PMR5406 Redes Neurais e Lógica Fuzzy

Aula 9 – Self-Organising Maps

Baseado em:

Neural Networks, Simon Haykin, Prentice-Hall, 2nd edition

Slides do curso por Marchiori

Unsupervised Learning

- Neural networks for unsupervised learning attempt to discover special patterns from available data without using external help (i.e. RISK FUNCTION).
  - **There is no information about the desired class (or output) d of an example x. So only x is given.**
  - **Self Organising Maps (SOM) are neural network models for unsupervised learning, which combine a competitive learning principle with a topological structuring of neurons such that adjacent neurons tend to have similar weight vectors.**

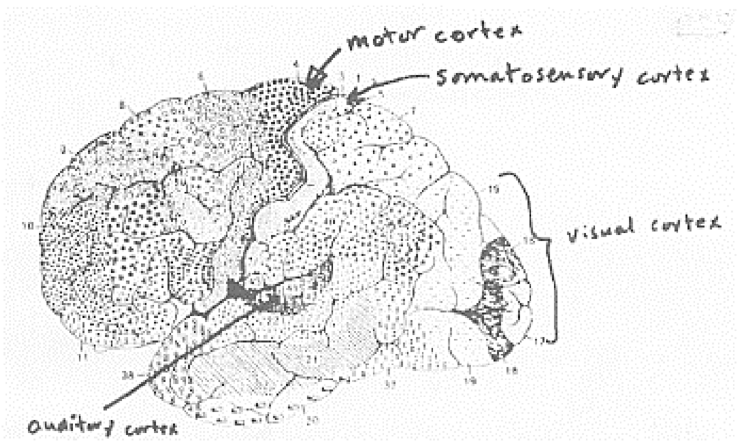
SOM: Biological Motivation

- Neurobiological hypothesis:
  - **The structure self-organises based on learning rules and system interaction.**
  - **Axons physically maintain neighborhood relationships as they grow.**

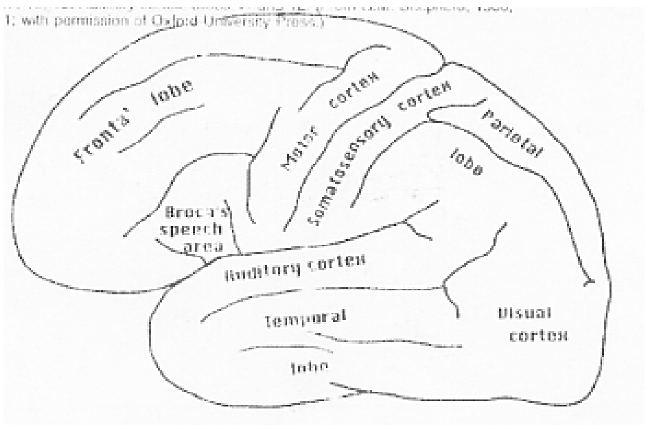
Topographic maps

- **Somatotopic map: projection of body surface onto a brain area, called somatosensory cortex, responsible for sense of touch.**
- **Motor map: Is similar for movement commands instead of touch.**
- **Retinotopic map: Is for vision. The area is called superior colliculus.**
- **Phonotopic map: Is for hearing: the auditory cortex.**

