Clustering 2
Self-Organizing Maps (SOMs)

Sebastian Risi
IT University of Copenhagen

Some slides adapted from Data Mining Course CSE5230 and "Lecture on SOMs" by Bruno Silva

Individual Assignment

- At least two different preprocessing methods, one sequential/frequent pattern mining method, one clustering method and one supervised learning method
- E.g. normalization+missing value replacement+Apriori+kmeans+ID3
- Instead of apriori ➔Anything from Pattern Mining II (e.g. GSP, FSG, …)

Group Projects : Proposals Due Next Week

- Organise yourselves into groups of 3-4 persons
- Define and conduct your own data mining project using data and tools of your choice
- Use at least two of the following: association mining, classification/prediction, clustering, sequence mining
- Plus preprocessing and evaluation
- Write a good report!

Quick Repetition

- How does the Perceptron Learning algorithm work?
- What problem can a single-layer perceptron not learn?
- Main idea behind backprop
- What does supervised learning mean?
- Why does the activation function need to be differentiable for backprop?
- Example of a differentiable function

Motivation

- The ANNs discussed last week are an example of a **supervised learning** technique
- In supervised learning, the aim is to discover a relationship between the **inputs** and **outputs** of a system
- This relationship can be used for tasks such as prediction, estimation or classification
- A known **training set** of input/output pairs is used to **train** the network

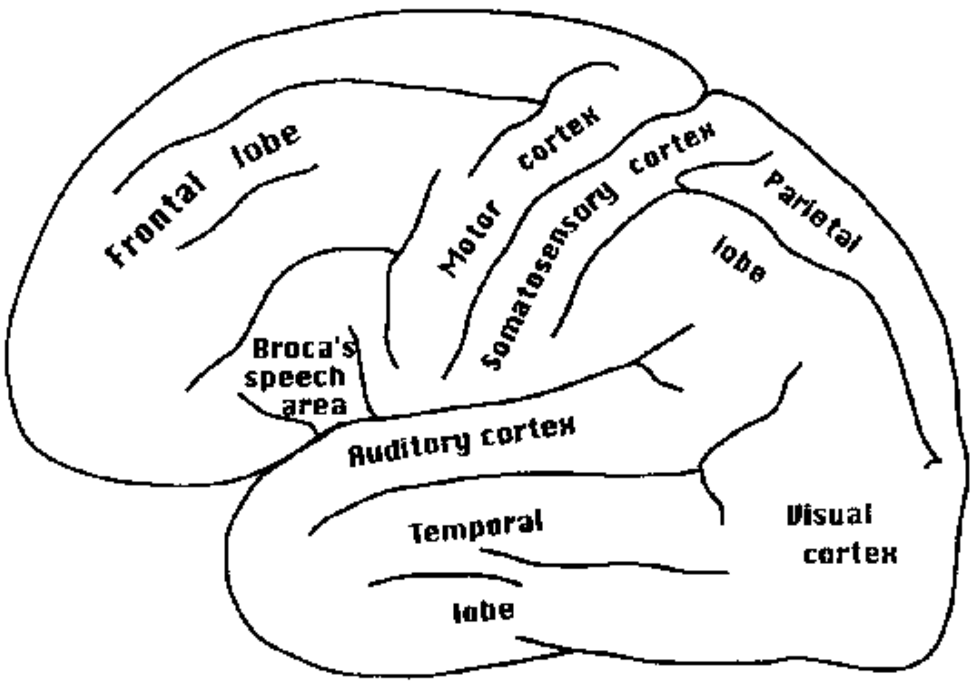
Motivation

- Many data mining tasks are not suited to this approach
- Often the data mining task is to **discover structure** in the data set, without any prior knowledge of what is there
- This is an example of **unsupervised learning** (we have already seen the example of the K-means clustering algorithm)
- A class of neural networks called **Self-Organizing Maps** (SOMs) can be used for this task

The Cortex

- SOMs research was inspired by the observation of topologically correct sensory maps in the cortex (e.g. the retinotopic, somatotopic, tonotopic maps)
- In humans, the cortex consists of a thin layer of nerve tissue
- It is highly convoluted to save space, and forms the exterior of the brain - it’s the folded, wrinkled stuff we see when we look at a brain

The Cortex



Lateral (schematic) view of the human left-brain hemisphere. Various cortical areas devoted to specialized tasks can be distinguished [RMS1992, p. 18]

