Chapter 9.
Self-Organizing Maps

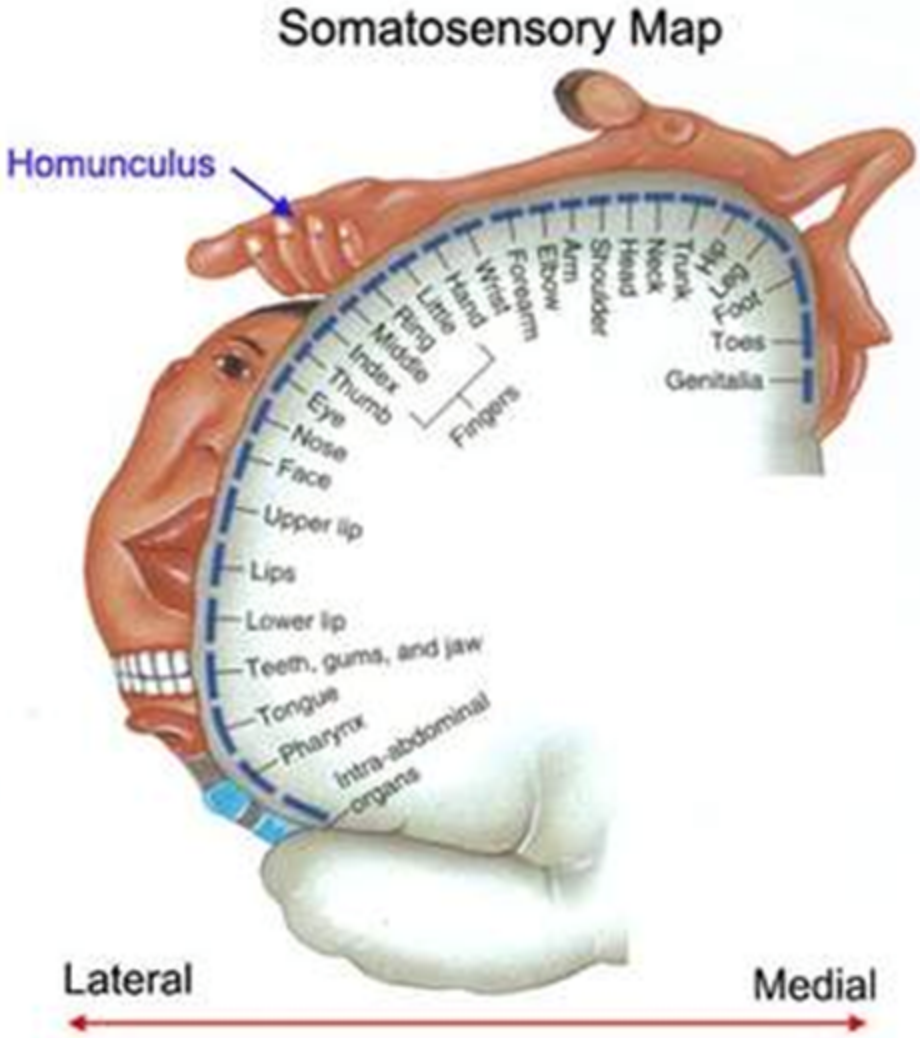
Neural Networks and Learning Machines
(Haykin)

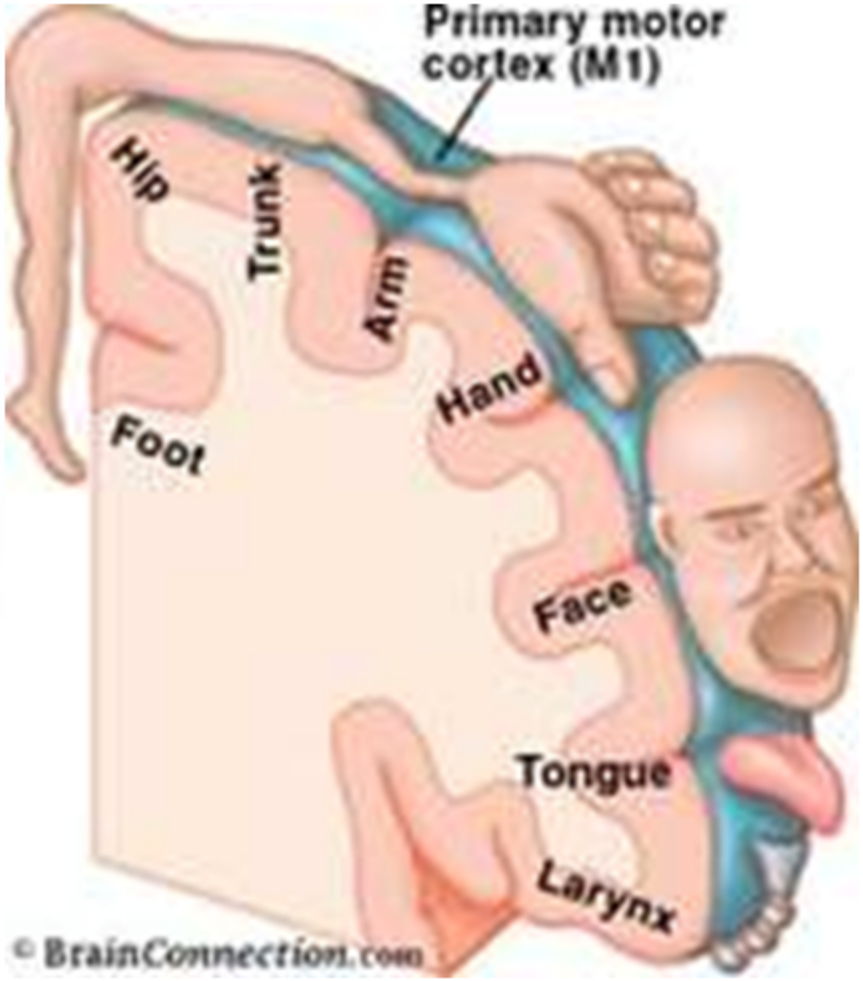
Lecture Notes on
Self-learning Neural Algorithms
(Version 2017.09.13)

Byoung-Tak Zhang
School of Computer Science and Engineering
Seoul National University

Contents

9.1 Introduction (1/2)



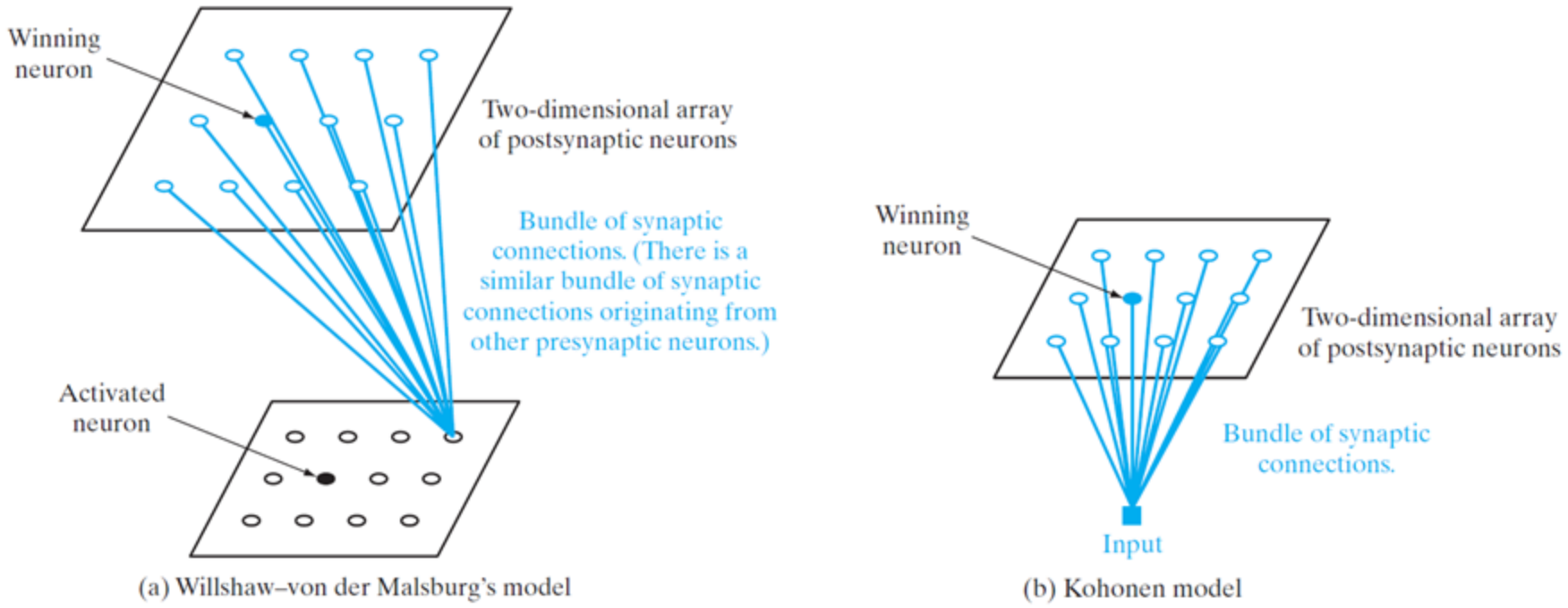


9.1 Introduction (1/2)

- Self-organizing feature maps
  - Competitive learning, nonlinear
  - Winner-takes-all neurons
  - Topographic (topology-preserving) maps
  - Lattice structure
  - Place-coded probability distribution
- A self-organizing map is a topographic map of the input patterns, in which the spatial locations (i.e. coordinates) of the neurons in the lattice are indicative of intrinsic statistical features contained in the input patters.
- The brain is organized in many places in such a way that different sensory inputs are represented by topologically ordered computational maps.

