Lecture 8
Self-Organizing Maps

Sven Krüger
skrueger@iesk.et.uni-magdeburg.de

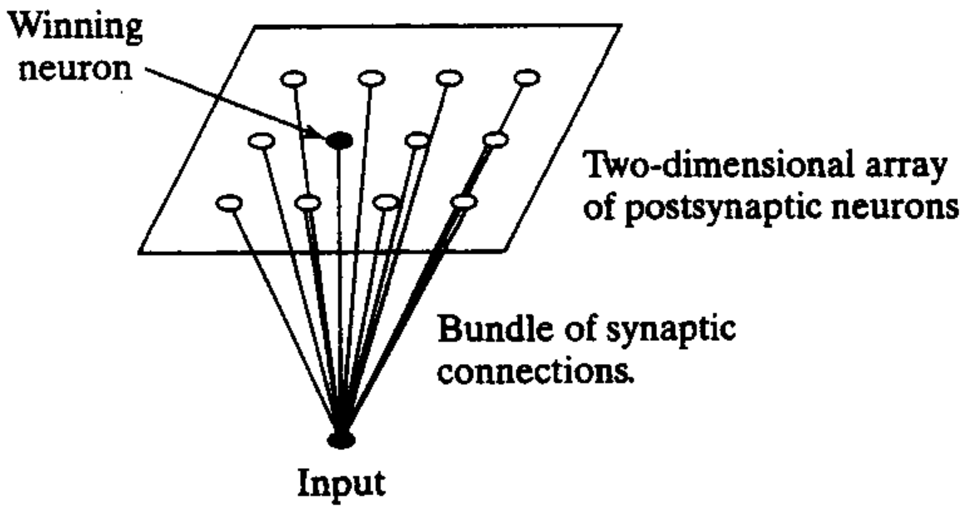
Introduction

self-organizing maps: a class of artificial neural networks
based on *competitive learning* – output neurons compete among themselves
⇒ only one neuron is fired at any one time (*winner-takes-all neuron*)

self-organizing map (SOM)

- neurons are placed at the nodes of a (usually) one- or two dimensional lattice
- locations of winning neurons become ordered with respect to each other
  → meaningful coordinate system for different input *features* is created

→ self-organizing map is characterized by the formation of a topographic map of the input pattern in which *spatial locations* of the neurons in the lattice are indicative of statistical features of the input pattern



Kohonen model

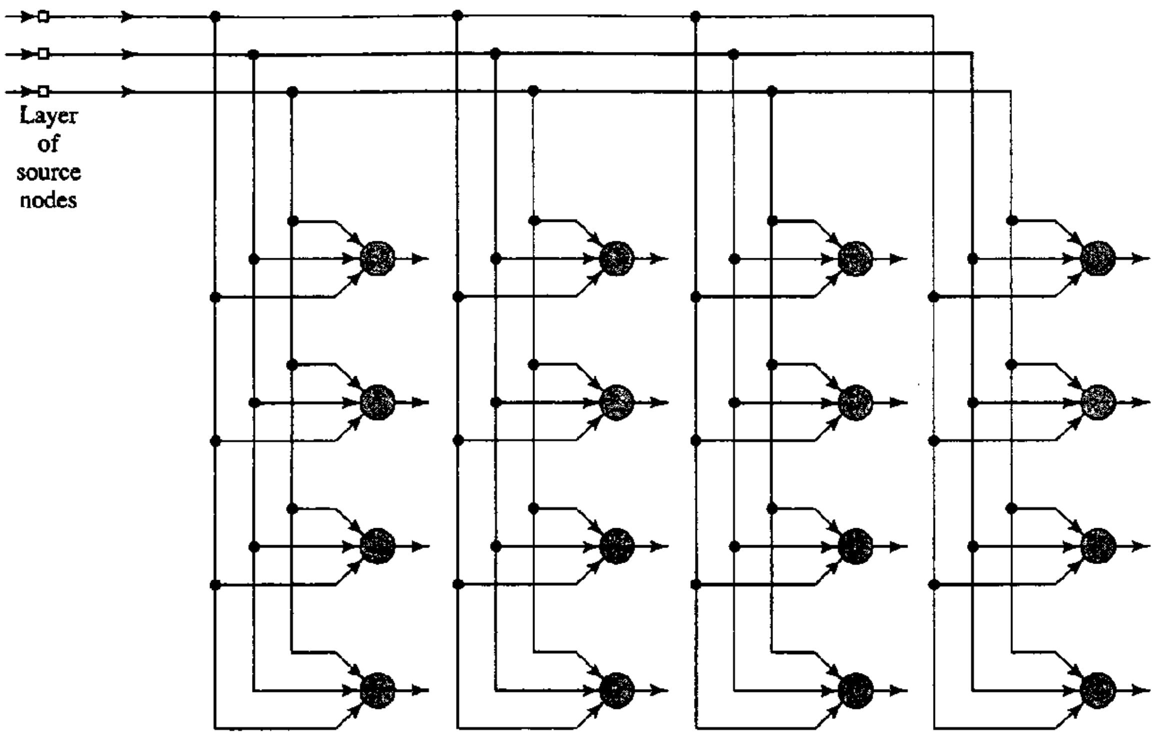
neurobiological motivation by a distinct feature of the brain:
different sensory inputs (e.g., visual) are mapped onto corresponding areas of the cerebral cortex in a *topologically ordered* manner, i.e. inputs are represented by topologically ordered *computational maps*