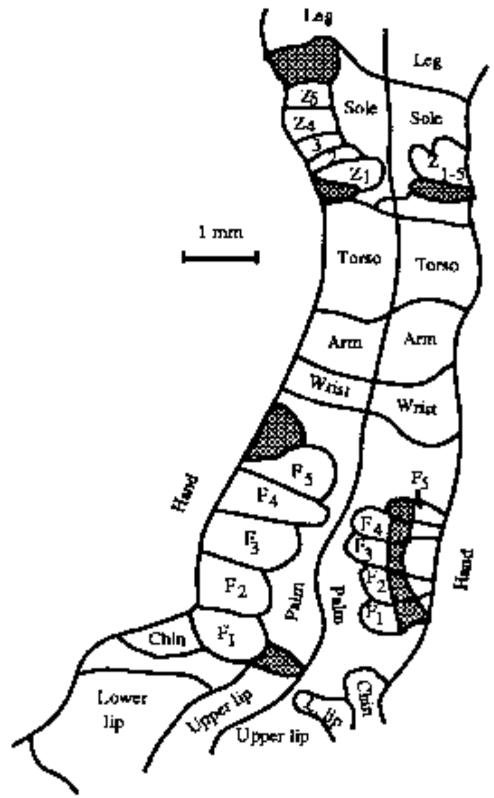
Sensory Surfaces

- Most signals that the brain receives from the environment come from “sensory surfaces” covered with receptors:
  - skin (touch and temperature)
  - retina (vision)
  - cochlea [in the ear] (1-D sound sensor)
- It is usually found that the “wiring” of the nervous system exhibits **topographic ordering**:
  - signals from adjacent receptors tend to be conducted to adjacent neurons in the cortex

Topographic Feature Maps

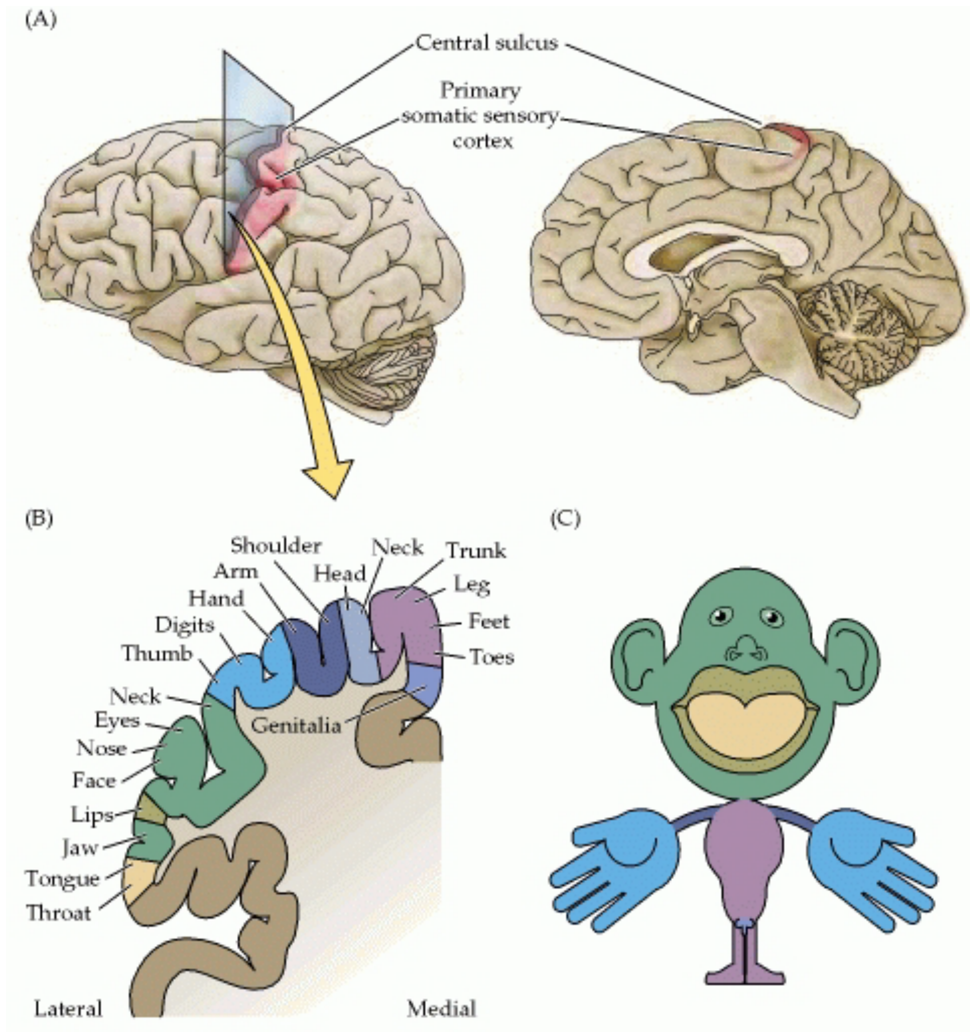
- This neighbourhood-preserving organization of the cortex is called a **topographic feature map**
  - For touch, maps of the body are found in the somatosensory cortex
  - In the primary visual cortex, neighbouring neurons tend to respond to stimulation of neighbouring regions of the retina
- As well as these simple maps, the brain also constructs topographic maps of **abstract features**:
  - In the auditory cortex of many higher brains, a tonotopic map is found, where the pitch of received sounds is mapped regularly