9.2 Two Basic Feature Mapping Models (1/2)

Figure 9.1: Two self-organized feature maps.



Willshaw-von der Malsburg Model

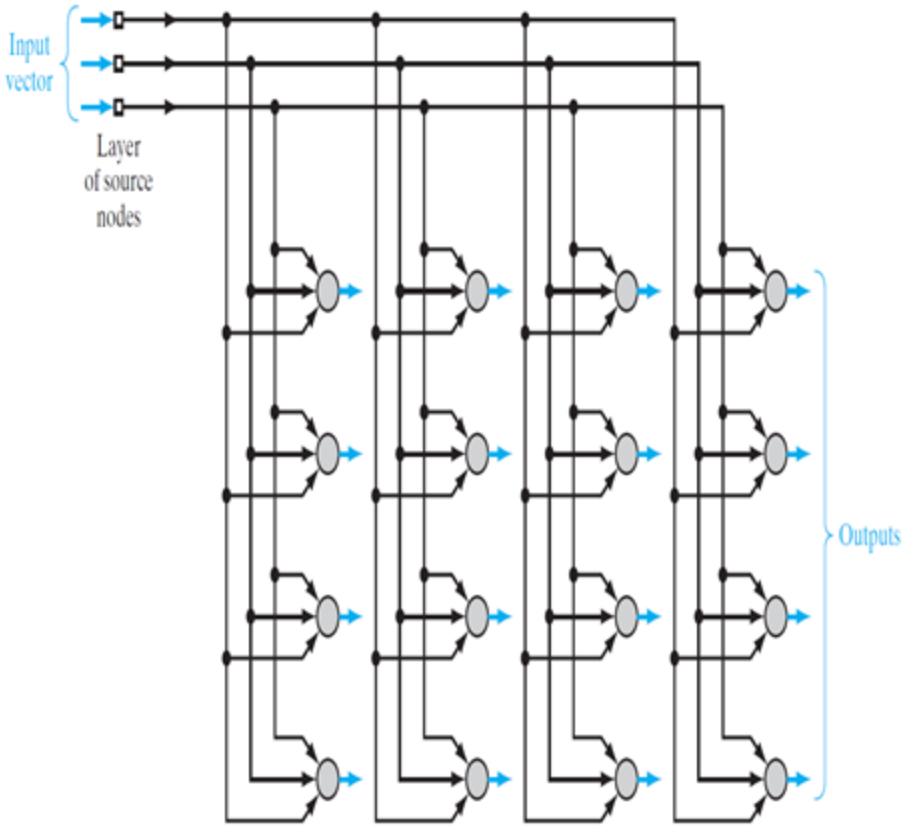
Kohonen Model

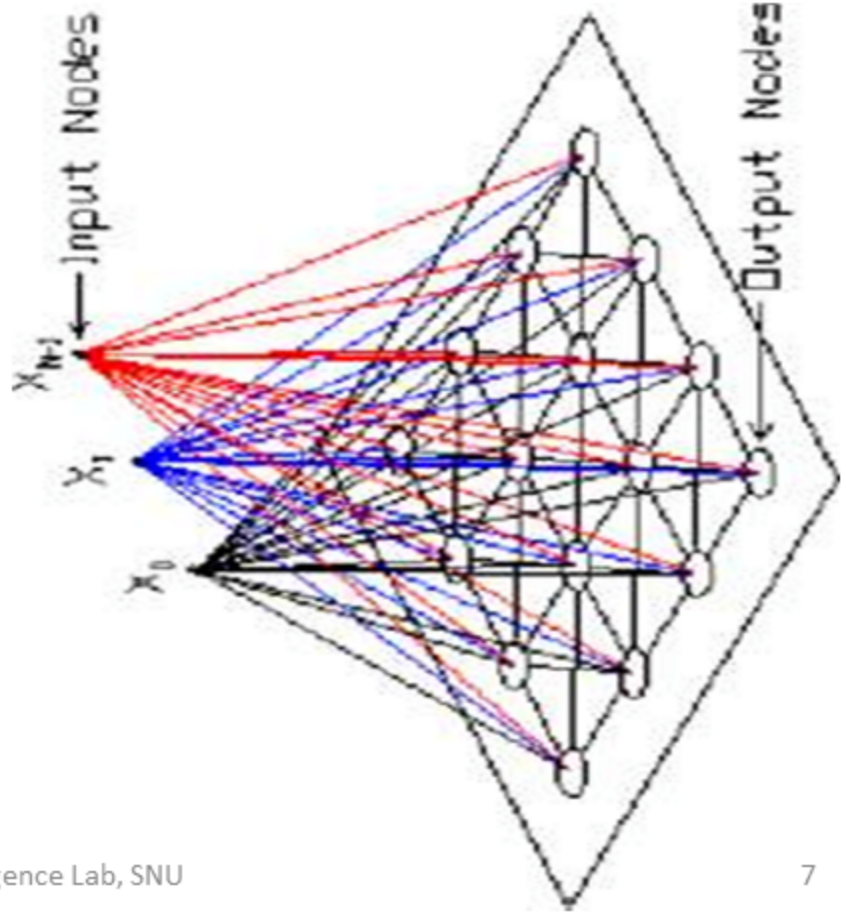
9.2 Two Basic Feature Mapping Models (2/2)

- Principle of topographic map formation: 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
- Willshaw-von der Malsburg Model
  - Map from input lattice to output lattice (same dimensions)
  - Not winner-take-all neurons (but thresholds)
  - Topologically ordered mapping
  - Short-range excitation and long-range inhibition
  - Model of retinotopic mapping from retina to visual cortex
- Kohonen Model
  - Map from input (no lattice) to output lattice
  - Winner-take-all neurons
  - Vector-coding algorithm

9.3 Self-Organizing Map (1/4)

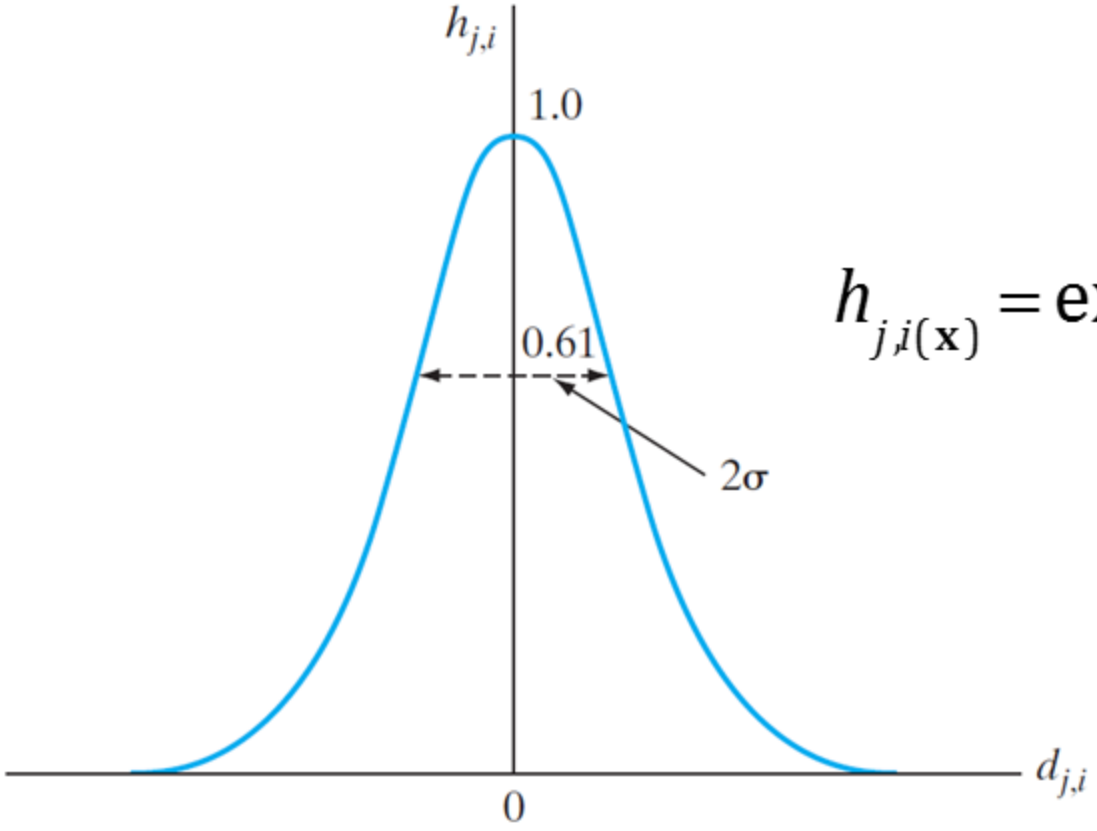
Figure 9.2: Two-dimensional lattice of neurons, illustrated for a three-dimensional input and four-by-four dimensional output (all shown in blue).