The cytoarchitectural map



The cytoarchitectural map



Two self-organised maps

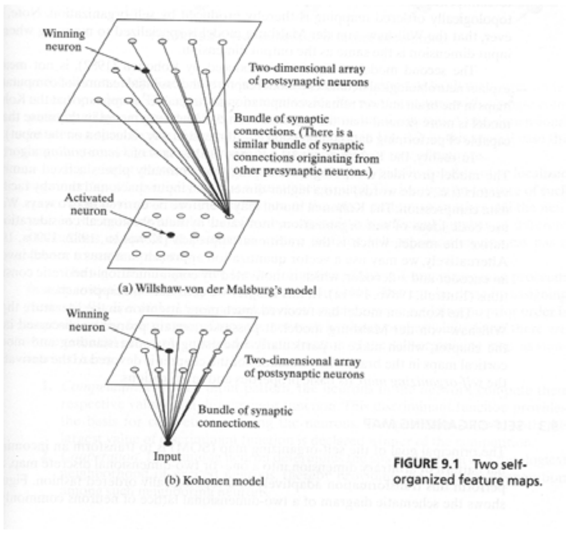
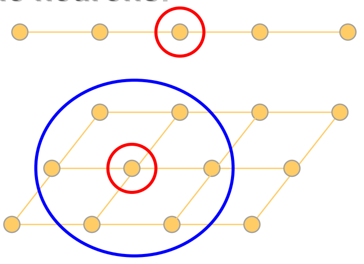

Winning neuron
Two-dimensional array of postsynaptic neurons
Bundle of synaptic connections. (There is a similar bundle of synaptic connections originating from other presynaptic neurons.)
Activated neuron
(a) Willshaw-von der Malsburg's model
Winning neuron
Two-dimensional array of postsynaptic neurons
Bundle of synaptic connections.
Input
(b) Kohonen model

FIGURE 9.1 Two self-organized feature maps.

ARCHITECTURE

- **The input is connected with each neuron of a lattice.**
- **Lattice Topology: It determines a neighbourhood structure of the neurons.**



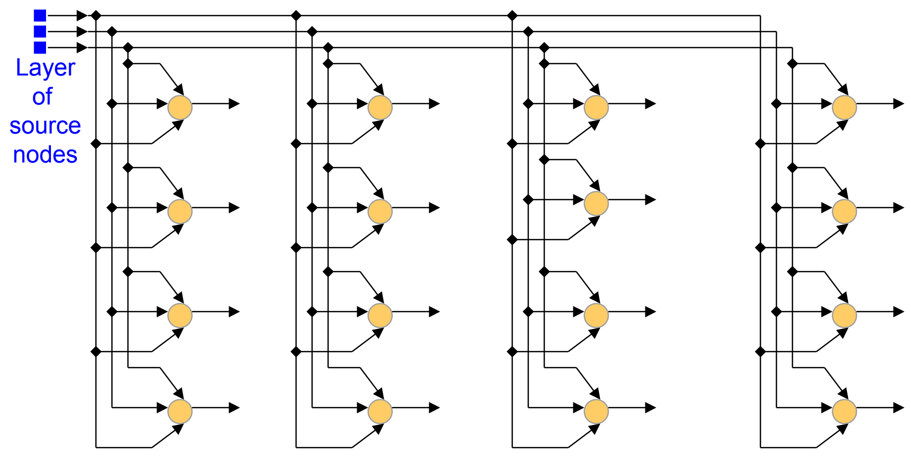
1-dimensional topology

A small neighbourhood

2-dimensional topology

Two possible neighbourhoods

Two-dimensional lattice



The goal

- We have to find values for the weight vectors of the links from the input layer to the nodes of the lattice, in such a way that adjacent neurons will have similar weight vectors.
- For an input, the output of the neural network will be the neuron whose weight vector is most similar (with respect to Euclidean distance) to that input.
- In this way, each (weight vector of a) neuron is the center of a cluster containing all the input examples which are mapped to that neuron.

The learning process (1)

An informal description:

- **Given: an input pattern *x***
- **Find: the neuron *i* which has closest weight vector by competition ($w_i^T x$ will be the highest).**
- **For each neuron *j* in the neighbourhood *N(i)* of the winning neuron *i*:**
  - **update the weight vector of *j*.**

The learning process (2)

- **Neurons which are not in the neighbourhood are left unchanged.**
- **The SOM algorithm:**
  - **Starts with large neighbourhood size and gradually reduces it.**
  - **Gradually reduces the learning rate η.**

The learning process (3)

- Upon repeated presentations of the training examples, the weight vectors tend to follow the distribution of the examples.

- This results in a topological ordering of the neurons, where neurons adjacent to each other tend to have similar weights.

