31 Jul 2020

YRM71, YS591 – Mathematical Neuroscience
Project 4: Self-Organizing Maps

Assignment

1. Perform computer simulation of self-organizing maps to reproduce the Fig.3.5 $\sim$ Fig.3.7 of Kohonen (2001). $p(\boldsymbol{x})$ is uniform distribution from the square $-1 \leqq x_1, x_2 \leqq 1$ for Fig 3.5 (left) and Fig 3.6, and the triangle which has vertices at $(-1,-1), (1,-1), (0,1)$ for Fig 3.5 (right) and Fig.3.7.

2. Try various input distribution $p(\boldsymbol{x})$.

3. Try different number of nodes N, and parameter values α, σ.

4. Try your idea of anything.

Computer Experiment Procedure

1. Set reference vectors $\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_N$ with each components $m_{i1}, m_{i2}, i = 1, \cdots, N$ random values ([-0.1,0.1] uniform distribution, for example), where $N = 16 \times 16 = 256$ in Fig.3.5, $N = 24 \times 24 = 576$ in Fig.3.6, and $N = 64$ units in Fig.3.7.

2. Generate an input signal $\boldsymbol{x} \sim p(\boldsymbol{x})$ randomly, and find the winner unit c

$$c = \underset{i}{\operatorname{argmin}} ||\boldsymbol{x} - \boldsymbol{m}_i||$$

3. Learning: Update the reference vectors $\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_N$

$$\Delta \boldsymbol{m}_i = \alpha(\boldsymbol{x} - \boldsymbol{m}_i) \exp\left\{-\frac{||\boldsymbol{r}_c - \boldsymbol{r}_i||^2}{2\sigma^2}\right\}$$

   where $\boldsymbol{r}_i$ is the coordinate of ith unit in the neural field or array of units, and $\alpha > 0$ is small learning constant.

4. Repeat from 2. until the update doesn't change $\{\boldsymbol{m}_i\}$ so much.

5. How to draw the graph: Plot reference vectors of all units. Connect two reference vectors (two points in the graph) with a line when the two units are neighbors in the neural field.

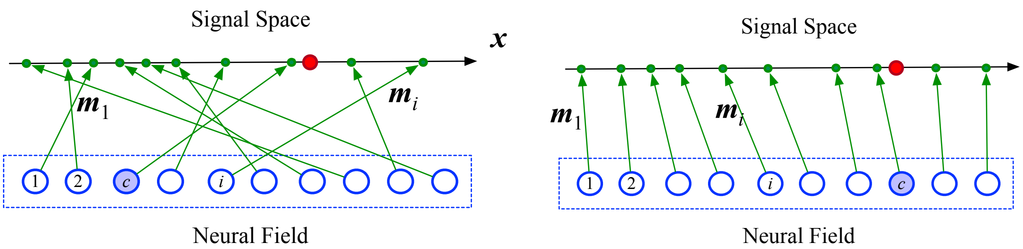
Mathematical Neuroscience
31 July 2020
A. Date

Self-Organizing Maps

Key words: unsupervised learning, reference vectors, topology-preserving mapping, neighbors

1. **Set up**

   - input signal: $\boldsymbol{x} = (x_1, x_2, \cdots x_n)$. $x_k \in \mathcal{R}, k = 1, \cdots, n$. $\boldsymbol{x} \sim p(\boldsymbol{x})$.
   - node (neuron or unit): i, $i = 1, \cdots, N$
   - array of nodes (neural field): e.g., 1-dimensional or 2 dimensional array.
   - reference vector ($\approx$ connection weights): $\boldsymbol{m}_i = (m_{i1}, m_{i2}, \cdots, m_{in})$
   - forming a topological map: $f_M : \boldsymbol{x} \to r_c$ (location of node). $M = \{\boldsymbol{m}_i\}$.



Maps: Before learning (left) and after learning (right). c is winner node for input $\boldsymbol{x}$ (red).