9.3 Self-Organizing Map (2/4)

Figure 9.3: Gaussian neighborhood function.



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

9.3 Self-Organizing Map (3/4)

SOM Algorithm

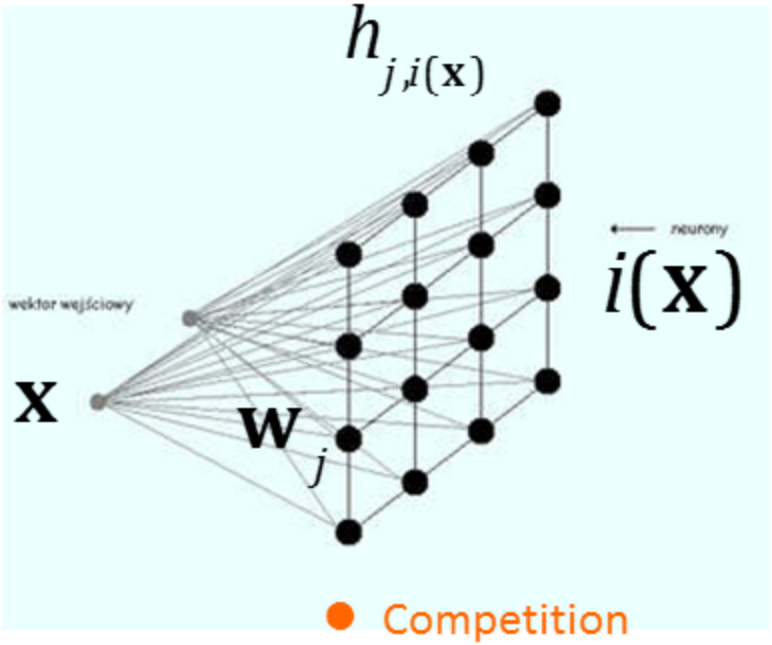
1. Initialization. Weights $\mathbf{w}_j(0)$

   - Random, different, small magnitude

2. Sampling. Input $\mathbf{x}$

3. Similarity matching.

$$i(\mathbf{x}) = \arg\min_j ||\mathbf{x}(n) - \mathbf{w}_j||$$



- Competition
- Cooperation
- Synaptic adaptation

1. Updating.

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

9.3 Self-Organizing Map (4/4)

- Two phases of the adaptive process
  - Ordering phase and convergence phase
- Adaptation of neighborhood and learning rate parameters

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

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

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau_2} \right), \quad n = 0,1,2,\ldots$$

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau_1} \right), \quad n = 0,1,2,\ldots$$

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

SOM Animations

- https://www.youtube.com/watch?v=zyYZuAQZWTM
- https://www.youtube.com/watch?v=b3nG4c2NECI&t=35s
- https://www.youtube.com/watch?v=k7DK5fnJH94

- https://www.youtube.com/watch?v=lttfH2nwdb4&t=9s
- https://www.youtube.com/watch?v=dASyjPQtbS8&t=27s
- https://www.youtube.com/watch?v=3YhiU2_uk5I
- https://www.youtube.com/watch?v=71wmOT4lHWc&t=38s

- https://www.youtube.com/watch?v=WIGxS-quGSo
  (Growing Neural Gas Algorithm)

- https://www.youtube.com/watch?v=GdZckTLNqsY
- **(Video Lecture)**

9.4 Properties of the Feature Map (1/5)

Property 1. Approximation of the Input Space

The feature map, represented by the set of synaptic weight vectors in the output space, provides a good approximation to the input space.

Property 2. Topological Ordering

The feature map computed by the SOM algorithm is topologically ordered in the sense that the spatial location of the neuron in the lattice corresponds to a particular domain or feature of input patterns.

Property 3. Density Matching

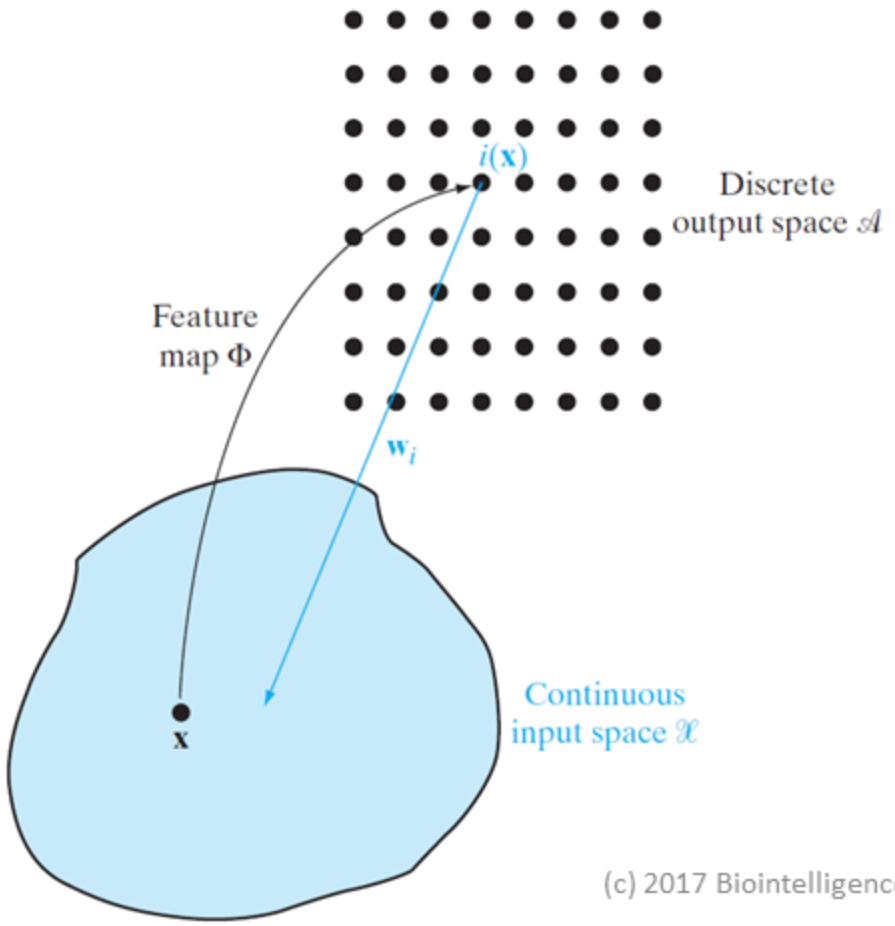
Regions in the input space from which sample vectors are drawn with a high probability of occurrence are mapped onto larger domains of the output space.

Property 4. Feature Selection

Given data from an input space, the self-organizing map is able to select a set of best features for approximating the underlying distribution.

9.4 Properties of the Feature Map (2/5)

Figure 9.4: Illustration of the relationship between feature map Φ and weight vector w_i of winning neuron i.

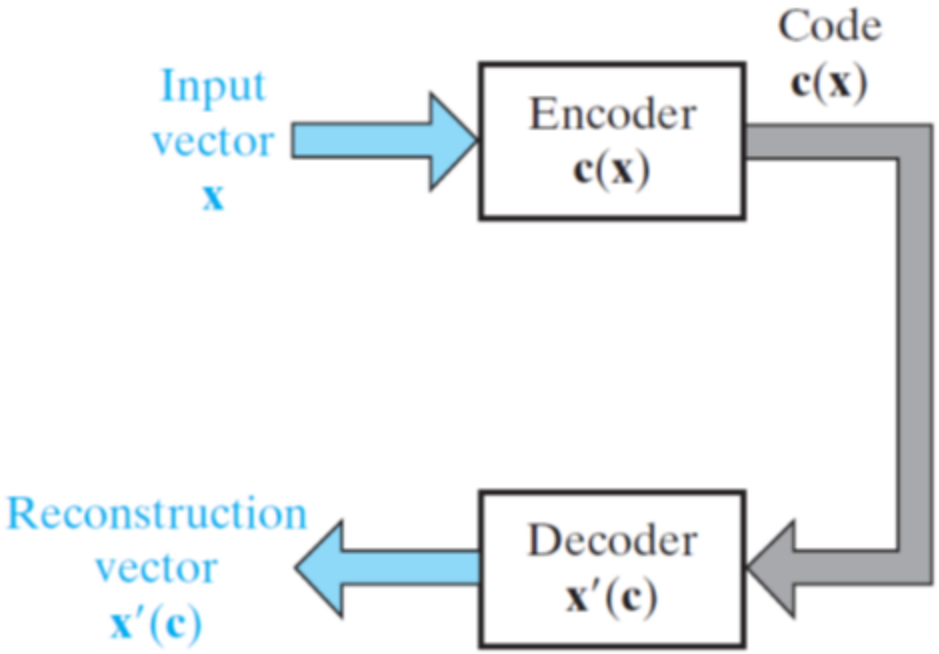


Aim: to store a large set of input vectors by finding a smaller number of Prototypes so as to provide a good approximation to the original input space

$$i(\mathbf{x}) = \arg\min_{j} ||\mathbf{x} - \mathbf{w}_j||$$

9.4 Properties of the Feature Map (3/5)

Figure 9.5: Encoder–decoder model for describing Property 1 of the SOM model.