The learning process (4)

- There are basically three essential processes:
  - competition
  - cooperation
  - weight adaption

The learning process (5)

- **Competition**:
  - **Competitive process: Find the best match of input vector *x* with weight vectors:**

$$\underset{\text{winning neuron}}{i(x)} = \arg\min_j \| x - w_j \| \qquad j = 1, 2, \ldots, \underset{\text{total number of neurons}}{\lambda}$$

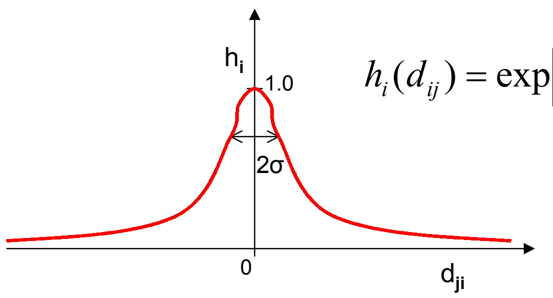
  - **The input space of patterns is mapped onto a discrete output space of neurons by a process of competition among the neurons of the network.**

The learning process (6)

- **Cooperation:**
  - **Cooperative process: The winning neuron locates the center of a topological neighbourhood of cooperating neurons.**
  - **The topological neighbourhood depends on lateral distance d_{ji} between the winner neuron *i* and neuron *j*.**

Learning Process (7) - neighbourhood function

– **Gaussian neighbourhood function**



$$h_i(d_{ij}) = \exp\left(-\frac{d_{ij}^{\,2}}{2\sigma^2}\right)$$

Learning process (8)

– **σ (effective width) measures degree to which excited neurons in the vicinity of the winning neuron participate to the learning process.**

exponential decay update

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

time constant

– **d_{ji}: lateral distance**

- **in one dimension lattice || *j – i* ||**
- **in two dimension lattice || r_j - r_i ||**
  r_j is the position of neuron *j* in the lattice.

Learning process (9)

- Applied to all neurons inside the neighbourhood of the winning neuron *i*.

$$\Delta w_j = \eta \underbrace{y_j x}_{\text{Hebbian term}} - \underbrace{g(y_j) w_j}_{\text{forgetting term}}$$

scalar function of response y_j

$$g(y_j) = \eta y_j$$
$$y_j = h_{i,j(x)}$$
$$w_j(n+1) = w_j(n) + \eta(n)\, h_{ij(x)}(n) \left(\mathrm{x} - w_j(n)\right)$$

exponential decay update: $\eta(n) = \eta_0 \exp\left(- n/T_2\right)$

Two phases of weight adaption

- Self organising or ordering phase:
  - Topological ordering of weight vectors.
  - May take 1000 or more iterations of SOM algorithm.
- Important choice of parameter values:
  - $\eta(n)$: $\eta_0 = 0.1$ $T_2 = 1000$
    $\Rightarrow$ decrease gradually $\eta(n) \geq 0.01$
  - $h_{ji(x)}(n)$: σ_0 big enough $T_1 = \dfrac{1000}{\log(\sigma_0)}$
  - Initially the neighbourhood of the winning neuron includes almost all neurons in the network, then it shrinks slowly with time.

Two phases of weight adaption

- Convergence phase:
  - Fine tune feature map.
  - Must be at least 500 times the number of neurons in the network $\Rightarrow$ thousands or tens of thousands of iterations.
- Choice of parameter values:
  - $\eta(n)$ maintained on the order of 0.01.
  - $h_{ji(x)}(n)$ contains only the nearest neighbours of the winning neuron. It eventually reduces to one or zero neighbouring neurons.

A summary of SOM

- **Initialization:** choose random small values for weight vectors such that $w_j(0)$ is different for all neurons j.
- **Sampling:** drawn a sample example x from the input space.
- **Similarity matching:** find the best matching winning neuron $i(x)$ at step n:

$$i(x) = \arg\min_j \| x(n) - w_j \| \qquad j \in [1, 2, \dots, \lambda]$$

- **Updating:** adjust synaptic weight vectors

$$w_j(n+1) = w_j(n) + \eta(n)\, h_{ij(x)}(n)\, (\mathrm{x} - w_j(n))$$

- **Continuation:** go to Sampling step until no noticeable changes in the feature map are observed.