2. **Activity dynamics:** (Winner takes all.)

   For a given input $\boldsymbol{x}$, only one node (we call winner) c is active or on.

$$c = \underset{i}{\operatorname{argmin}} ||\boldsymbol{x} - \boldsymbol{m}_i||$$

3. **Learning dynamics:** (Neighborhood learning)

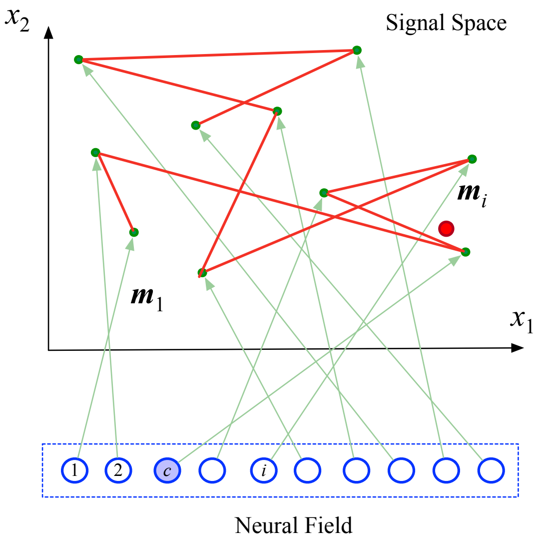
   Update the reference vectors $\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_N$

$$\Delta \boldsymbol{m}_i = \alpha(\boldsymbol{x} - \boldsymbol{m}_i) \exp\left\{-\frac{||\boldsymbol{r}_c - \boldsymbol{r}_i||^2}{2\sigma^2}\right\}$$

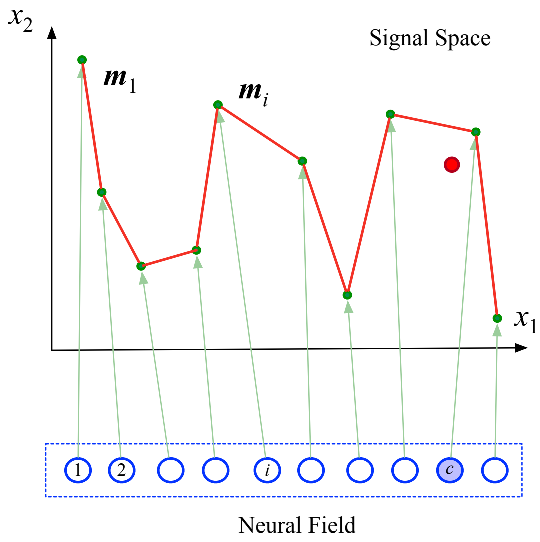
   where $\alpha > 0$ is small constant (learning coefficient).

4. **Topological map**

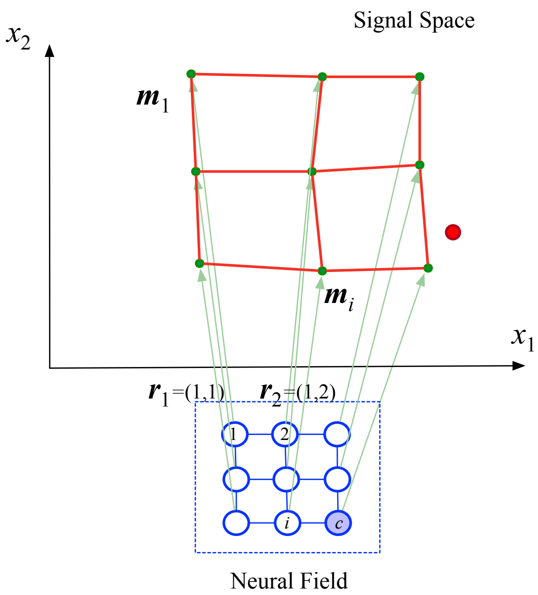
   - An input $\boldsymbol{x}$ define a (winner) node $c : \boldsymbol{x} \to c$.
   - One area in the signal space (X_i) corresponds to one node (i), $i = 1, \cdots, N$.
   - The areas of adjacent nodes are almost always adjacent in the signal space.