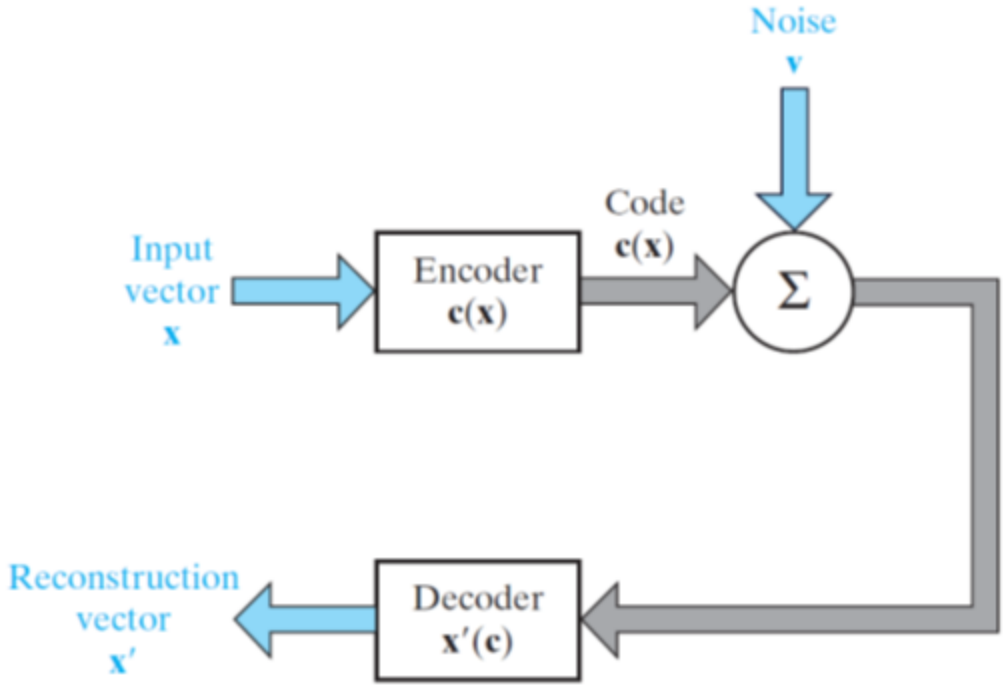
Expected distortion

$$D = \frac{1}{2}\int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) d(\mathbf{x},\mathbf{x}') d\mathbf{x}$$

$$= \frac{1}{2}\int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) ||\mathbf{x}-\mathbf{x}'||^2 \, d\mathbf{x}$$

9.4 Properties of the Feature Map (4/5)

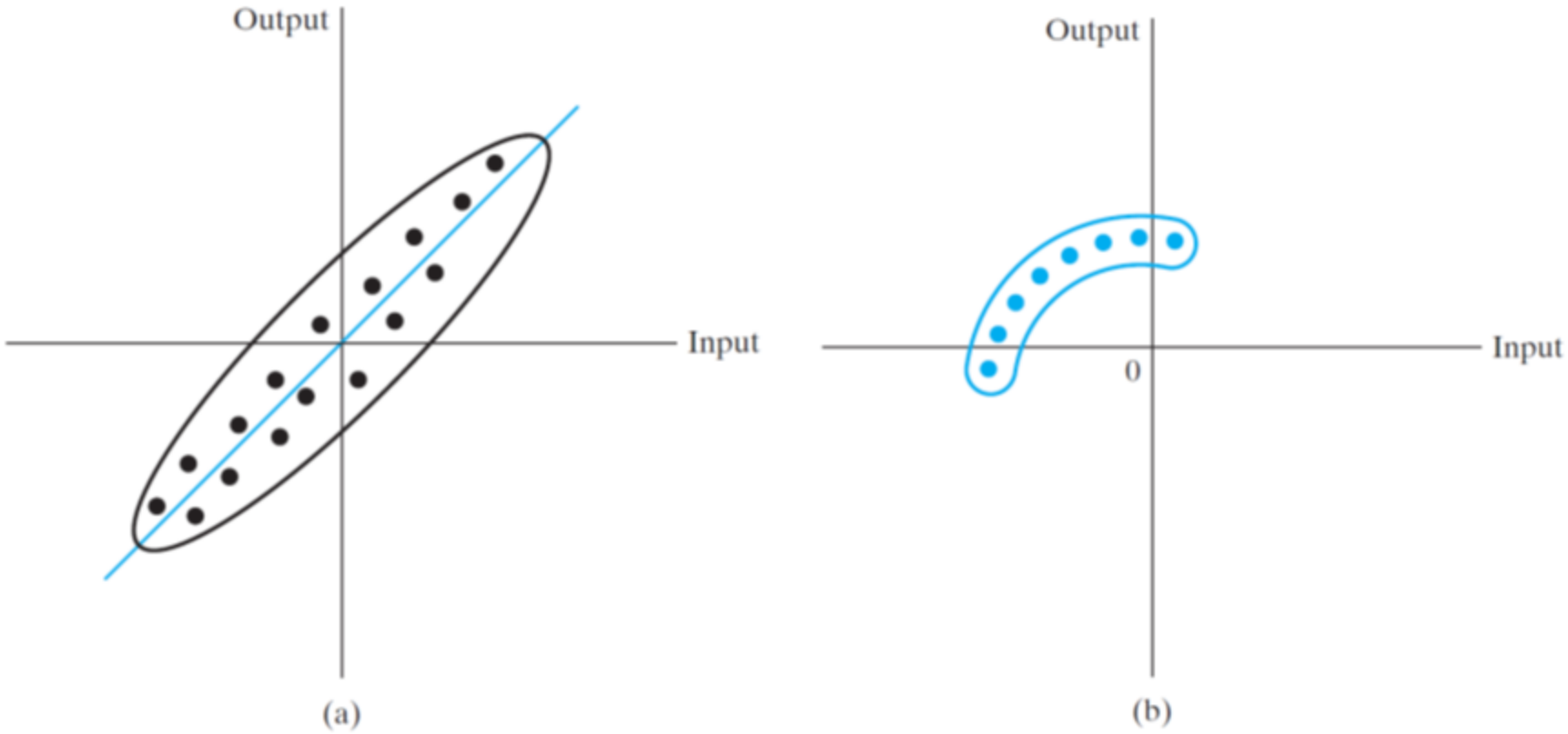
Figure 9.6: Noisy encoder–decoder model.



$$D_1 = \frac{1}{2}\int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) \int_{-\infty}^{\infty} \pi(\mathbf{x}) ||\mathbf{x} - \mathbf{x}'(\mathbf{c}(\mathbf{x}) + \mathbf{v})||^2 \, d\mathbf{v} \, d\mathbf{x}$$

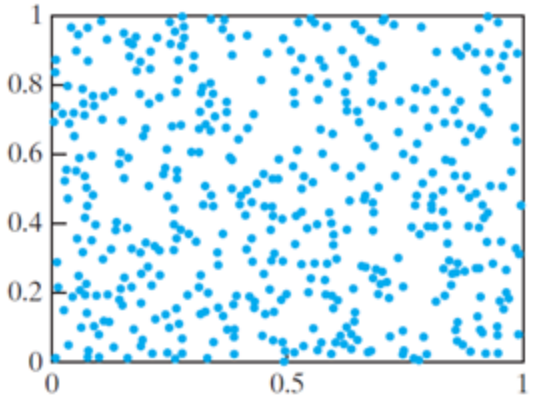
9.4 Properties of the Feature Map (5/5)

Figure 9.7: (a) Two-dimensional distribution produced by a linear input–output mapping. (b) Two dimensional distribution produced by a nonlinear input–output mapping.

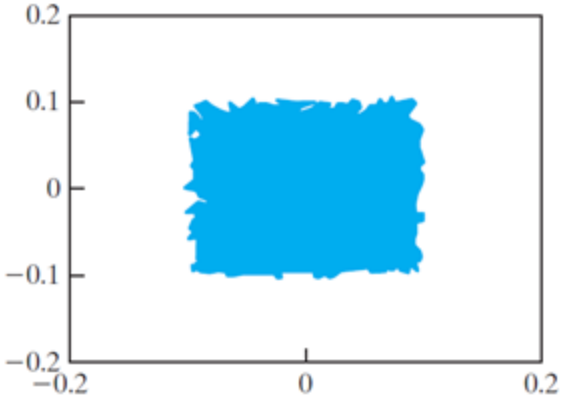


9.5 Computer Experiments: Disentangling Lattice Dynamics Using SOM (1/2)

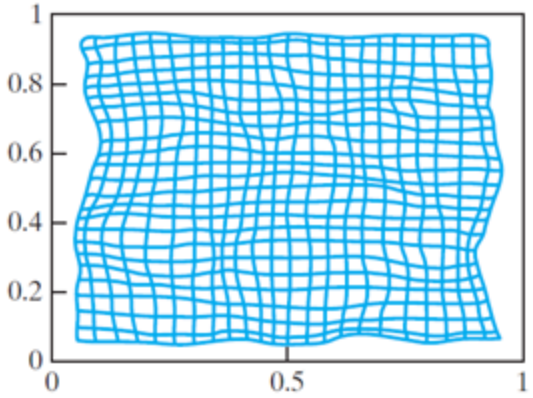
Figure 9.8: (a) Distribution of the input data. (b) Initial condition of the two-dimensional lattice. (c) Condition of the lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase. The times indicated under maps (b), (c), and (d) represent the numbers of iterations.