Example 1

A 2-dimensional lattice driven by a 2-dimensional distribution:

- **100 neurons arranged in a 2D lattice of 10 x 10 nodes.**
- **Input is bidimensional: *$x = (x_1, x_2)$* from a uniform distribution in a region defined by:**
  $\{ (-1 < x_1 < +1); (-1 < x_2 < +1) \}$
- **Weights are initialised with *random* values.**

Visualisation

- Neurons are visualised as changing positions in the *weight space* (which has the same dimension of the input space) as training takes place.

Example 1: results

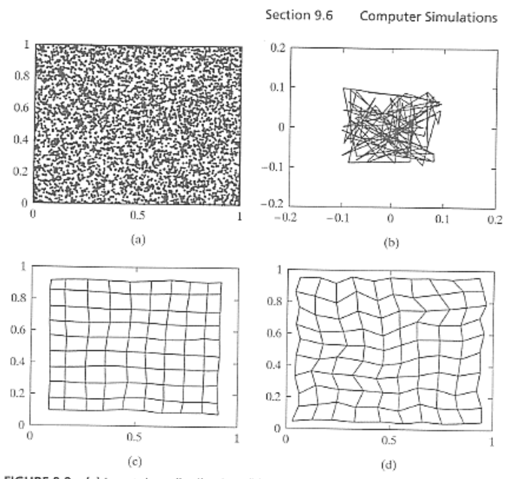


FIGURE 9.8 (a) Input data distribution. (b) Initial condition of the two-dimensional lattice. (c) Condition of the lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase.

Example 2

A one dimensional lattice driven by a two dimensional distribution:

- **100 neurons arranged in one dimensional lattice.**
- **Input space is the same as in Example 1.**
- **Weights are initialised with *random* values (again like in example 1).**
- **(Matlab programs for Examples 1, 2 available at ftp://ftp.mathworks.com/pub/books/haykin)**

Example 2: results

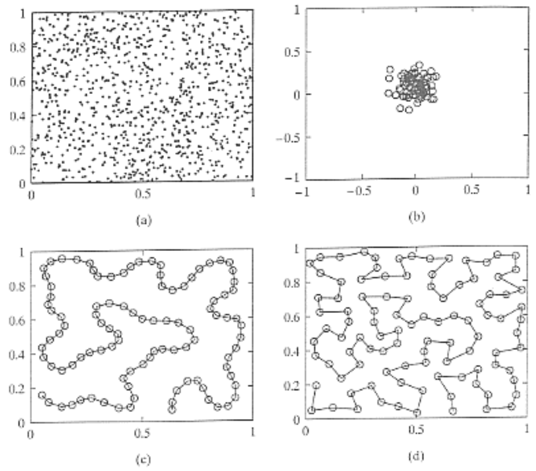


FIGURE 9.9 (a) Two-dimensional input data distribution. (b) Initial condition of the one-dimensional lattice. (c) Condition of the lattice at the end of the ordering phase. (d) Condition of the lattice at the end of the convergence phase.

PMR5406 Redes Neurais e Lógica Fuzzy SOM 27

Example 2: parameter evolution

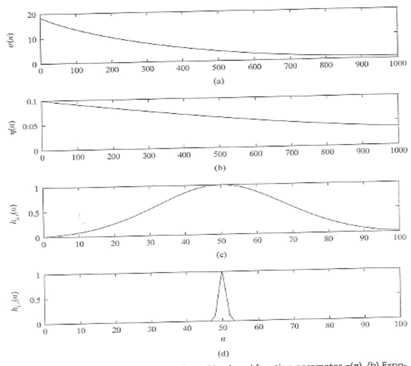


FIGURE 9.10 (a) Exponential decay of neighborhood function parameter $\sigma(n)$. (b) Exponential decay of learning-rate parameter $\eta(n)$. (c) Initial shape of the Gaussian neighborhood function. (d) Shape of the neighborhood function at the end of the ordering phase (i.e., beginning of the convergence phase).

