Kohonen's self-organizing feature map

Both MLPs and RBFs require a "teacher": the error between the network output and actual output is used in model training. This is called *supervised learning*.

Note that the k-means algorithm does not require a "teacher", i. e. no output information is used. The algorithm leads to the placement of a set of vectors in input space. This is called *unsupervised learning*.

Another *unsupervised learning* is the Kohonen's self-organizing feature map.

The *unsupervised learning* is often used to discover significant features of the input data.

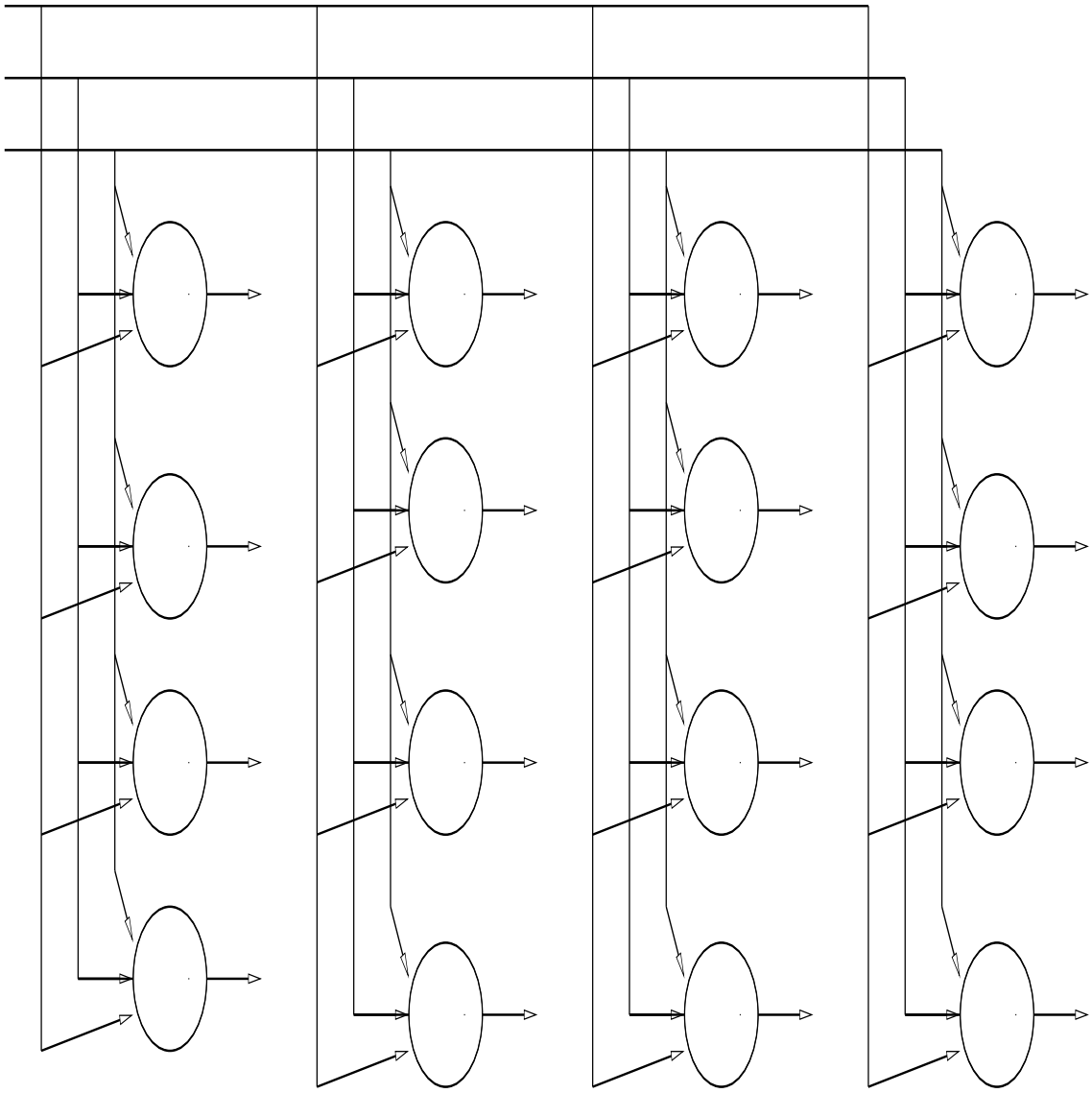

Figure: Two dimensional lattice of neurons. This is commonly used in SOM architecture.