principle of topographic map formation (Kohonen):
The spatial location of an output neuron in a topographic map corresponds to a particular domain or feature of data drawn from the input space.
→ provides neurobiological motivation for *feature-mapping models* (such as SOM)

Training algorithm for SOM

principle goal of the SOM:
to transform (adaptively in a topologically ordered fashion) an incoming signal pattern of arbitrary dimension in a one- or two-dimensional discrete map

schematic diagram of a 2D-lattice of neurons
(each neuron of the lattice is fully connected to all node of the input layer)



Two-dimensional lattice of neurons.

three processes for the formation of the SOM:

- *Competition*: neurons compute values of a discriminant function
  → competition among neurons
  largest value of discriminant function → winner
- *Cooperation*: with excited neurons in a topological neighborhood
- Synaptic Adaptation: applied to all neurons inside the topological neighborhood such that response of winning neurons to similar input pattern is enhanced

Competitive Process

$\mathbf{x}$ – input patter (m dimensional)
$\mathbf{w}_j$ – synaptic weight vector of neuron j (m dimensional)
l – total number of neurons on the lattice

winning neuron: the one with the largest inner product $\mathbf{w}_j\mathbf{x}$
equivalent to minimizing Euclidean distance

$$i(\mathbf{x}) = \arg\min_j ||\mathbf{x} - \mathbf{w}_j||, \quad j = 1, 2, \ldots, l$$

neuron i is *best-matching* or *winning neuron* for input vector $\mathbf{x}$

Cooperative Process

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

neurobiological evidence: *lateral interaction* among set of excited neurons
$\rightarrow$ topological neighborhood should decay smoothly

$h_{j,i}$ – topological neighborhood centered on winning neuron i
$d_{j,i}$ – lateral distance between winning neuron i and excited neuron j

topological neighborhood $h_{j,i}$ should be unimodal function of $d_{i,j}$
typical choice:

$$h_{j,i(\mathbf{x})} = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2}\right)$$

in 2D:

$$d_{j,i}^2 = ||\mathbf{r}_j - \mathbf{r}_i||^2$$

unique feature of SOM: size of topological neighborhood should shrink with time step n:

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1}\right)$$

$$\Rightarrow h_{j,i(\mathbf{x})}(n) = \exp\left(-\frac{d_{j,i}^2}{2\sigma^2(n)}\right), \quad n = 0, 1, 2, \ldots$$

Adaptive Process

Hebbian learning $\propto y_j\mathbf{x}$ with a *forgetting term* $g(y_j)\mathbf{w}_j$ (with response y_j)

$$\Delta\mathbf{w}_j = \eta y_j \mathbf{x} - g(y_j)\mathbf{w}_j$$

simplifications:

$$g(y_j) = \eta y_j, \quad \text{and} \quad y_j = h_{j,i(\mathbf{x})}$$

it follows:

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

finally, using discrete-time formalism (applied to all neurons inside the topological neighborhood of winning neuron i):

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

with learning-rate parameter

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2}\right)$$

effect of the updating process: moving synaptic weight vector $\mathbf{w}_i$ of winning neuron i toward the input vector $\mathbf{x}$
→ weight vectors tend to follow the distribution of the input vectors
→ algorithm leads to a topological ordering of feature map in input space (ie., adjacent neurons have similar weight vectors)

Summary of the SOM Algorithm

1. *Initialization.* Choose random values for the initial weight vectors $\mathbf{w}_j(0)$. The only restriction here is that the $\mathbf{w}_j(0)$ be different for $j = 1, 2, \ldots, l$, where l is the number of neurons in the lattice. It may be desirable to keep the magnitude of the weights small.