in Fig. 9.10a, starts with an initial value $\sigma_0 = 18$ and then shrinks to about 1 in 1000 iterations during the ordering phase. During that same phase, the learning-rate parameter $\eta(n)$ starts with an initial value $\eta_0 = 0.1$ and then decreases to 0.037. Figure 9.10c shows the initial Gaussian distribution of neurons around a winning neuron located at the midpoint of the one-dimensional lattice. Figure 9.10d shows the shape of the neighborhood function at the end of the ordering phase. During the convergence phase the learning-rate parameter decreases linearly from 0.037 to 0.001 in 5000 iterations. During the same phase the neighborhood function decreases essentially to zero.

The specifications of the ordering phase and convergence phase for the computer simulations in Fig. 9.8 involving the two-dimensional lattice are similar to those used

PMR5406 Redes Neurais e Lógica Fuzzy SOM 28

Learning Vector Quantisation (1)

- Vector quantisation is a technique designed for data compression.
- A vector quantiser with minimum encoding distortion is a called a Voronoi or neares-neighbour quantiser.

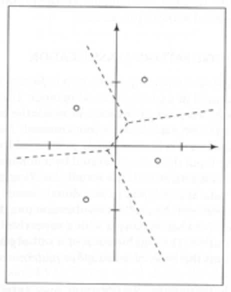


FIGURE 9.12 Voronoi diagram involving four cells. (Adapted from R.M Gray, 1984, with permission of IEEE.)

Learning Vector Quantisation (2)

- Learning Vector Quantisation (LVQ) is a supervised learning technique that move the Voronoi vectors slightly. So as to improve the quality of the classifier decision regions.
- It can be devided into two parts: a competitive layer and a vector quantisation layer:

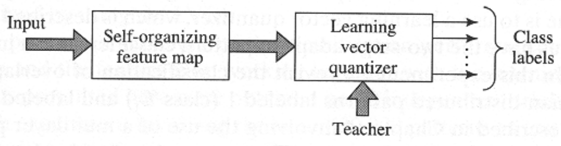


FIGURE 9.13 Block diagram of adaptive pattern classification, using a self-organizing feature map and learning vector quantizer.

Learning Vector Quantisation (3)

- The algorithm:
  - Let $\{w_j\}_{j=1}^{l}$ denote the Voronoi vectors and let $\{x_i\}_{i=1}^{N}$ the input vectors.
  - C_{wc} is the class associated with w_c. C_{wi} is the class associated with w_i.

1. If $C_{wc} = C_{wi}$ then:
   $W_c(n+1) = w_c(n) + \alpha_n [x_i - w_c(n)]$
2. If $C_{wc} <> C_{wi}$ then:
   $W_c(n+1) = w_c(n) - \alpha_n [x_i - w_c(n)]$
3. The other Voronoi vectors are not modified.

Learning Vector Quantisation (4)

- **Um exemplo:**

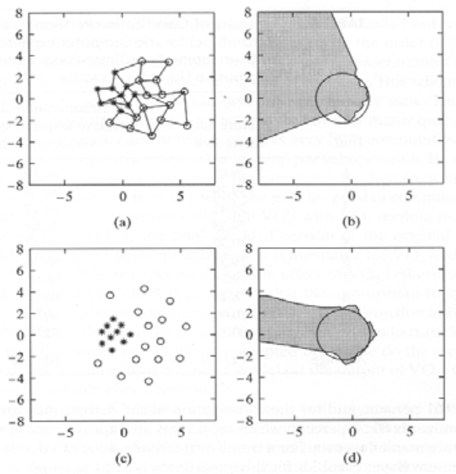
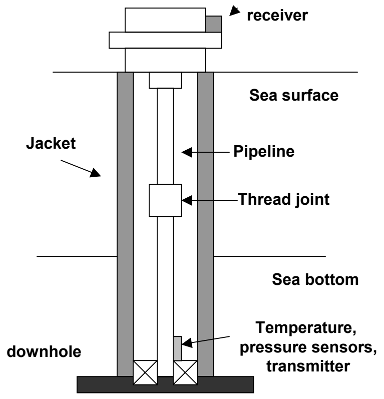