x_2
Signal Space
$\boldsymbol{m}_i$
$\boldsymbol{m}_1$
x_1
1
2
c
i
Neural Field



x_2
Signal Space
$\boldsymbol{m}_1$
$\boldsymbol{m}_i$
x_1
1
2
i
c
Neural Field



Signal Space
x_2
$\boldsymbol{m}_1$
$\boldsymbol{m}_i$
x_1
$\boldsymbol{r}_1$=(1,1)
$\boldsymbol{r}_2$=(1,2)
1
2
i
c
Neural Field

Examples of Ordering. It may be quite surprising that when starting with random $m_i(0)$, the reference vectors will attain *ordered* values in the long run, even in high-dimensional spaces. This ordering is first illustrated by means of two-dimensional input data $x = [\xi_1, \xi_2]^{\mathrm{T}} \in \Re^2$ that have some arbitrarily structured distribution. For simplicity, if x is a stochastic vector, its probability density function $p(x)$ is in this example assumed uniform within the framed areas in Fig. 3.5 and zero outside them. The topological relations between the neurons in a square array can be visualized by auxiliary lines that are drawn between the neighboring reference or codebook vectors (points in the signal space). The reference vectors in these graphs now correspond to the crossings and end points of this network of auxiliary lines, whereby the relative topological order becomes immediately visible.

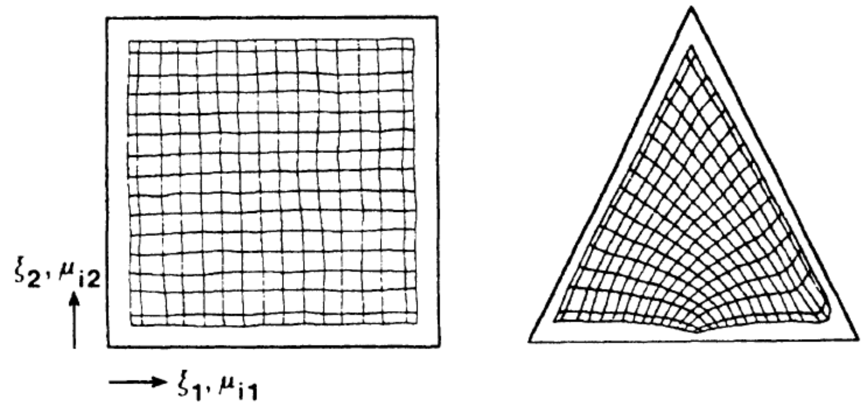


Fig. 3.5. Two-dimensional distributions of input vectors (framed areas), and the networks of reference vectors approximating them

Kohonen, T. *Self-Organizing Maps* (Springer-Verlag, Berlin, 2001).