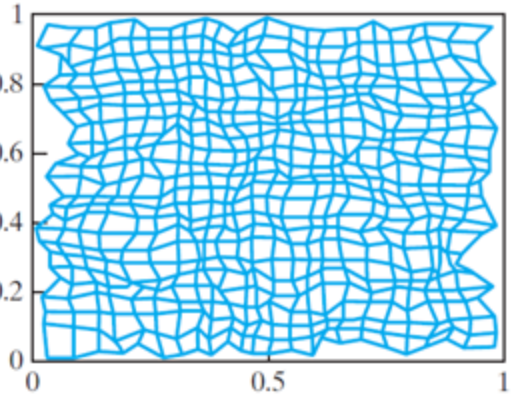
(a) Input distribution



Time = 0
(b) Initial weights



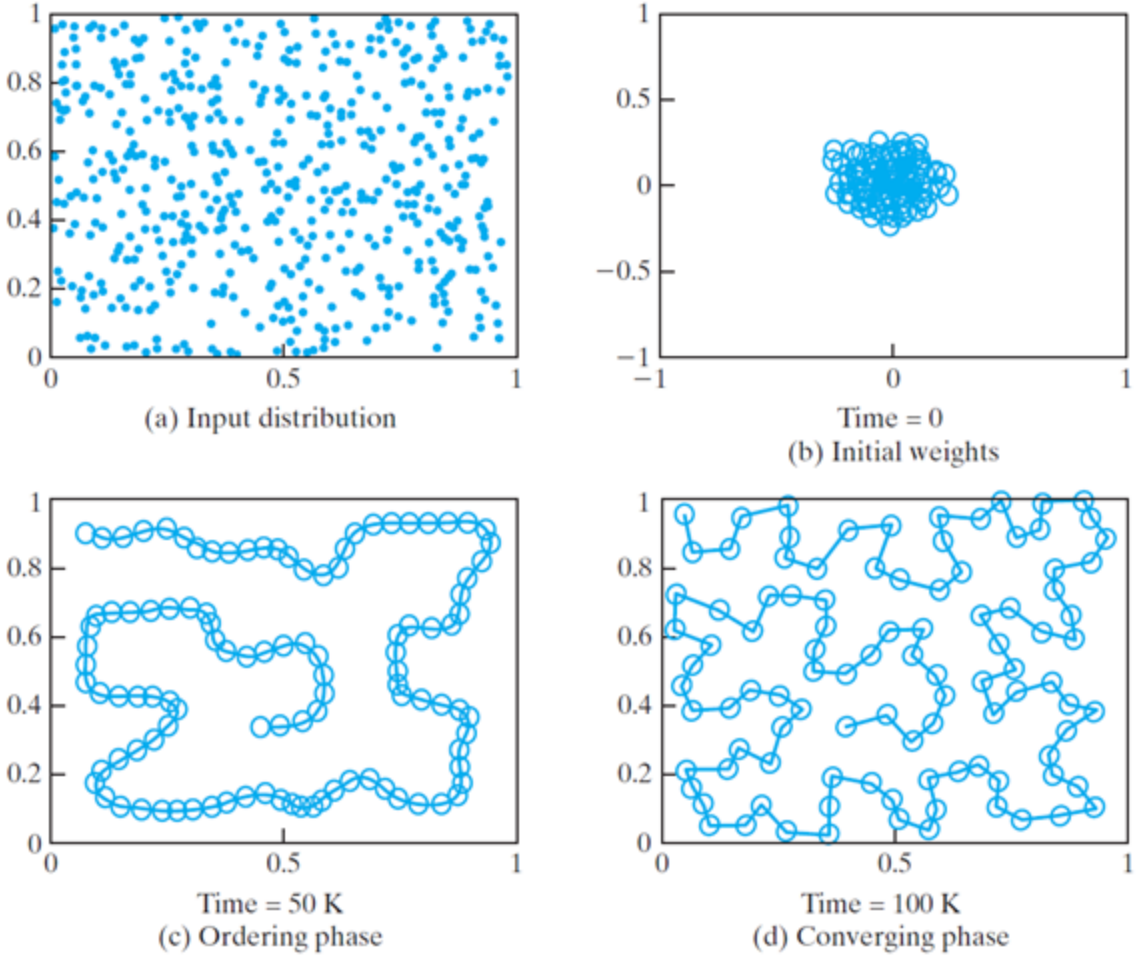
Time = 160 K
(c) Ordering phase



Time = 800 K
(d) Convergence phase

9.5 Computer Experiments: Disentangling Lattice Dynamics Using SOM (2/2)

Figure 9.9: (a) Distribution of the two-dimensional input data. (b) Initial condition of the one-dimensional lattice. (c) Condition of the one-dimensional lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase. The times included under maps (b), (c), and (d) represent the numbers of iterations.



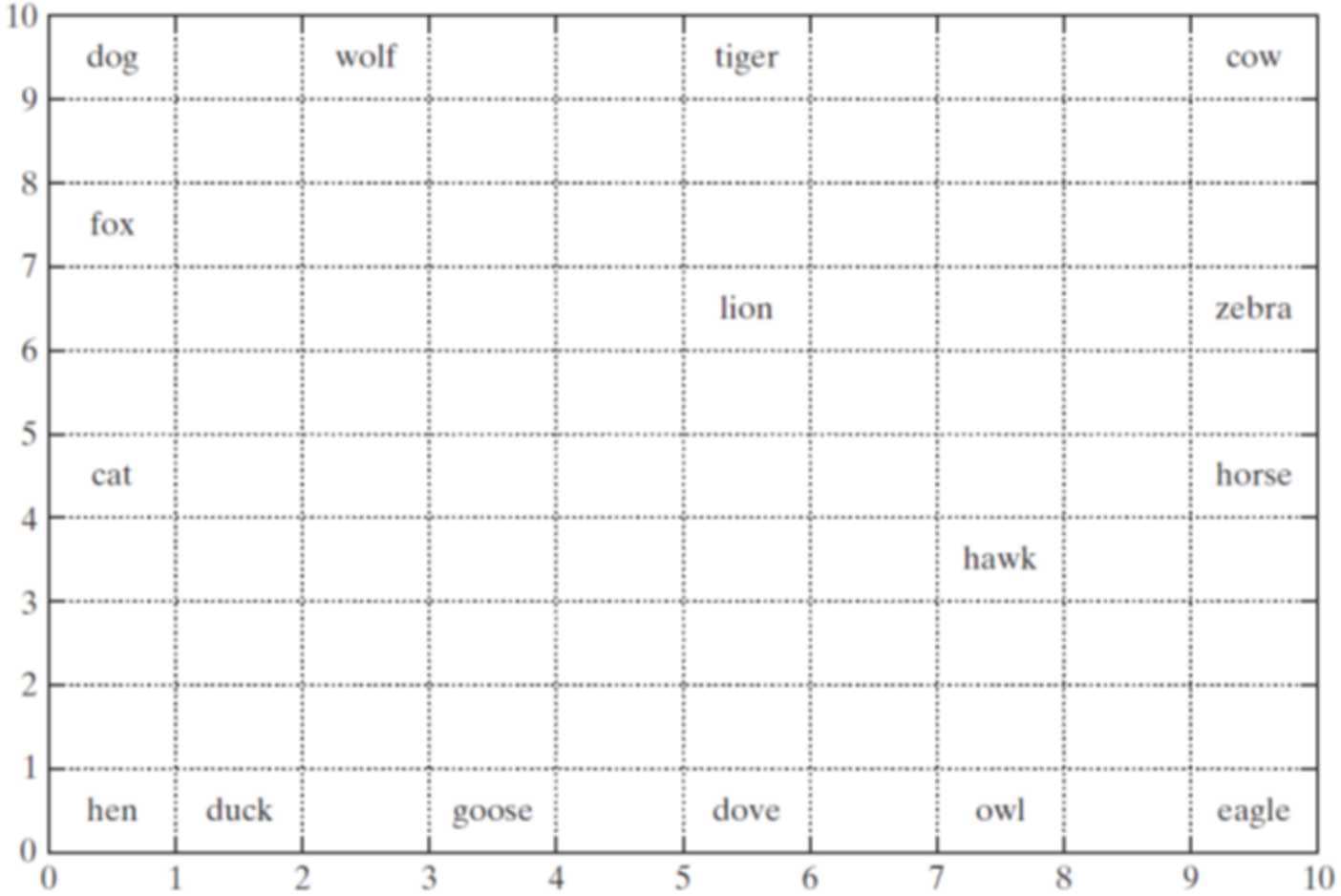
9.6 Contextual Maps

Table 9.2 Animal names and their attributes

Animal		Dove	Hen	Duck	Goose	Owl	Hawk	Eagle	Fox	Dog	Wolf	Cat	Tiger	Lion	Horse	Zebra	Cow
is	small	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0
	medium	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
	big	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
has	2 legs	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
	4 legs	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hair	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
	hooves	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
	mane	0	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0
	feathers	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
likes to	hunt	0	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0
	run	0	0	0	0	0	0	0	0	1	1	0	1	1	1	1	0
	fly	1	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0
	swim	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0