   Another way of initalizing the algorithm is to select the weight vectors $\{\mathbf{w}_j(0)\}_{j=1}^{l}$ from the available set of input vectors $\{\mathbf{x}_i\}_{i=1}^{N}$ in a random manner.
2. *Sampling.* Draw a sample $\mathbf{x}$ from the input space with a certain probability; the vector $\mathbf{x}$ represents the activation pattern that is applied to the lattice. The dimension of vector $\mathbf{x}$ is equal to m.
3. *Similarity Matching.* Find the best-matching (winning) neuron $i(\mathbf{x})$ at time step n by using the minimum-distance Euclidean criterion:

$$i(\mathbf{x}) = \arg\min_j \|\mathbf{x}(n) - \mathbf{w}_j\|, \quad j = 1, 2, \ldots, l$$

4. *Updating.* Adjust the synaptic weight vectors of all neurons by using the update formula

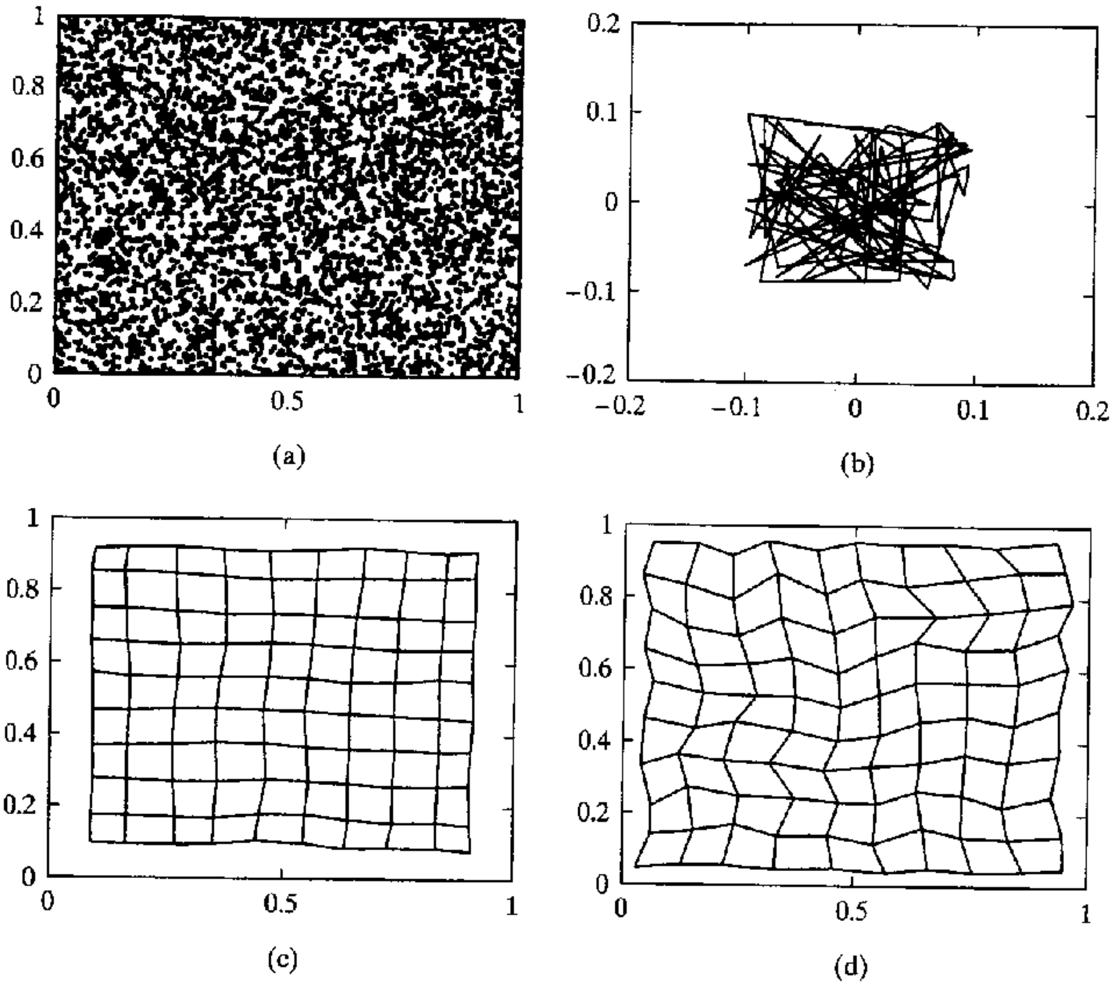
$$\mathbf{w}_j(n+1) = \mathbf{w}_j(n) + \eta(n) h_{j,i(\mathbf{x})}(n)(\mathbf{x}(n) - \mathbf{w}_j(n))$$