FIGURE 9.14 (a) Self-organizing map after labeling. (b) Decision boundary constructed by the feature map of part a. (c) Labeled map after learning-vector quantization. (d) Decision boundary constructed by the feature map of part c.

Acoustic Transmission System for Oil-Well Monitoring

- Periodic measurement of temperature and pressure at the downhole.
- Column length used in deep-sea exploration might be ordinarily longer than 3,000m.
- With the elimination of cabling significant cost savings and increased reliability can be attained.



- Some important issues:
  - Multiple reflections take place at the pipeline junctions,
  - Propagation through thread joints introduces nonlinear characteristics,
  - The oil flow produces vibration therefore being a source of acoustic noise.
- The chosen solution:
  - Frequency-Shift Keying (FSK) modulation is chosen, I.e., a frequency f1 is chosen for a bit “1” and a frequency f2 is chosen for a bit “0”
  - Demodulation process using a Linear Vector Quantization based Neural Network

- **u(k)=f(k)**
- **v(k)=f(k+l)**
- **A histogram matrix H(u,v) is designed.**
- **A geometric series generator was used to compress histogram peaks and reinforce other points of the image:**
- **$Z(u,v)=(1-0.5^{H(u,v)})/(1-0.5)$.**

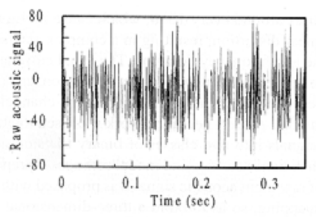


Fig. 10. Acoustic transmission signal.

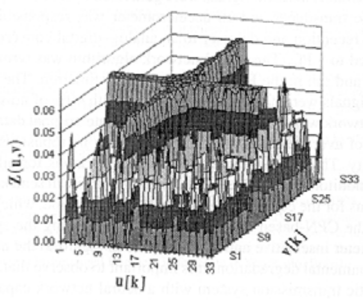
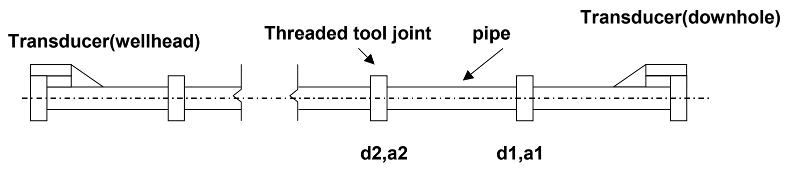


Fig. 11. Image signal pattern after the discrete mapping operation.

How FSK frequencies are selected ?

- The acoustic waves propagating on such mechanical structure has phase and group velocities depending on frequency,
- Some frequencies are blocked for propagation within certain periodic bands,
- Therefore, the acoustic waves propagates at the expense of high distortion of phase and group velocities in terms of frequency,



- The relationship between the angular frequency ω and the wave number k is given by:

$$\cos k(d_1+d_2)=\cos\left(\frac{\omega d_1}{c_1}\right)\cos\left(\frac{\omega d_2}{c_2}\right)-\frac{1}{2}\left(\frac{a_1}{a_2}+\frac{a_2}{a_1}\right)\sin\left(\frac{\omega d_1}{c_c}\right)\sin\left(\frac{\omega d_2}{c_2}\right)$$

$c_1 = c_2$	**Extensional wave velocity in steel**
$c_f = \dfrac{\omega}{k}$	**Phase velocity**
$c_g = \dfrac{d\omega}{dk}$	**Group velocity**