Topographic Feature Maps



Map of part of the body surface in the somatosensory cortex of a monkey

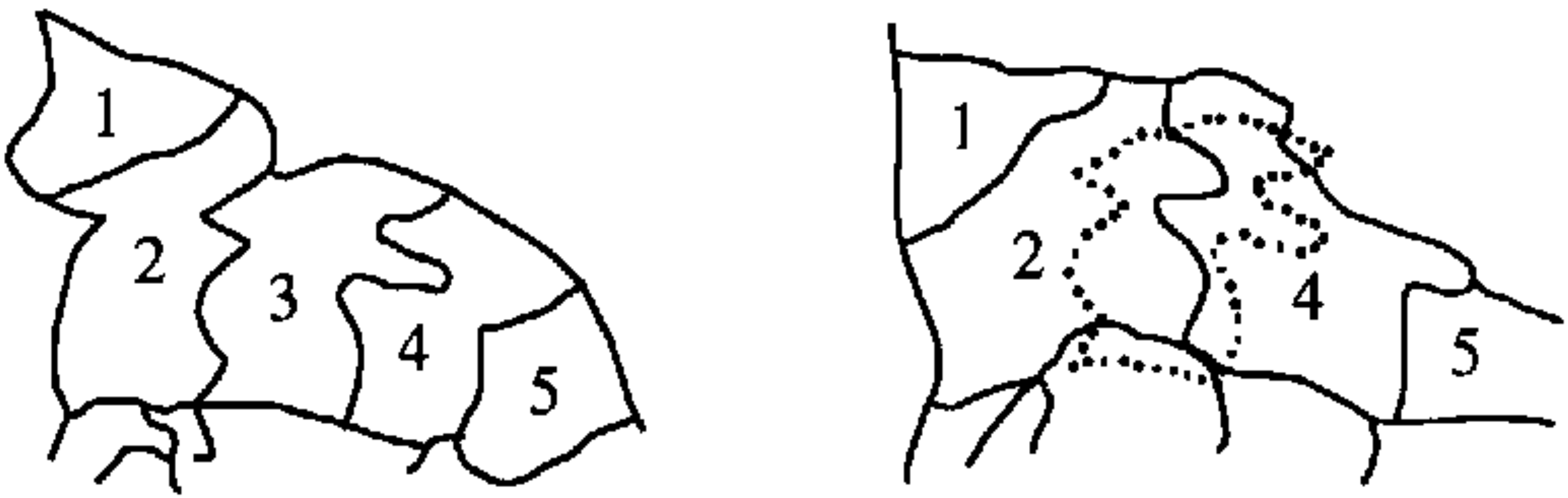
Maps in the brain



Biological Self-Organizing Maps

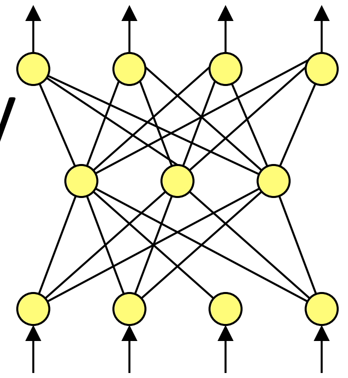
- How can such topology-preserving mappings arise in neural networks?
- In nature they form in **interaction with the environment**
  - The normal development of edge-detectors in the visual cortex of newborn kittens is suppressed in the absence of sufficient visual experience
  - The somatosensory maps of adult monkeys have been observed to adapt following the amputation of a finger

Biological Self-Organizing Maps



Readaptation of the somatosensory map of the hand region of an adult nocturnal ape due to the amputation of one finger. Several weeks after the amputation of the middle finger (3), the assigned region has disappeared and the adjacent regions have spread out. [RMS, p. 117]

Artificial Self-Organizing Maps



- In the NN models we have seen so far, every neuron in a layer is connected to every neuron in the next layer of the network
- The **location** of a neuron in a layer plays no role in determining its connectivity or weights
- With SOMs, the ordering of neurons within a layer plays an important role:
  - How should the neurons organize their connectivity to optimize the **spatial distribution of their responses within the layer**?

Artificial Self-Organizing Maps

- The purpose of this optimization is to achieve the mapping:

  Similarity of features $\longrightarrow$ **Proximity of excited neurons**

- It results in the formation of **topographic feature maps**:
  - **Similarity relationships** among the input signals are converted into **spatial relationships** between responding neurons

Self-Organizing Map

- This model proposed by Kohonen in 1982 captures the essential features of computational maps on the brain in a very simple, yet powerful, algorithm.
  - Sometimes also referred as Kohonen's Feature Map.

- The SOM is a very popular ANN algorithm based on competitive and unsupervised learning.

- Extensive research has been made with applications ranging from full text and financial analysis, pattern recognition, image analysis and process monitoring.