The algorithm for the formation of self-organizing map (SOM) proceeds first by initializing the synaptic weights in the network as small random numbers. There are three essential processes as summarized below:

1. Competition. For each input pattern, the neurons in the network computes their respective values of a discriminant function. A particular neuron is declared as the winner.

2. Cooperation. The winning neuron determines the spatial location of a topological neighborhood of excited neurons, thereby providing the basis for cooperation among such neighboring neurons.

3. Synaptic adaptation. Suitable adjustments are applied to the synaptic weights of the neighborhood of excited neurons. This mechanism enables the response of the excited neurons to a similar input pattern is enhanced.

Competition Process

Let m denote the dimension of the input data, and let an input pattern be denoted as

$$\mathbf{x} = [x_1, ...x_m]^T$$

The synaptic weight vector of each neuron in the network has the same dimension of the input vector. Suppose that there are a total of l number of neurons in the network. The synaptic weight vector of neuron j is denoted by

$$\mathbf{w}_j = [w_{j,1}, ...w_{j,m}]^T, \quad j = 1, 2, ..., l$$

We use the index i to identify the neuron that best matches the input $\mathbf{x}$ amongst l neurons

$$\min\{d(\mathbf{x}, \mathbf{w}_j)\} = d(\mathbf{x}, \mathbf{w}_i), \quad j = 1, ..., l$$

Neuron i is the winner.

Cooperative Process

The winning neuron locates the center of a topological neighborhood of cooperating neurons.

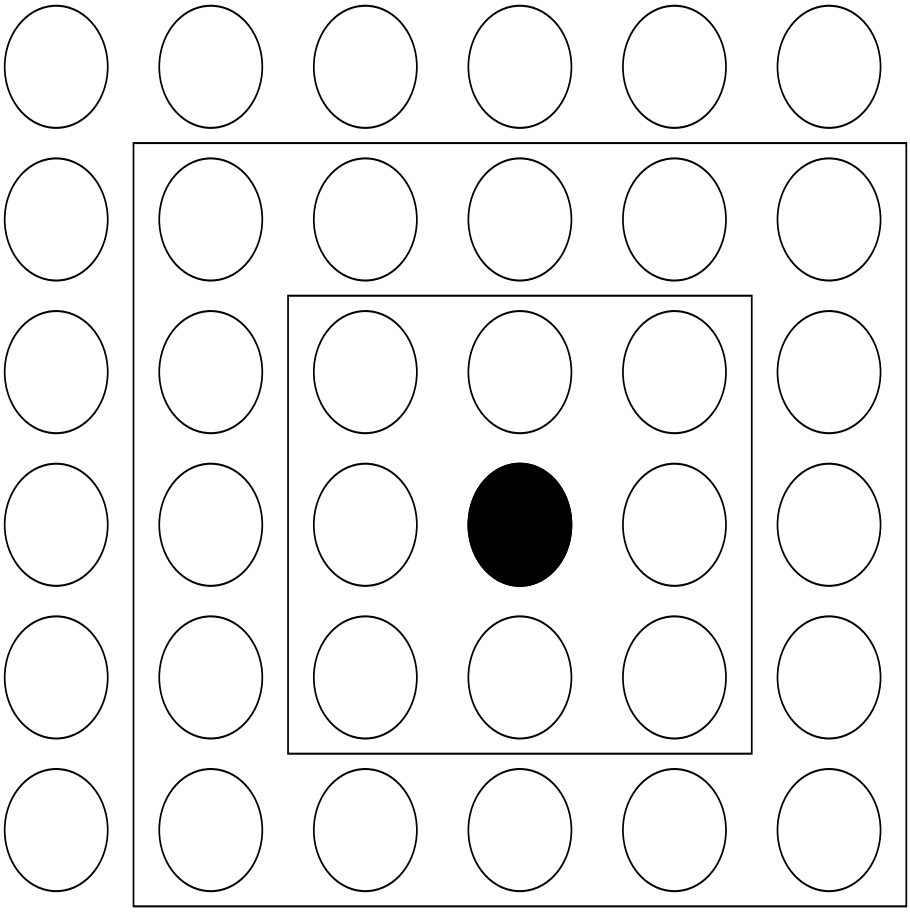

Figure: The distance between the neurons on the map lattice governs the degree of the interaction.