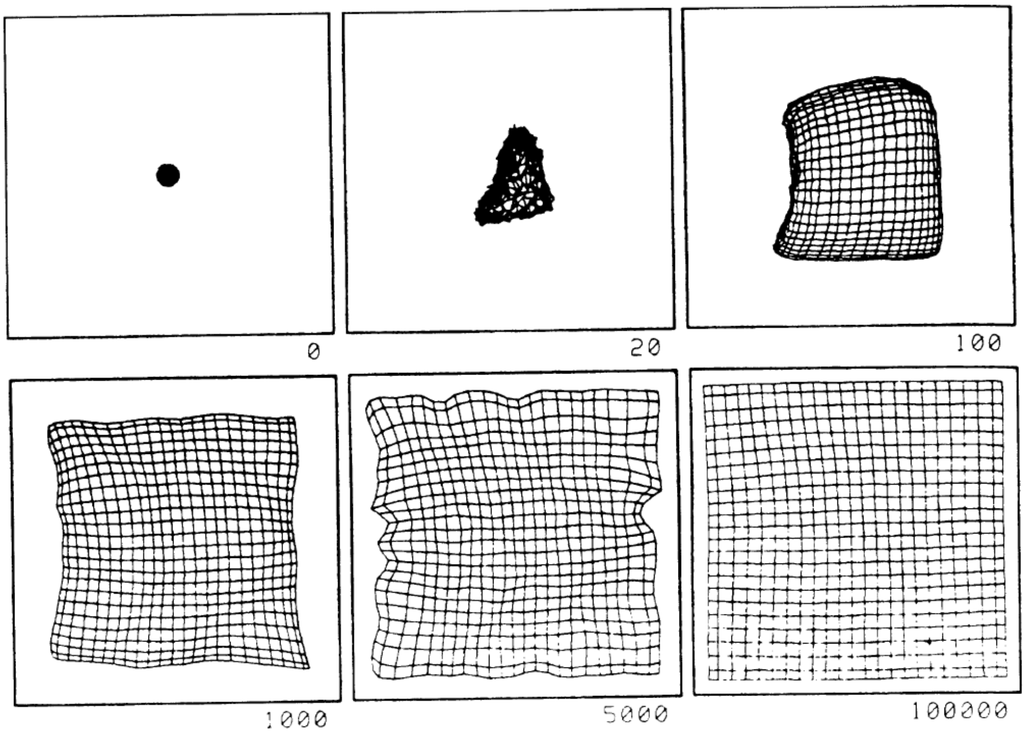


Fig. 3.6. Reference vectors during the ordering process, square array. The numbers at lower right-hand corner indicate learning cycles