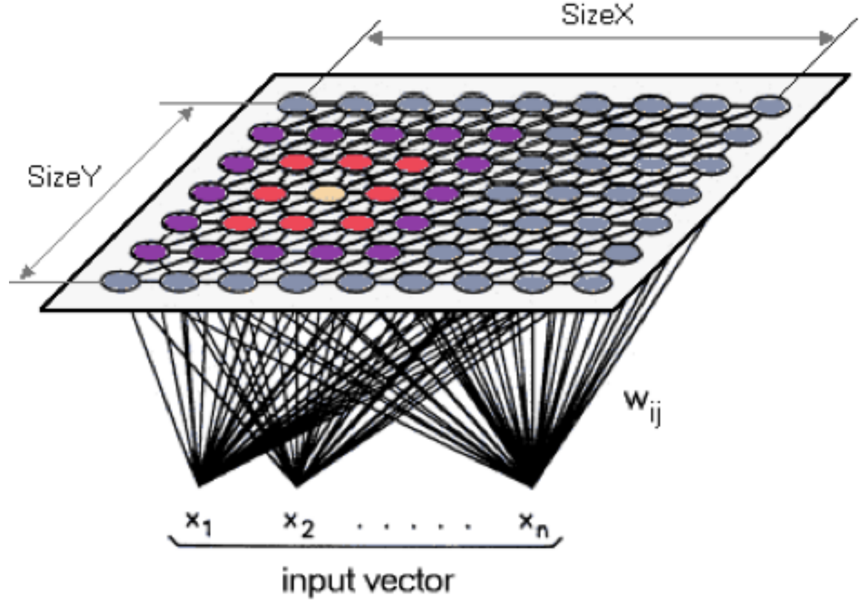
Self-Organizing Map

- The SOM is able to project high-dimensional data in a lower dimension, typically 2D, while preserving the relationships among the input data.

- The 2D pattern map is useful in
  - Analyzing and discovering patterns in the input space;
  - Detecting non-linear correlations between features;
  - Identifying clusters;
  - Classifying new observations.

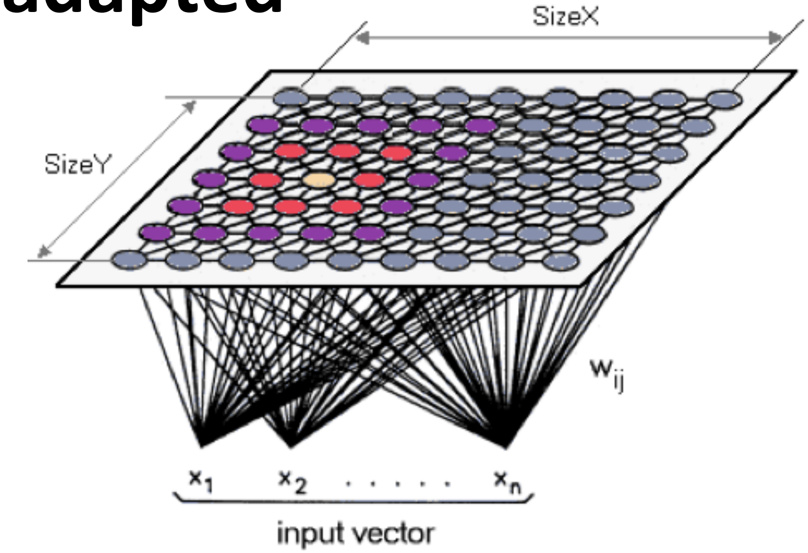
SOM Network Architecture

- Consists of two layers (feed-forward)
  - Input and output (competitive) layer
- K neurons arranged in a 2D lattice
- Neighboring relations between neuron
- Each neuron contains a weight (prototype) vector



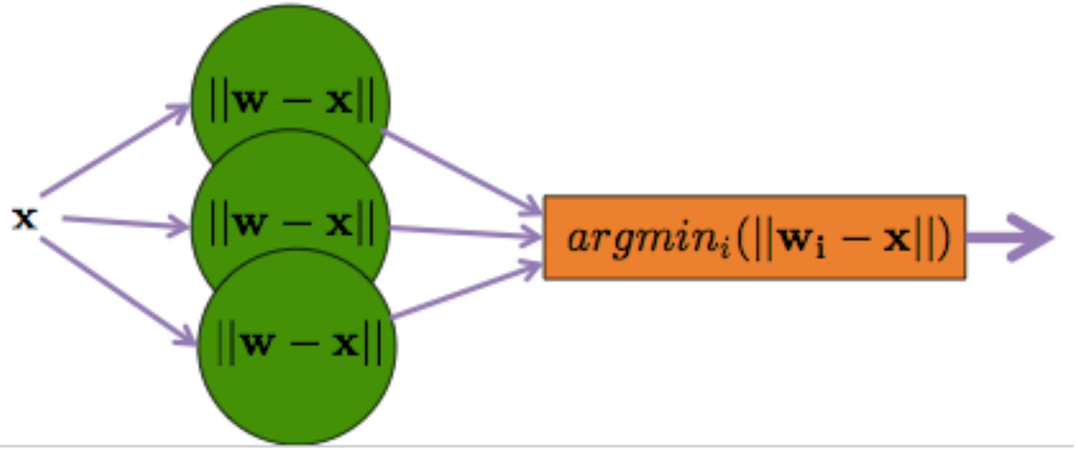
Competitive Learning Network

- Concept:
- A data point is presented to the network
- All neurons process the input
- Only one or few neurons (**competition**) are activated and weights are **adapted**