Synaptic adaptation

For all nodes j within neighbourhood of neuron i

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \eta h_{i,j}(t)(\mathbf{x}(t) - \mathbf{w}_j(t))$$

where t is the epoch number (discrete-time index). The input vector $\mathbf{x}(t)$ is obtained by selecting randomly a sample at iteration t. η is the learning rate. $h_{i,j}(t)$ is the neighborhood function that decreases to zero as the distance between neurons i and j increases. e.g. the Gaussian function

$$h_{i,j}(t) = \text{exp}(-\frac{d_{i,j}^2}{2\sigma^2})$$

$d_{i,j}^2$ is discrete and calculated based on the relative positions of neighboring neurons j and the winning neuron i.

Another feature is that the neighborhood should decrease with time.

Examples: The behavior of the SOM algorithm is illustrated using computer simulation.

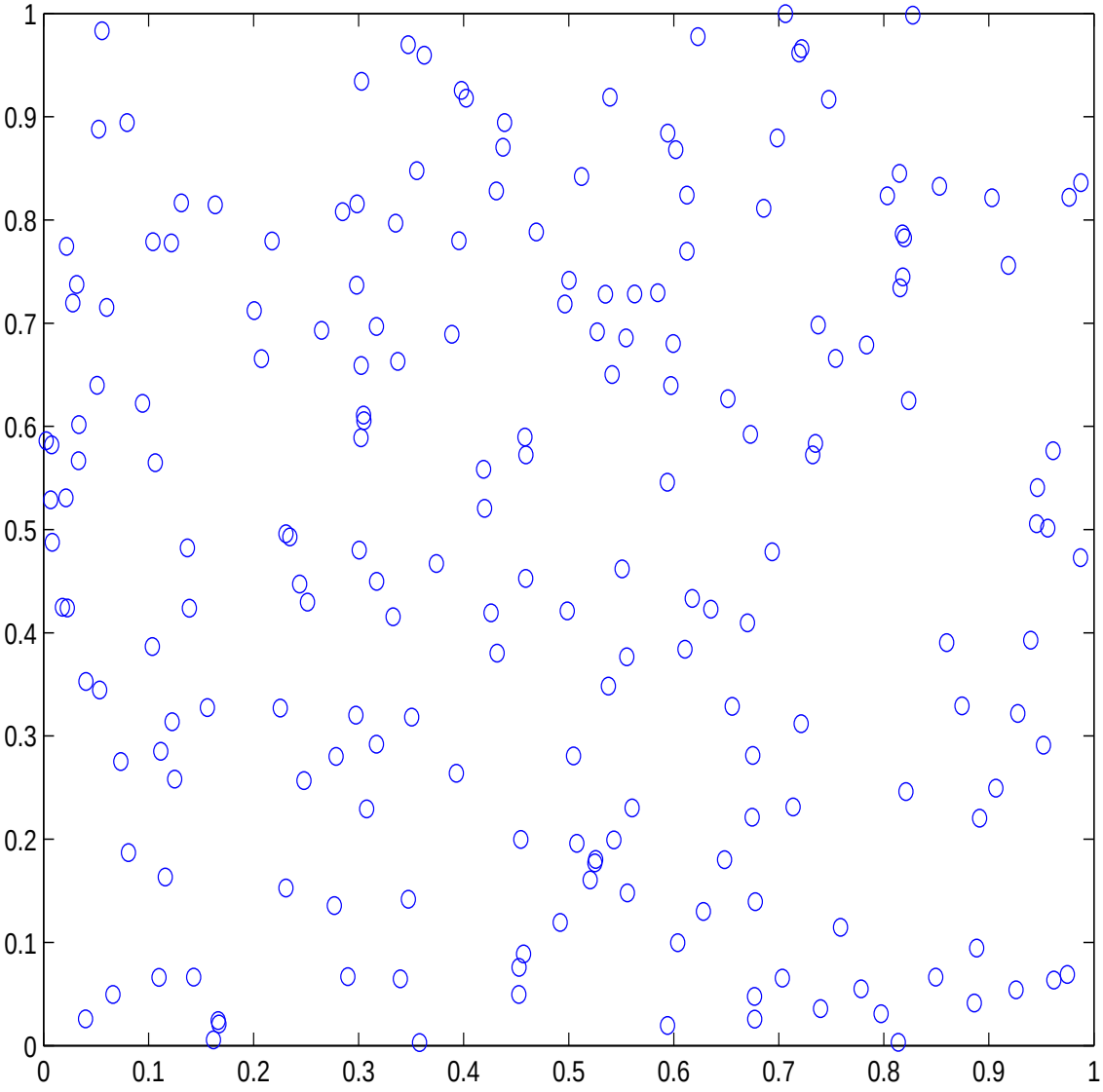

Figure: uniformly distributed input data set;

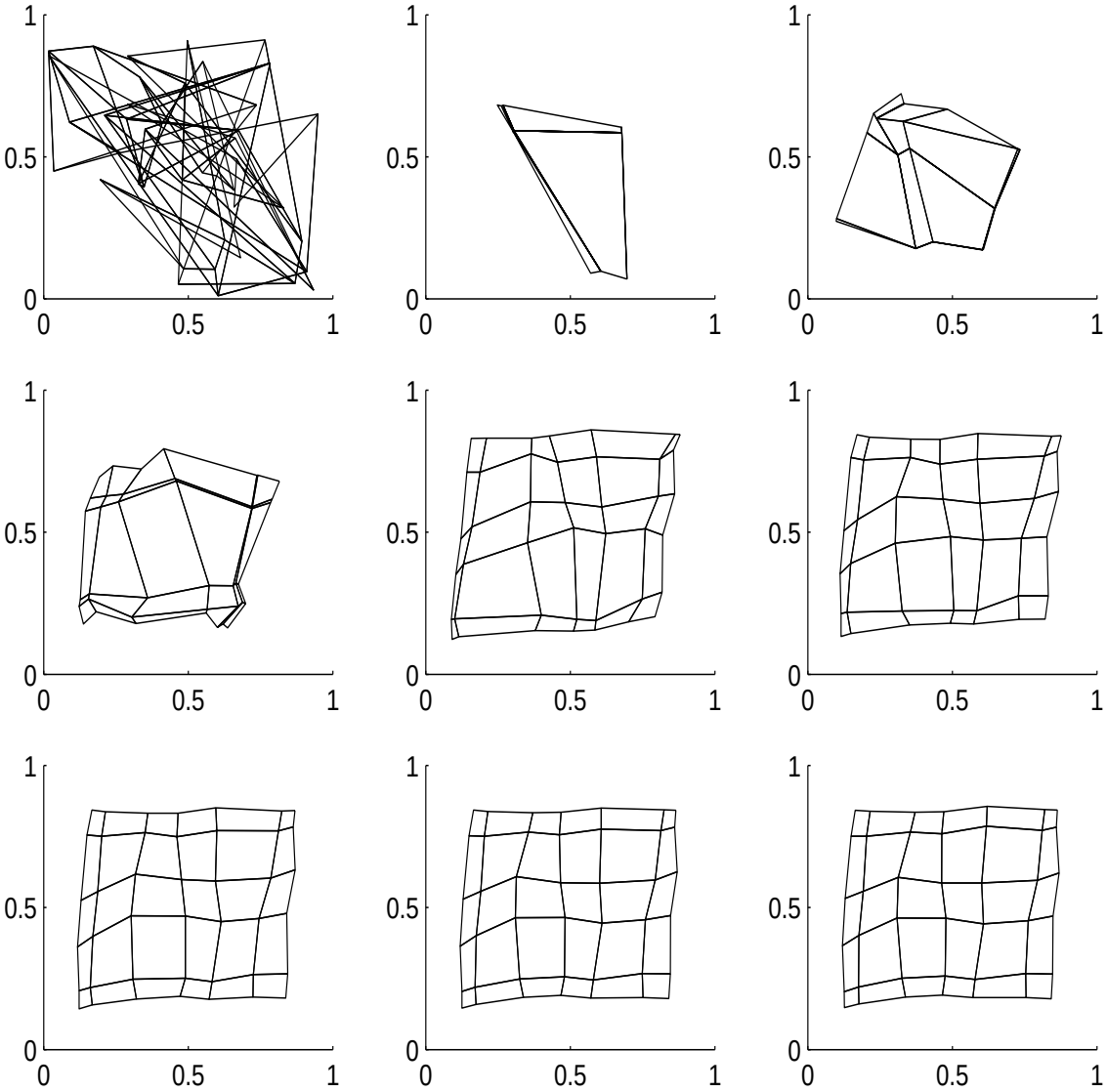

Figure: The evolution of SOM for the data set.