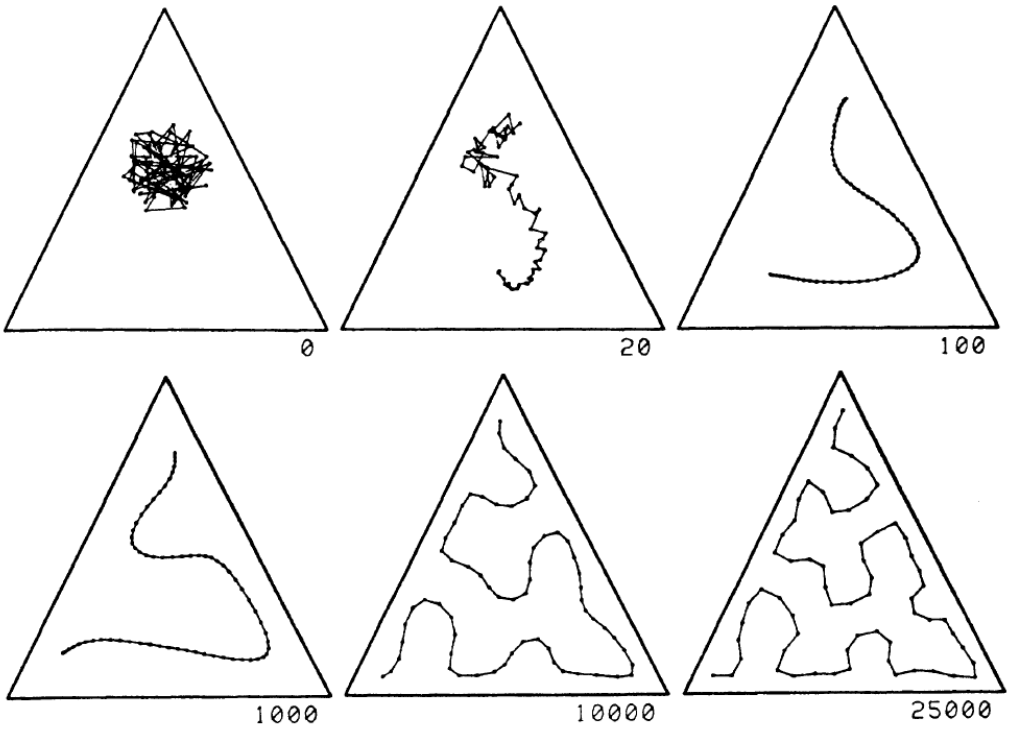


Fig. 3.7. Reference vectors during the ordering process, linear array

Kohonen, T. *Self-Organizing Maps* (Springer-Verlag, Berlin, 2001).

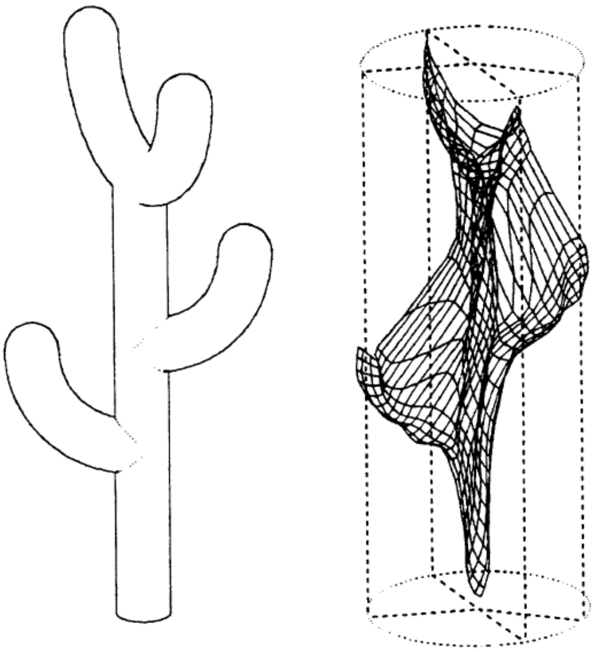


Fig. 3.9. SOM for a structured distribution of $p(x)$. For clarity, the three-dimensional $p(x)$, having uniform density value inside the "cactus" shape and zero outside it, is shown on the left, whereas the "net" of reference vectors is displayed on the right in a similar coordinate system

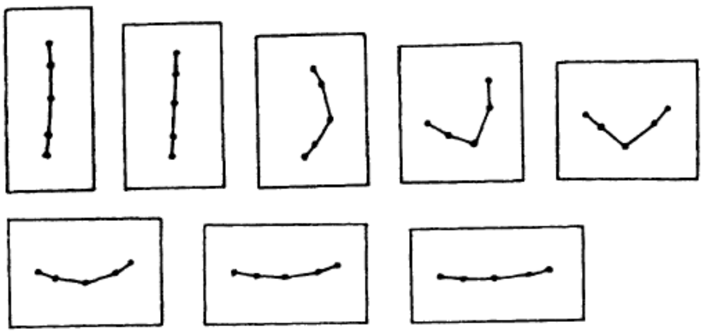


Fig. 3.10. Automatic selection of dimensions for mapping

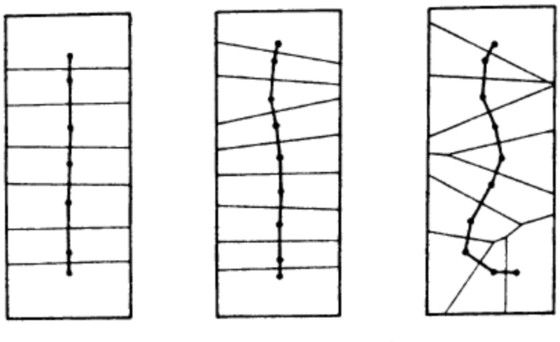


Fig. 3.11. Distribution of reference vectors with different length of a linear array

Kohonen, T. *Self-Organizing Maps* (Springer-Verlag, Berlin, 2001).