Neurons competing on SOM

- Each neuron calculates the Euclidean distance between its weight and the input
- The neuron with the lowest distance wins
- This is the best match



SOM – Learning Algorithm

- Start: Initialize the weights with random values
- Repeat for n epochs
  - For each data point
    - Find its best-match
    - Update winner so that it becomes more like x, together with the winner's neighbours for units within the radius according to an **update function**
- Update function: Kohonen's update rule
  - The weight update depends on:
    - Distance between the neuron and the best- match (distance in the map)
    - Difference between the data point and the weight (distance in the input space)
    - Epoch

SOM – Learning Algorithm

- Find its best-match
  - Compare x with weights wj for each neuron j

$$d_j = \sum_i (w_{ij} - x_i)^2$$

  - find unit j with the minimum distance

- Update function

$$w_j(t+1) = w_j(t) + \eta(t).h_{ij(x)}(t)\ \left(\mathrm{x}\text{-}w_j(t)\right)$$

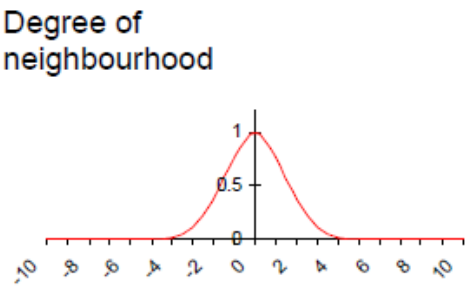
$$0 < \eta(t).h_{ij(x)}(t) \le 1$$

Neighborhood Function

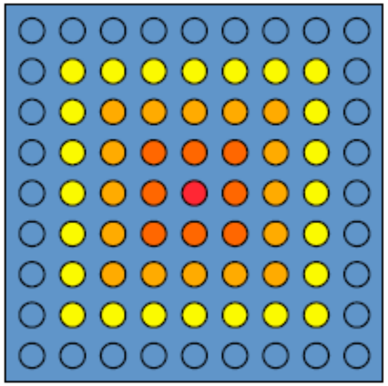
- Gaussian neighborhood function:

$$h_i(d_{ij}) = \exp\left(- d_{ij}^2 \Big/ 2radius^2 \right)$$

- d_{ji}: lateral distance of neurons i and j
  - in a 1-dimensional lattice $| j - i |$
  - in a 2-dimensional lattice $|| r_j - r_i ||$ where r_j is the position of neuron j in the lattice.