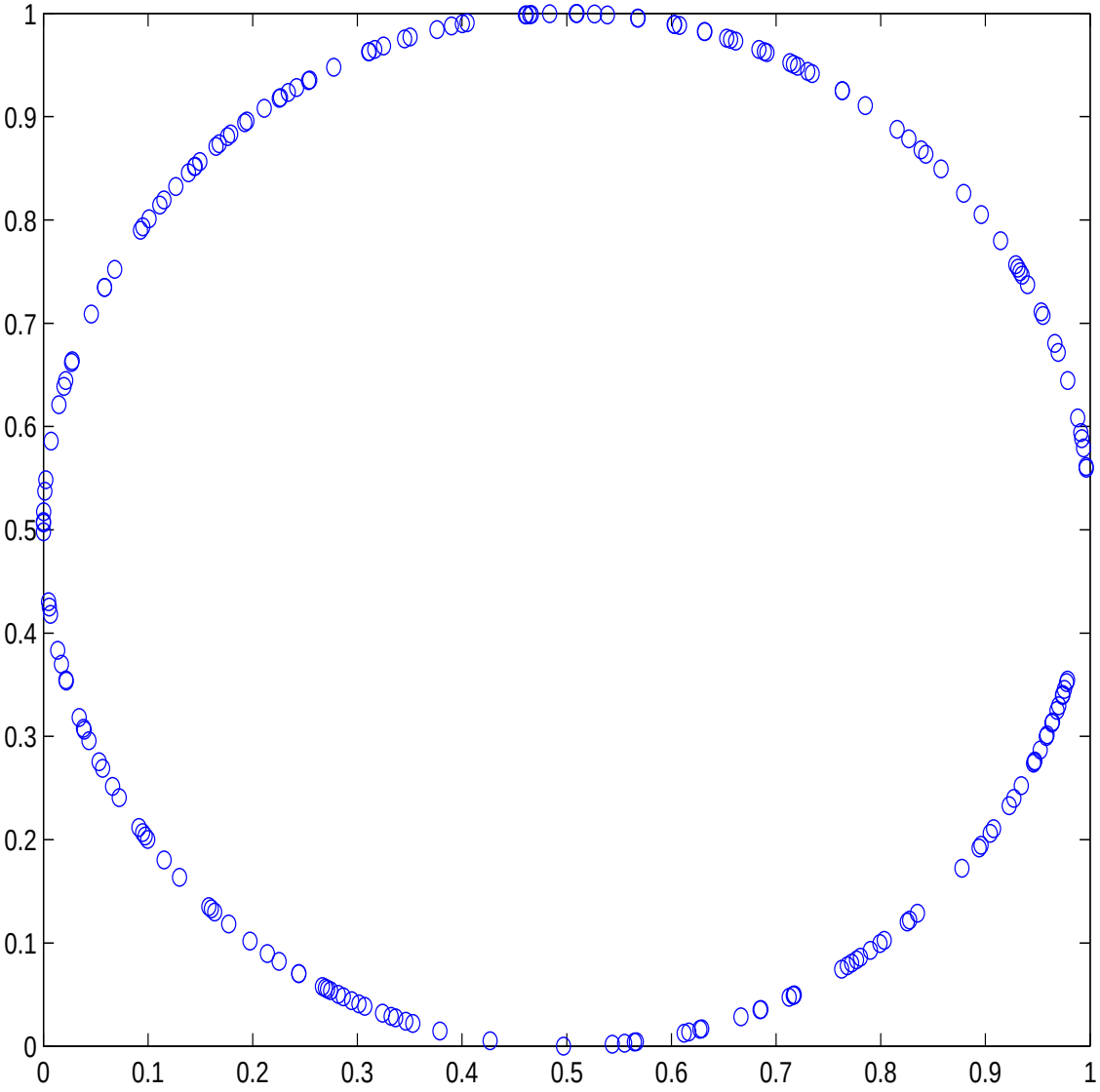

Figure: input data set uniformly distributed on a circle;