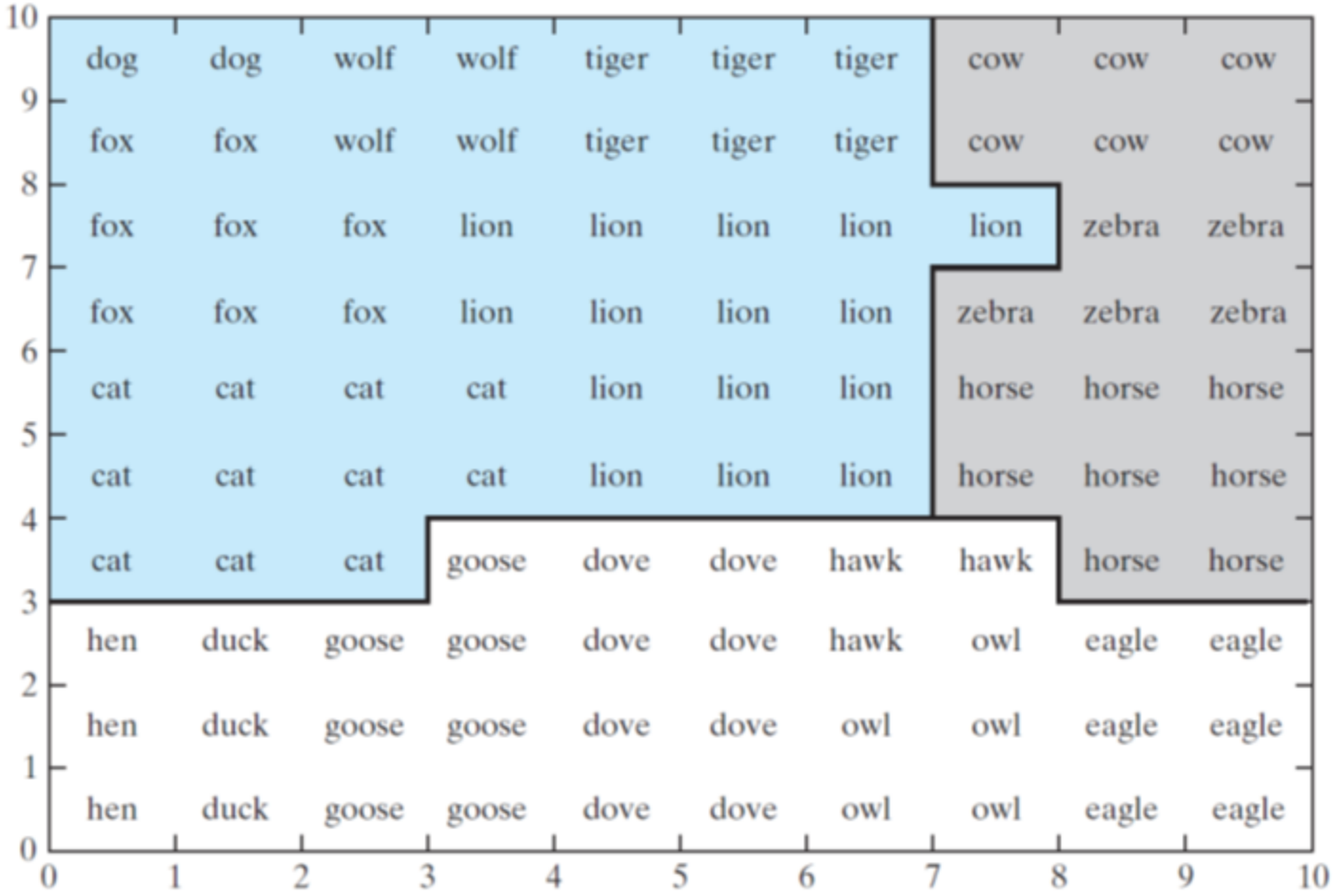
9.6 Contextual Maps

Figure 9.10: Feature map containing labeled neurons with strongest responses to their respective inputs.



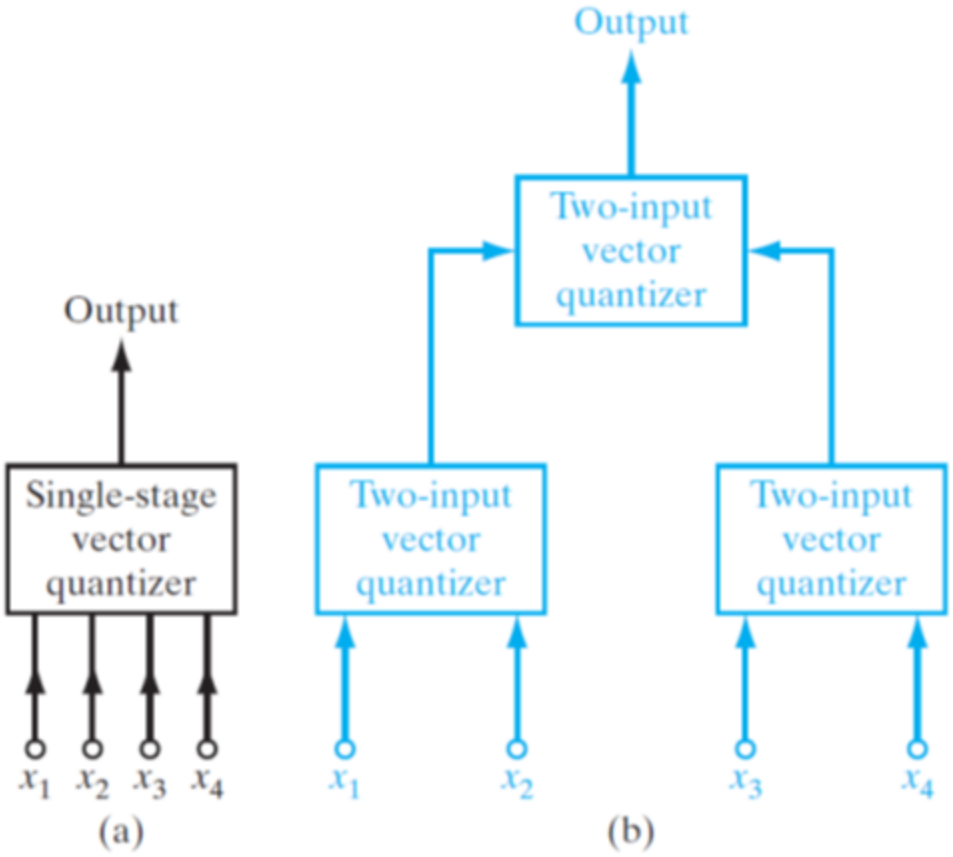
9.6 Contextual Maps

Figure 9.11: Semantic map obtained through the use of simulated electrode penetration mapping. The map is divided into three regions, representing birds (white), peaceful species (grey), and hunters (red).



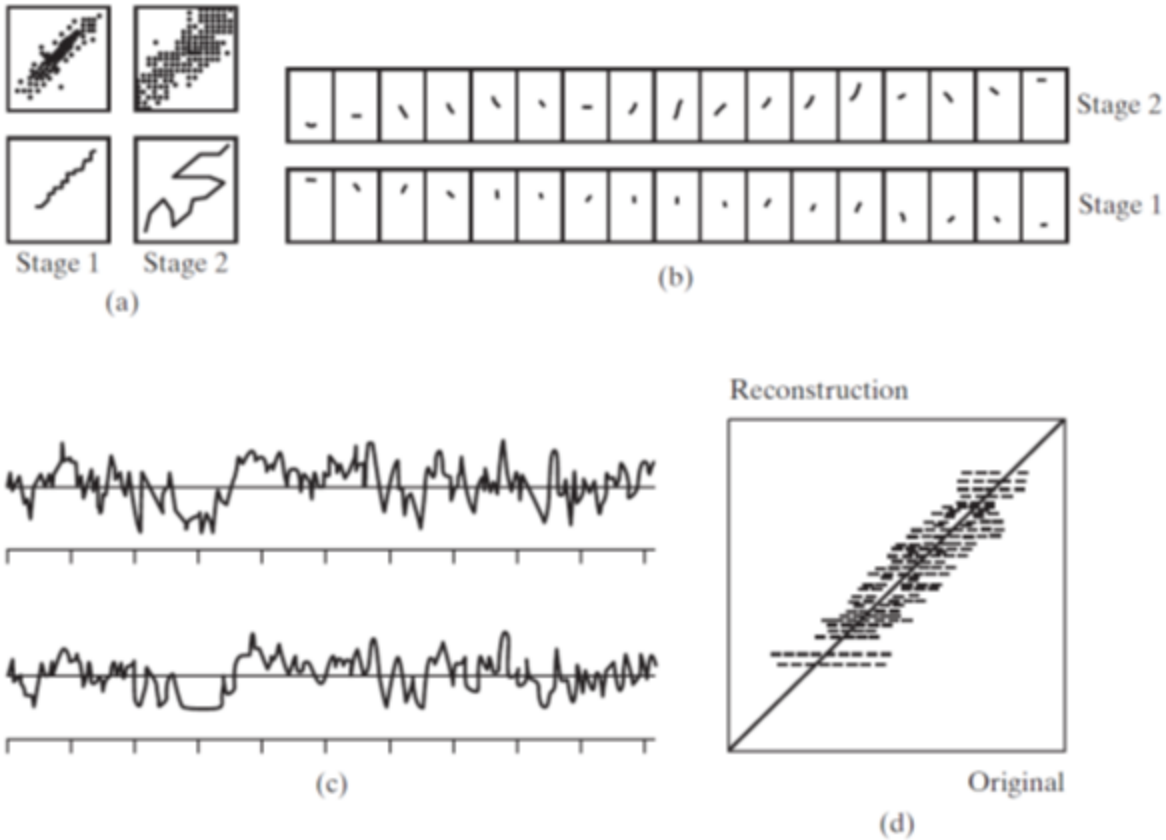
9.7 Hierarchical Vector Quantization (1/2)

Figure 9.12: (a) Single-stage vector quantizer with fourdimensional input. (b) Two-stage hierarchical vector quantizer using two-input vector quantizers. (From S.P. Luttrell, 1989a, British Crown copyright.)