Distance from winner



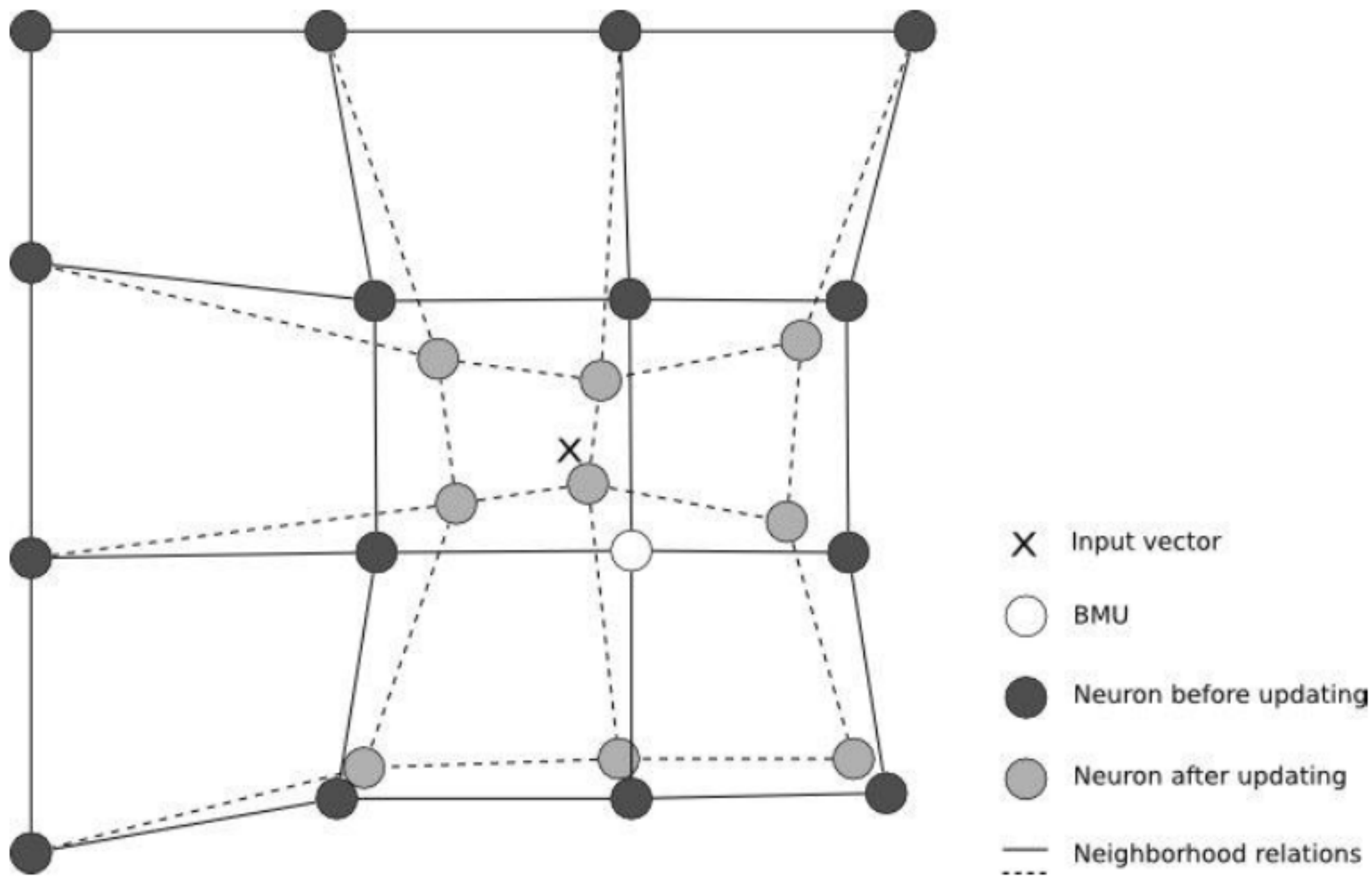
Learning rate

Could be linear, exponential,...

E.g. Exponential decay update of the learning rate:

$$\eta(n) = \eta_0 \exp\left(-{}^{n}\!/_{T}\right)$$

The Self-Organizing Map (SOM) Adaptation Process Illustrated



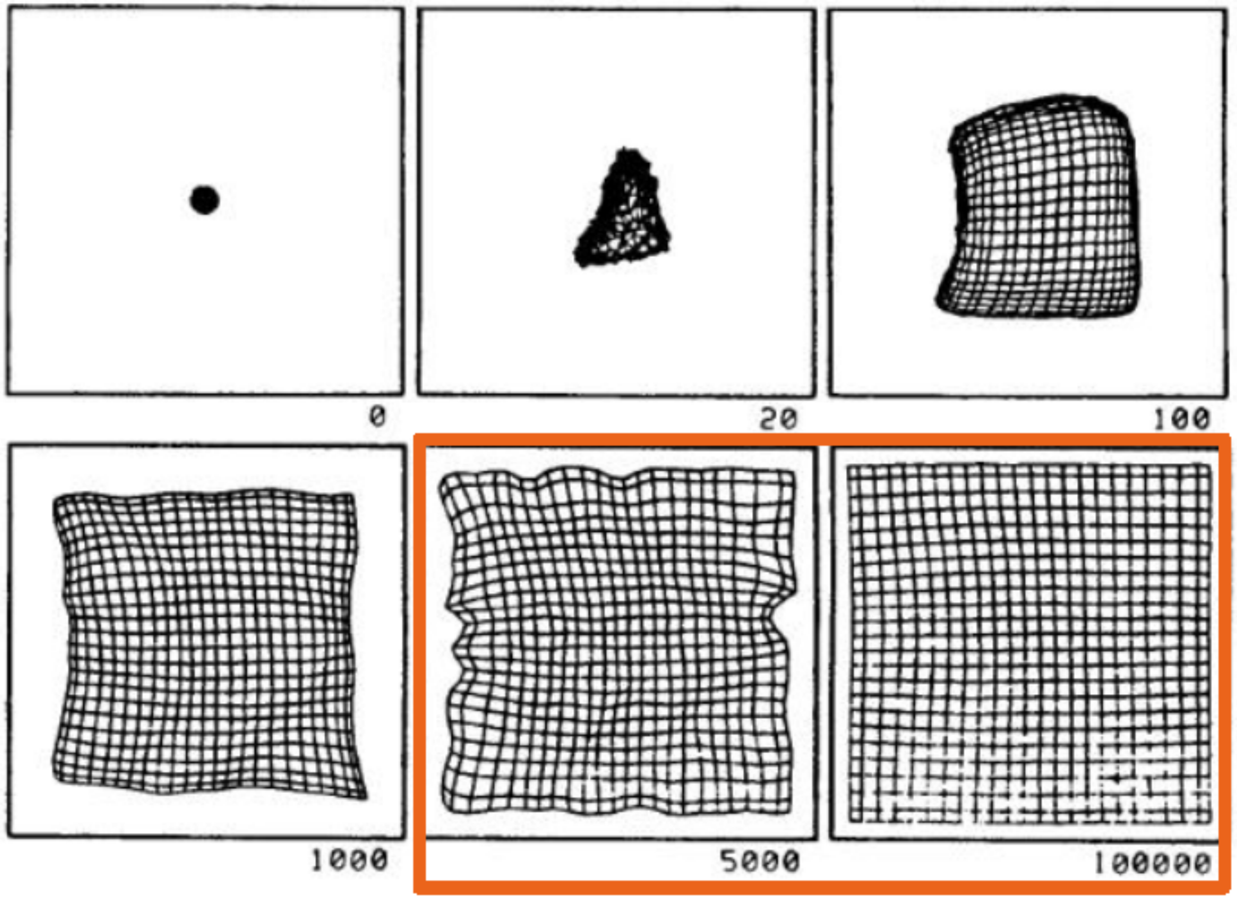
The Self-Organizing Map (SOM) Ordering and Convergence

