplo I: Learning a one-dimensional representation of a two-dimensional (triangular) input space:



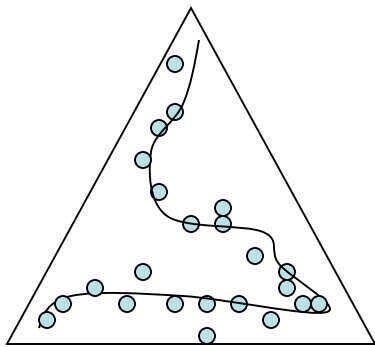
0



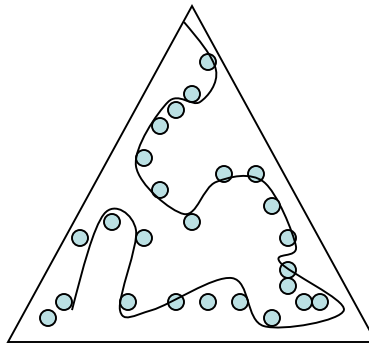
20



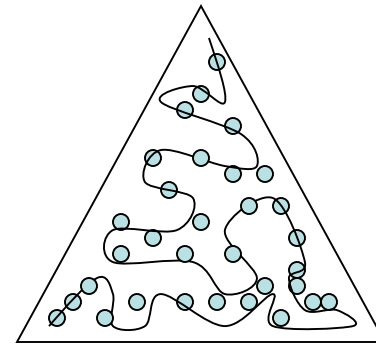
100



1000

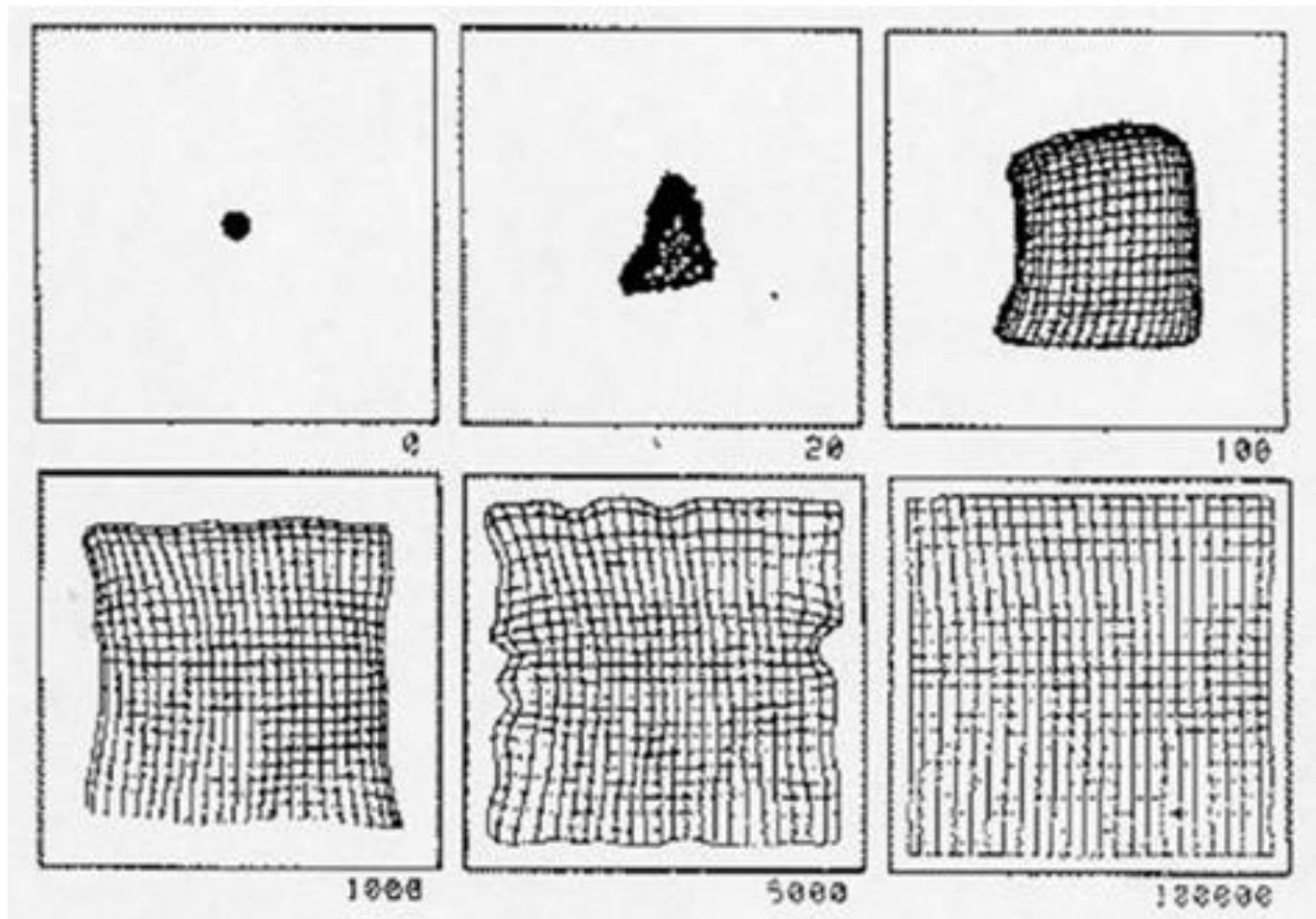


10000



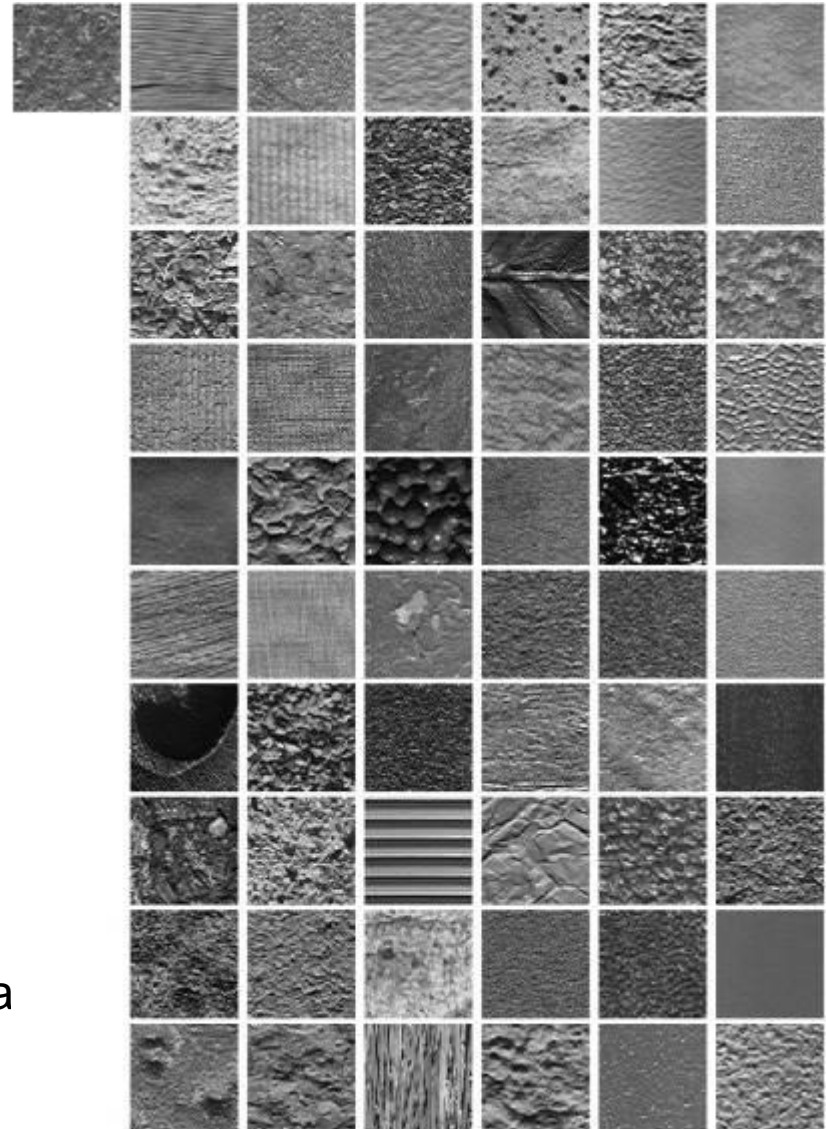
25000

Exemplo II: Learning a two-dimensional representation of a two-dimensional (square) input space:



Exemplo III:

Learning a two-dimensional
mapping of texture images



<https://www.sciencedirect.com/science/article/pii/S0031320303002310>

Bibliografia

- [\(wikipedia\) https://en.wikipedia.org/wiki/Self-organizing_map](https://en.wikipedia.org/wiki/Self-organizing_map)
- <https://rev-inv-ope.univ-paris1.fr/fileadmin/rev-inv-ope/files/39118/39118-01.pdf>