9.7 Hierarchical Vector Quantization (2/2)

Figure 9.13: Two-stage encoding–decoding results, using the binary tree shown in red in Fig. 9.12, for the compression of correlated Gaussian noise input. Correlation coefficient & = 0.85.
(From S.P. Luttrell, 1989a, British Crown copyright.)



9.8 Kernel Self-Organizing Map (1/5)

- Objective function: Joint entropy of the kernel (i.e., neural) outputs

$$H(Y_i) = -\int_{-\infty}^{\infty} p_{Y_i}(y_i) \log p_{Y_i}(y_i) dy_i$$

$$y_i = k(\mathbf{x}, \mathbf{w}_i, \sigma_i)$$

- Definition of the kernel

$$k(\mathbf{x}, \mathbf{w}_i, \sigma_i) = k\left(||\mathbf{x} - \mathbf{w}_i||, \sigma_i\right)$$

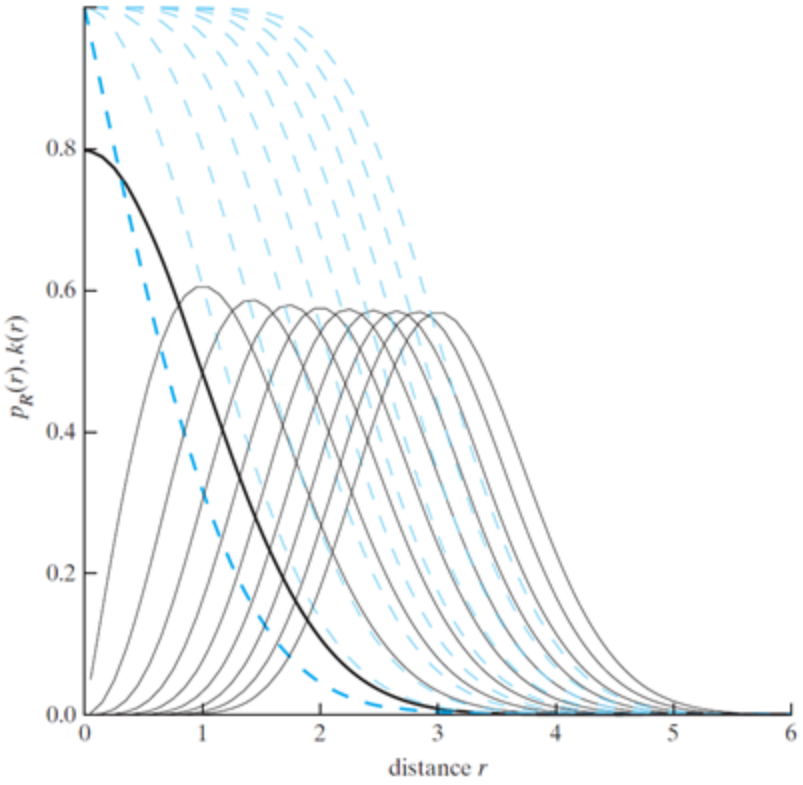
$$r = ||\mathbf{x} - \mathbf{w}_i|| \ : \text{incomplete gamma distribution}$$

$$k(\mathbf{x}, \mathbf{w}_i, \sigma_i) = \frac{1}{\Gamma(\frac{m}{2})} \Gamma\left(\frac{m}{2}, \frac{||\mathbf{x} - \mathbf{w}_i||^2}{2\sigma_i^2}\right), \ i = 1, 2, ..., \ell$$

9.8 Kernel Self-Organizing Map (2/5)

Figure 9.14: Two different sets of plots versus the distance r are shown in the figure for unit variance and increasing dimensionality m = 1, 2, 3, ...:

- The continuous curves (printed in black) are plots of the probability density function of Eq. (9.41).
- The dashed curves (printed in red) are plots of the complement of the incomplete gamma distribution or, equivalently, kernel k(r) of Eq. (9.44) with $r = \|\boldsymbol{x} - \boldsymbol{w}\|$.



9.8 Kernel Self-Organizing Map (3/5)

From SOM to Kernel SOM

- Similarity matching

$$i(\mathbf{x}) = \arg\min_j ||\mathbf{x} - \mathbf{w}_j||$$

$$\Rightarrow \quad i(\mathbf{x}) = \arg\min_j y_i = \arg\min_j k(\mathbf{x}, \mathbf{w}_i, \sigma_i)$$

- Neighborhood function

$$h_{j,i(\mathbf{x})} = \exp\left(-\frac{||\mathbf{x} - \mathbf{w}_j||^2}{2\sigma^2} \right), \quad j \in A$$

9.8 Kernel Self-Organizing Map (4/5)

Kernel SOM Algorithm

1. Initialization. Weights $\mathbf{w}_j(0)$

   - Random, different, small magnitude

2. Sampling. Input $\mathbf{x}$

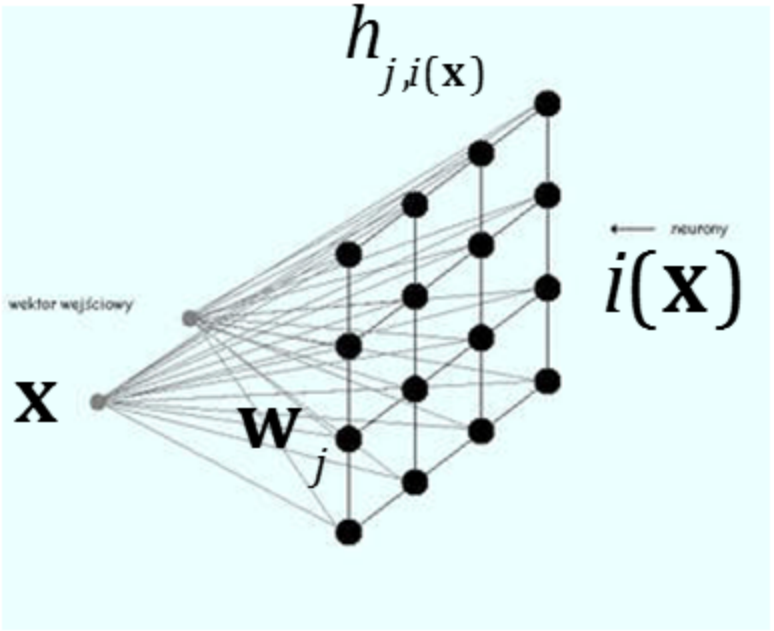
3. Similarity matching.

$$i(\mathbf{x}) = \arg\min_j y_i$$

$$= \arg\min_j k(\mathbf{x}, \mathbf{w}_i, \sigma_i)$$

1. Updating.

(see next slide)