- **Ordering (rough) phase**
  - It is during this first phase of the adaptation process that the topological ordering of the weight vectors takes place. This ordering phase is relatively *short* in comparison to the second phase. Large values for the neighborhood radius and learning rate should be used, such that the neuron's weights initially take large steps all together toward the area of input space where input vectors are occurring. These values then should decrease to their tuning values and, consequently, the neighborhood decreases to encompass only the closest neighbors.

The Self-Organizing Map (SOM) Ordering and Convergence

- **Convergence (fine-tuning) phase**
  - This phase lasts for the rest of the training or adaptation process and is necessary to fine tune the network and therefore provide an accurate statistical quantification of the input space. During this phase the weight vectors converge to their *correct* values. For this, the neighborhood should be fairly small, encompassing only the immediate neighbors. This also applies to the learning rate, such that the magnitude of the weight updates is very small. The convergence phase is usually several times longer than the ordering phase.

The Self-Organizing Map (SOM)
Ordering and Convergence



Fine-tuning Phase

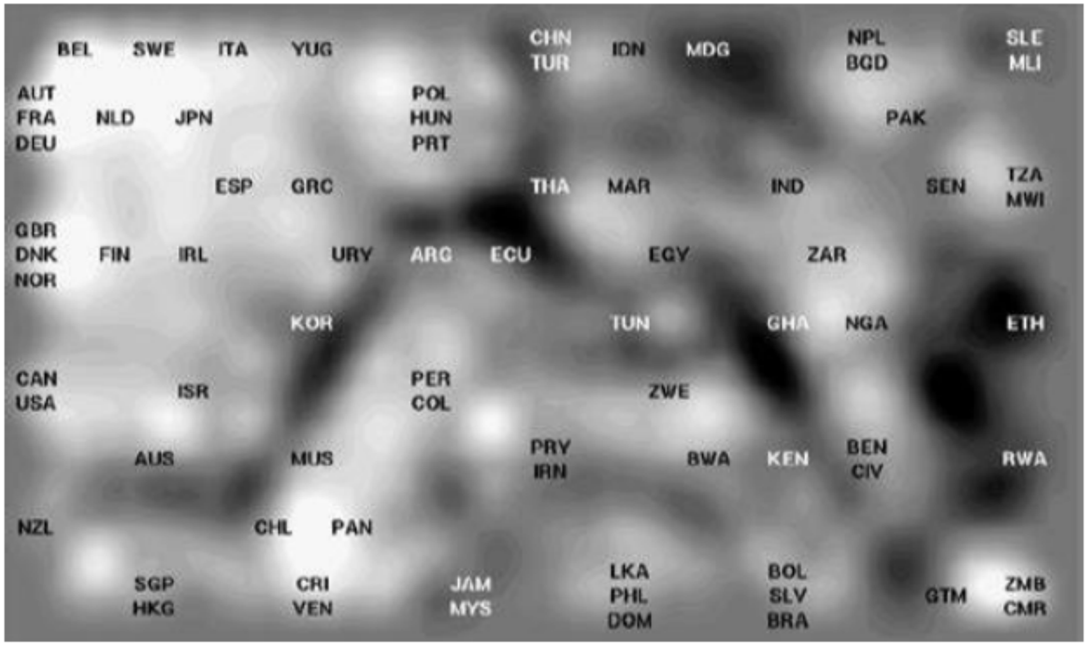
The Self-Organizing Map (SOM) Important Aspects

- Input vectors which are neighbors in the input space are mapped onto neighboring neurons.

- If the dimensionality of the input space and the network differ, only the most important similarity relationships are preserved and mapped onto neighborhood relations on the network
  - The less important ones are not retained.

- The SOM is very robust against “noise” in the input data, because the SOM also maps the input distribution density!