   where $\eta(n)$ is the learning-rate parameter, and $h_{j,i(\mathbf{x})}(n)$ is the neighborhood function centered around the winning neuron $i(\mathbf{x})$; both $\eta(n)$ and $h_{j,i(\mathbf{x})}(n)$ are varied dynamically during learning for best results.
5. *Continuation.* Continue with step 2 until no noticeable changes in the feature map are observed.

Computer Simulations

2D lattice driven by 2D input distribution

two-dimensional input vectors **x**, uniformly distributed
100 neurons, two-dimensional lattice, (10×10)



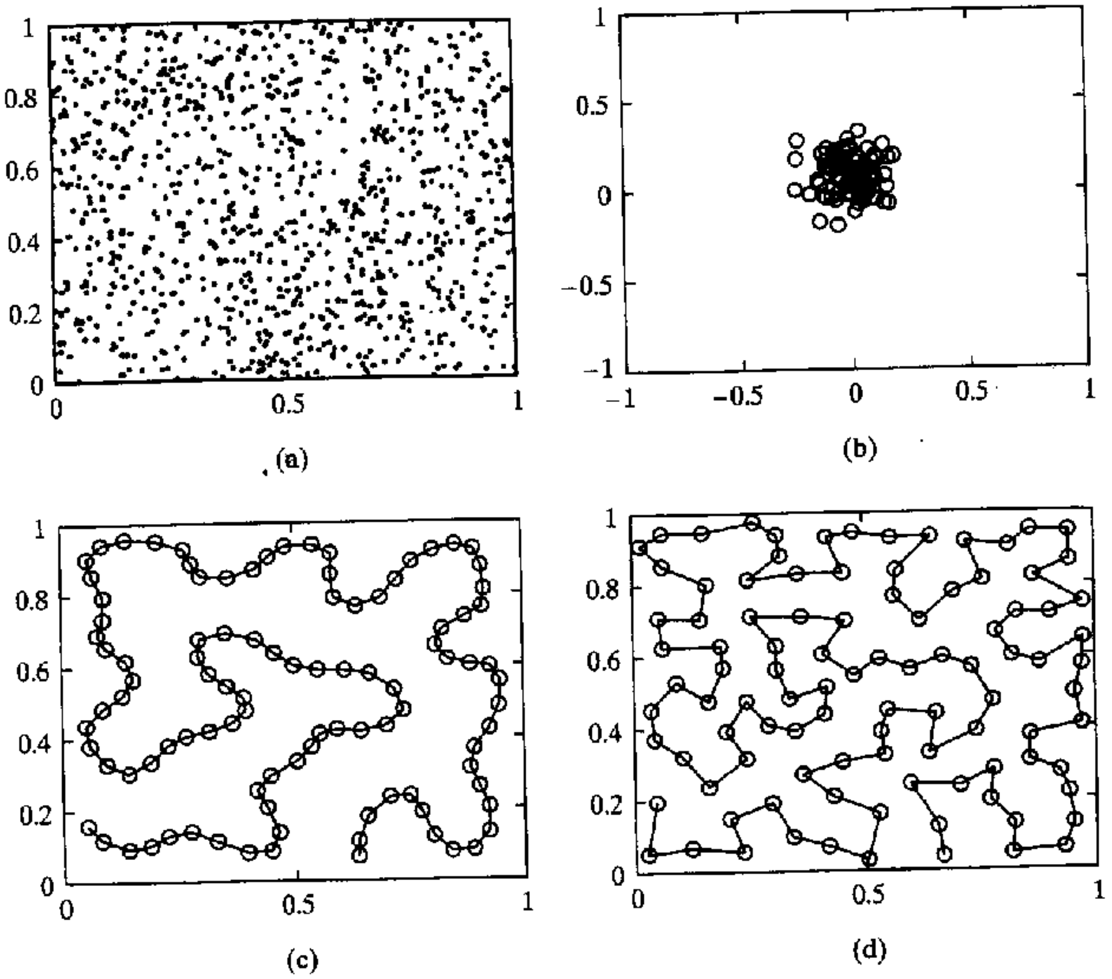
(a) Input data distribution. (b) Initial condition of the two-dimensional lattice. (c) Condition of the lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase.