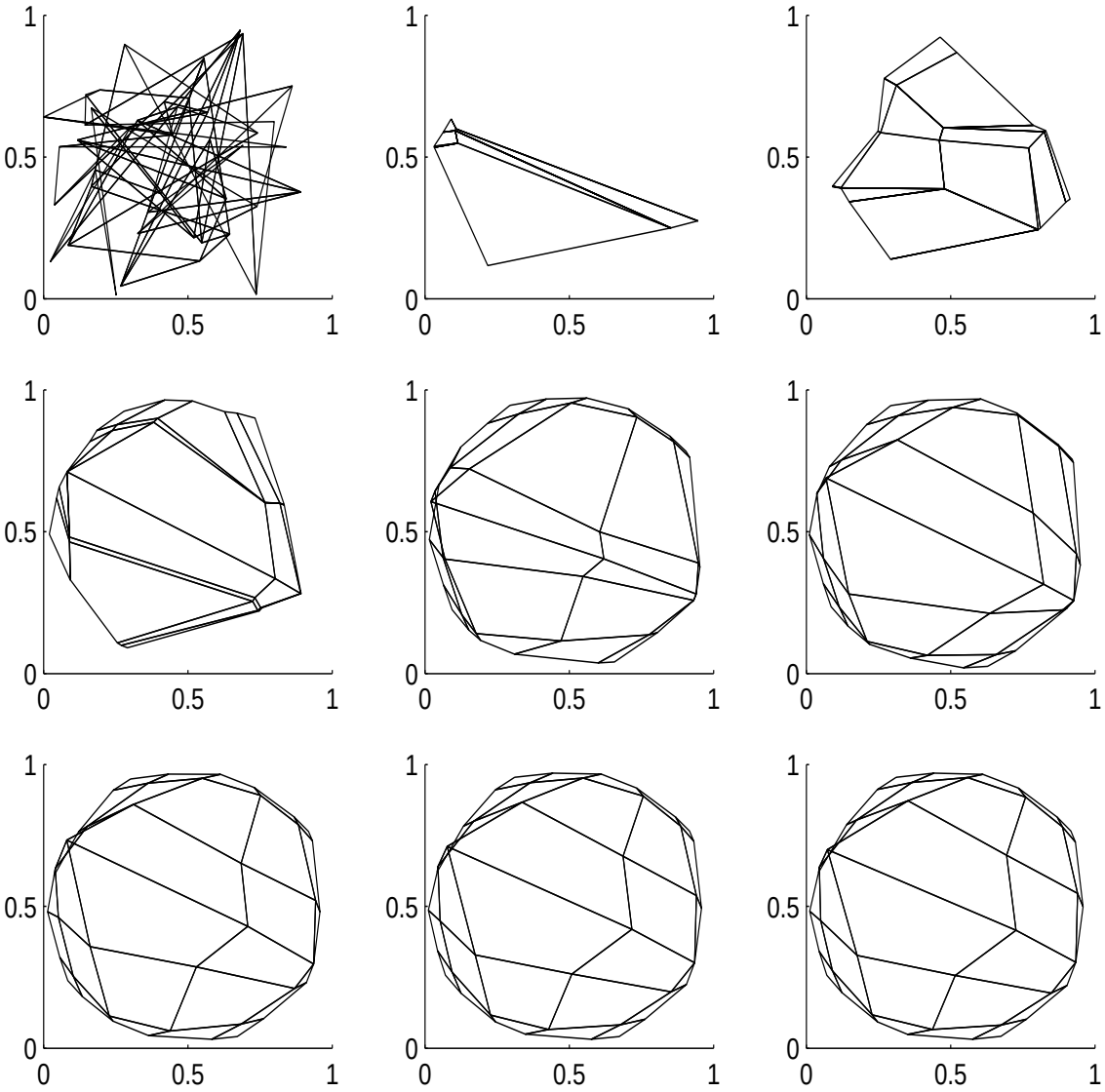

Figure: The evolution of SOM for data set sampled from a circle.