9.8 Kernel Self-Organizing Map (5/5)

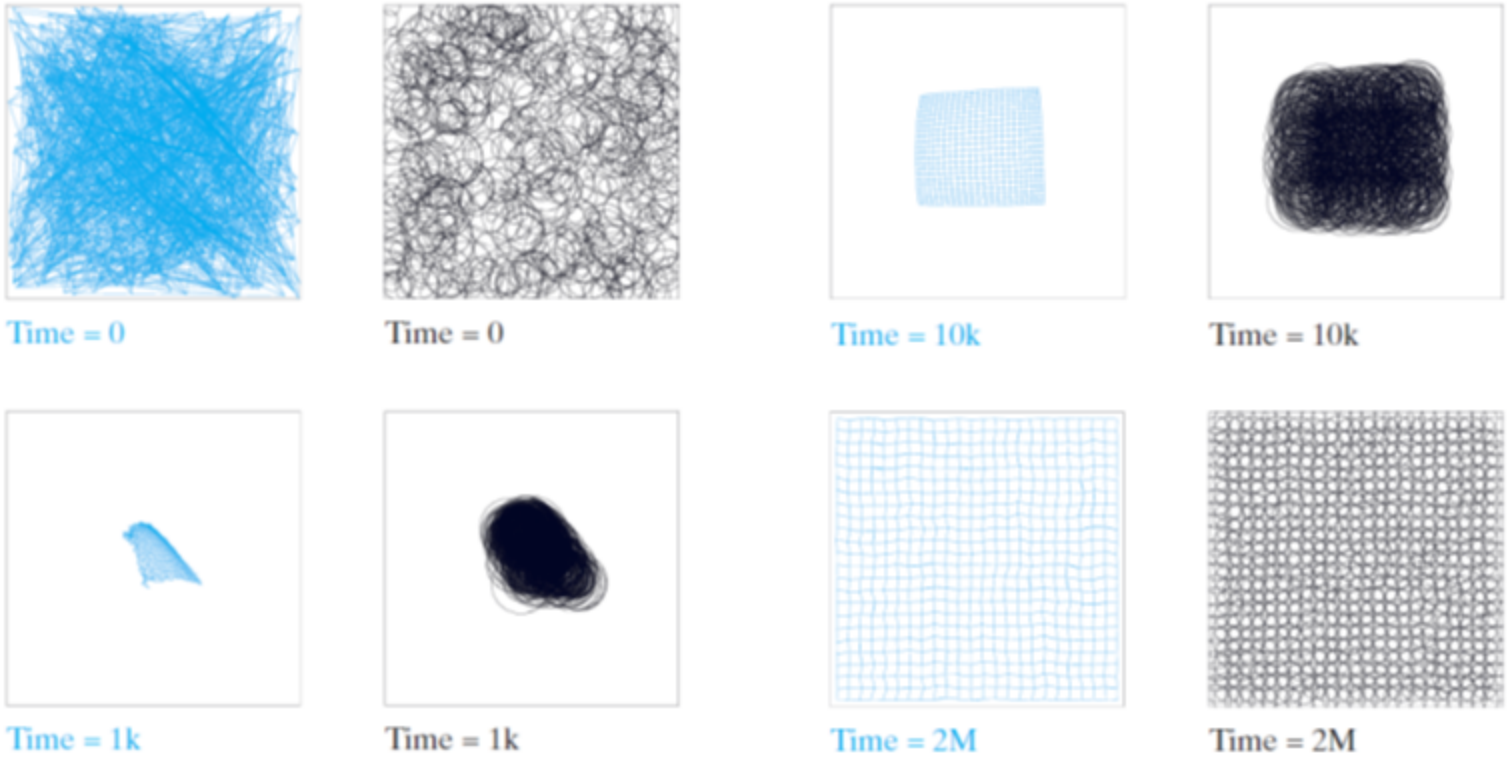
Kernel SOM: Update Equations

$$\mathbf{w}_j(n+1) = \begin{cases} \mathbf{w}_j(n) + \dfrac{\eta_{\mathbf{w}} h_{j,i(\mathbf{x})}}{\sigma_j^2}(\mathbf{x}(n) - \mathbf{w}_j(n)), & j \in A \\ \mathbf{w}_j(n), & \text{otherwise} \end{cases}$$

$$\sigma_j(n+1) = \begin{cases} \sigma_j(n) + \dfrac{\eta_{\sigma} h_{j,i(\mathbf{x})}}{\sigma_j(n)} \left[\dfrac{\|(\mathbf{x}(n) - \mathbf{w}_j(n)\|^2}{m\sigma_j^2(n)} - 1 \right], & j \in A \\ \sigma_j(n), & \text{otherwise} \end{cases}$$

9.9 Computer Experiments: Disentangling Lattice Dynamics Using Kernel SOM

Figure 9.15: The evolution of a 24-by-24 lattice over time, the values of which (in terms of the number of iterations) are given below each picture. Left column: Evolution of the kernel weights. Right column: Evolution of the kernel widths. Each box in the figure outlines the result of a uniform input distribution.The time given below each map represents the number of iterations. (This figure is reproduced with the permission of Dr. Marc Van Hulle.)



9.10 Relationship Between Kernel SOM and Kullback-Leibler Divergence (1/3)

Quality of density estimate $\hat{p}_{\mathbf{x}}(\mathbf{x})$ against true density $p_{\mathbf{x}}(\mathbf{x})$

$$D_{p_{\mathbf{x}}||\hat{p}_{\mathbf{x}}} = \int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) \log\left(\frac{p_{\mathbf{x}}(\mathbf{x})}{\hat{p}_{\mathbf{x}}(\mathbf{x})}\right) d\mathbf{x}$$

Density estimate as a mixture of Gaussian density function

$$\hat{p}_{\mathbf{x}}(\mathbf{x}) = \hat{p}_{\mathbf{x}}(\mathbf{x}|\mathbf{w}_i, \sigma_i) = \frac{1}{\ell}\sum_{i=1}^{\ell} \frac{1}{(2\pi)^{m/2}\sigma_i^m} \exp\left(-\frac{1}{2\sigma_i^2}||\mathbf{x}-\mathbf{w}_j||^2\right)$$

Partial derivative w.r.t. $\mathbf{w}_i$

$$\frac{\partial}{\partial \mathbf{w}_i}\left(D_{p_{\mathbf{x}}||\hat{p}_{\mathbf{x}}}\right) = \frac{\partial}{\partial \mathbf{w}_i}\int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) \log\left(\frac{p_{\mathbf{x}}(\mathbf{x})}{\hat{p}_{\mathbf{x}}(\mathbf{x}|\mathbf{w}_i, \sigma_i)}\right) d\mathbf{x}$$

$$= \int_{-\infty}^{\infty} \frac{\partial}{\partial \mathbf{w}_i}\left(p_{\mathbf{x}}(\mathbf{x}) \log p_{\mathbf{x}}(\mathbf{x}) - p_{\mathbf{x}}(\mathbf{x}) \log \hat{p}_{\mathbf{x}}(\mathbf{x}|\mathbf{w}_i, \sigma_i)\right) d\mathbf{x}$$

$$= -\int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) \frac{\partial}{\partial \mathbf{w}_i}\left(\log \hat{p}_{\mathbf{x}}(\mathbf{x}|\mathbf{w}_i, \sigma_i)\right) d\mathbf{x}$$

$$= -\int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}) \left(\frac{1}{\hat{p}_{\mathbf{x}}(\mathbf{x}|\mathbf{w}_i, \sigma_i)} \frac{\partial}{\partial \mathbf{w}_i} \hat{p}_{\mathbf{x}}(\mathbf{x}|\mathbf{w}_i, \sigma_i)\right) d\mathbf{x}$$

9.10 Relationship Between Kernel SOM and Kullback-Leibler Divergence (2/2)

Similarly, partial derivative w.r.t. σ_i

$$\frac{\partial}{\partial \sigma_i}\left(D_{p_{\mathbf{X}}\|\hat{p}_{\mathbf{X}}}\right) = -\int_{-\infty}^{\infty} p_{\mathbf{X}}(\mathbf{x})\left(\frac{1}{\hat{p}_{\mathbf{X}}(\mathbf{x}|\mathbf{w}_i,\sigma_i)}\frac{\partial}{\partial \sigma_i}\hat{p}_{\mathbf{X}}(\mathbf{x}|\mathbf{w}_i,\sigma_i)\right)d\mathbf{x}$$

Setting

$$\frac{\partial}{\partial \mathbf{w}_i}\left(D_{p_{\mathbf{X}}\|\hat{p}_{\mathbf{X}}}\right) \to 0$$

$$\frac{\partial}{\partial \sigma_i}\left(D_{p_{\mathbf{X}}\|\hat{p}_{\mathbf{X}}}\right) \to 0$$

and invoking stochastic approximation theory,

we obtain the learning rules

$$\Delta \mathbf{w}_i = \eta_{\mathbf{w}}\hat{p}_X(\mathbf{x}|\mathbf{w}_i,\sigma_i)\left(\frac{\mathbf{x}-\mathbf{w}_i}{\sigma_i^2}\right)$$

$$\Delta \sigma_i = \eta_{\mathbf{w}}\hat{p}_X(\mathbf{x}|\mathbf{w}_i,\sigma_i)\frac{m}{\sigma_i}\left(\frac{\|\mathbf{x}-\mathbf{w}_i\|^2}{m\sigma_i^2}-1\right)$$

9.10 Relationship Between Kernel SOM and Kullback-Leibler Divergence (3/3)

Suppose we set the conditional posterior density

$$\hat{p}_{\mathbf{X}}(\mathbf{x}_j \mid \mathbf{w}_i, \sigma_i) = \delta_{ji} \quad \text{for } j = 1,2,\ldots,\ell$$

$$\delta_{ji} = \begin{cases} 1 & \text{for } j = i \\ 0 & \text{for } j \neq i \end{cases}$$

When this ideal condition is satistifed, neuron i is the winning neuron.

We may therefore view $\hat{p}_{\mathbf{X}}(\mathbf{x} \mid \mathbf{w}_i, \sigma_i)$ as playing the role of the topological neighborhood function

$$\hat{p}_{\mathbf{X}}(\mathbf{x} \mid \mathbf{w}_i, \sigma_i) = h_{j,i(\mathbf{x})}$$

Minimization of the KL divergence is equivalent to maximization of the joint entropy (defined in terms of incomplete gamma distribution kernels and an activity-based neighborhood function which is the core of the kernel SOM).

Summary and Discussion

- Self-organizing map
  - ✓ 1D or 2D lattice map, Order out of disorder, Vector quantization
- Convergence considerations of SOM
  - ✓ "Almost sure" convergence
- Neurobiological considerations
  - ✓ SOM and cortical maps in the brain
  - ✓ Formation of computational maps in primary visual cortex
- Applications of SOM
  - ✓ Semantically related object groupings (classes)
  - ✓ Visual images, millions of documents
- Kernel SOM
  - ✓ Online, stochastic-gradient-based algorithm
  - ✓ Automatically adjust the kernel, but requires careful tuning of the learning rate parameters for the weights and width