two phases of adaptive process:

- *Ordering:* topological ordering of weight vectors
- *Convergence:* fine tune the feature map, ie., provide accurate statistical quantification of input space

1D lattice driven by 2D input distribution

two-dimensional input vectors **x**, uniformly distributed
100 neurons in one dimension



(a) Two-dimensional input data distribution. (b) Initial condition of the one-dimensional lattice. (c) Condition of the lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase.

parameter specification for this simulation
ordering phase: (1000 iterations)
$\sigma(n)$: $\sigma_0 = 18$, shrinks to about 1
$\eta(n)$: $\eta_0 = 0.1$, decreases to 0.037

convergence phase: (5000 iterations)
$\eta(n)$: decreases linearly from 0.037 to 0.001

Exercise 1: Write a matlab program to do the simulation in 1D!

Properties of the feature map

SOM algorithm computes the *feature map*
feature map: displays important statistical characteristics of the input space

$\mathcal{H}$ – spatially continuous input (data) space ($\mathbf{x} \in \mathcal{H}$)
$\mathcal{A}$ – spatially discrete output space ($\mathbf{w}_j \in \mathcal{A}$)
Φ – feature map (a nonlinear transformation), $\Phi : \mathcal{H} \rightarrow \mathcal{A}$

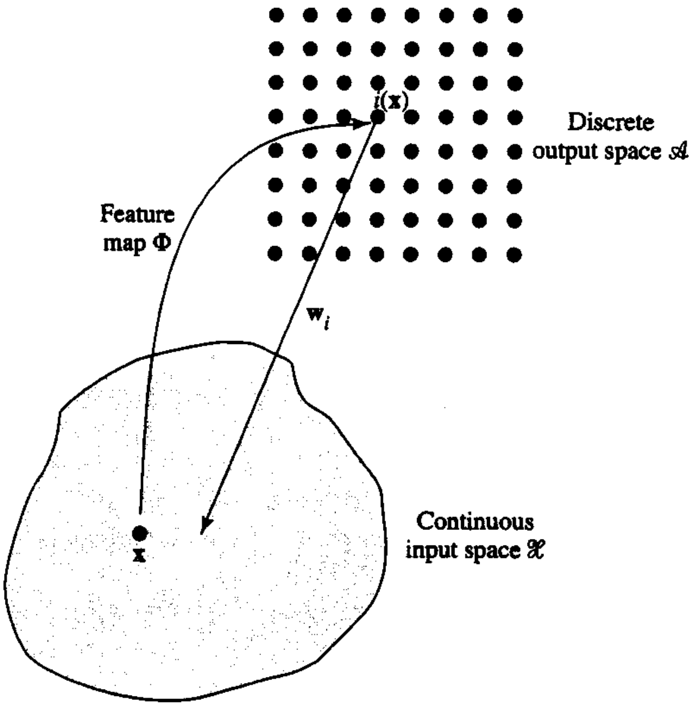


Illustration of the relationship between feature map Φ and weight vector $\mathbf{w}_i$ of winning neuron i.

properties of Φ

- *Approximation of the input space*
  represent large set of input vectors $\mathbf{x}$ with smaller set of prototypes $\mathbf{w}_j$
  $\rightarrow$ *vector quantization theory*
- *Topological ordering*
  spatial location of a neuron in the lattice corresponds to a particular domain or feature of input patterns
- *Density matching*
- *Feature selection*
  ability to select a set of best features
  feature map can be viewed as a *nonlinear* generalization of PCA

Exercise 2: Sometimes it is said that SOM *preserves* the topological relationships (that exist in the input space) in the neural lattice. Discuss this statement!