The Self-Organizing Map (SOM) Example – World Poverty Map (1992)



Information about welfare and poverty-related aspects of 77 countries are treated by the SOM algorithm. The map is visualized with a special visualization technique, which we'll discuss later in this lecture.

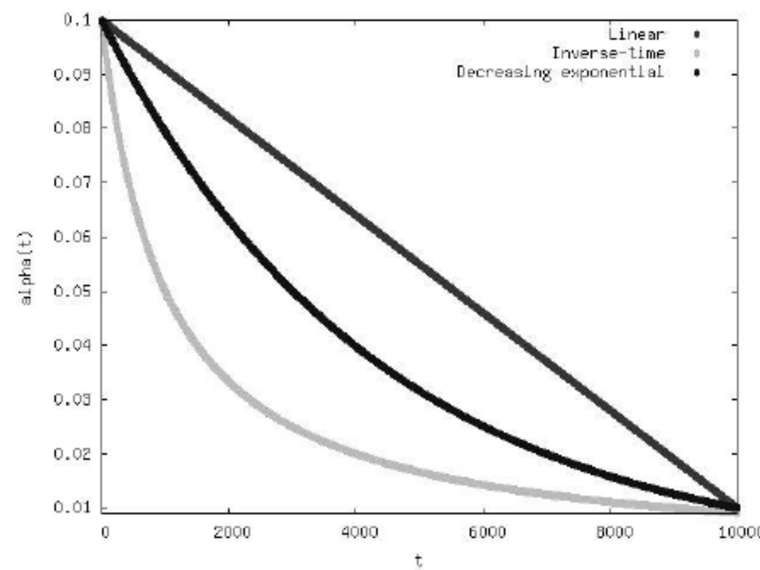
The Self-Organizing Map (SOM) Parameterization

- Appropriate selection of the initial parameters is of great importance to obtain good maps.
- Unfortunately, parameterizing the SOM is an "art" :-(
  - But there are some "rules-of-thumb".
- We'll talk briefly about:
  - The learning rate;
  - The width of the neighborhood kernel;
  - The map's shape;
  - The map's size.

- The number of epochs usually depends on the size of the training set. Small training sets may require more epochs.

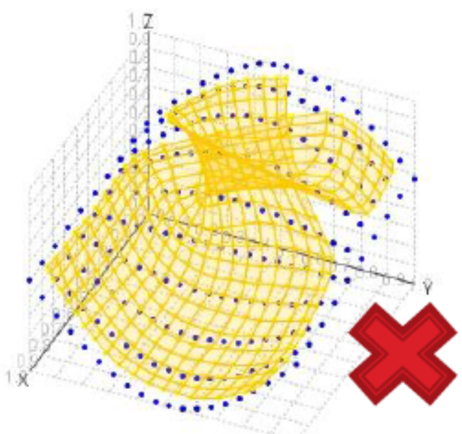
The Self-Organizing Map (SOM) Parameterization – α and σ

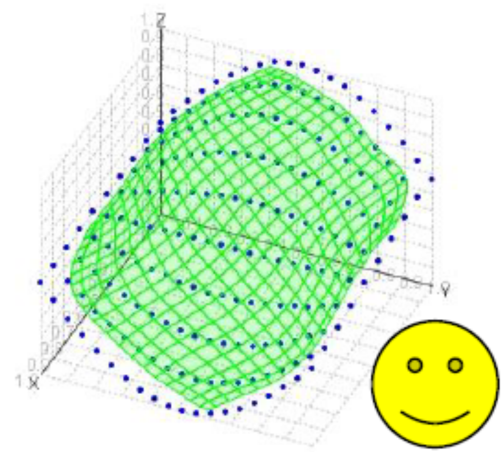
- They should be ruled by a function that decreases monotonically over time.
  - Its exact form is not critical.
- The learning rate is hard to estimate and is "problem-specific". General values can be $0.1 > \alpha > 0.01$. Although higher values may avoid local minima.
- The neighborhood radius should start with a value that spans the whole map down to the immediate neighbors.



The Self-Organizing Map (SOM) Parameterization – α and σ

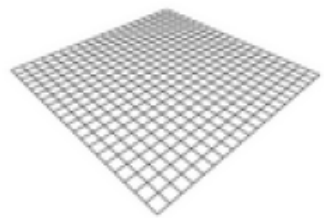
- The properly initialization of the neighborhood radius has a strong impact in the final topology of the map.
  - If one chooses a small initial value, a distorted map can be obtained – which is not desirable!



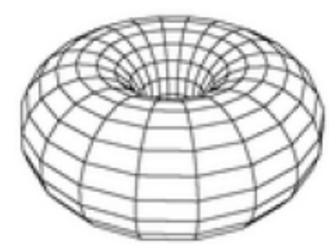


The Self-Organizing Map (SOM) Parameterization – Shape

- The SOM shape determines the neurons' neighboring relations.
  - Usually the sheet-like shape is the most common and straightforward:



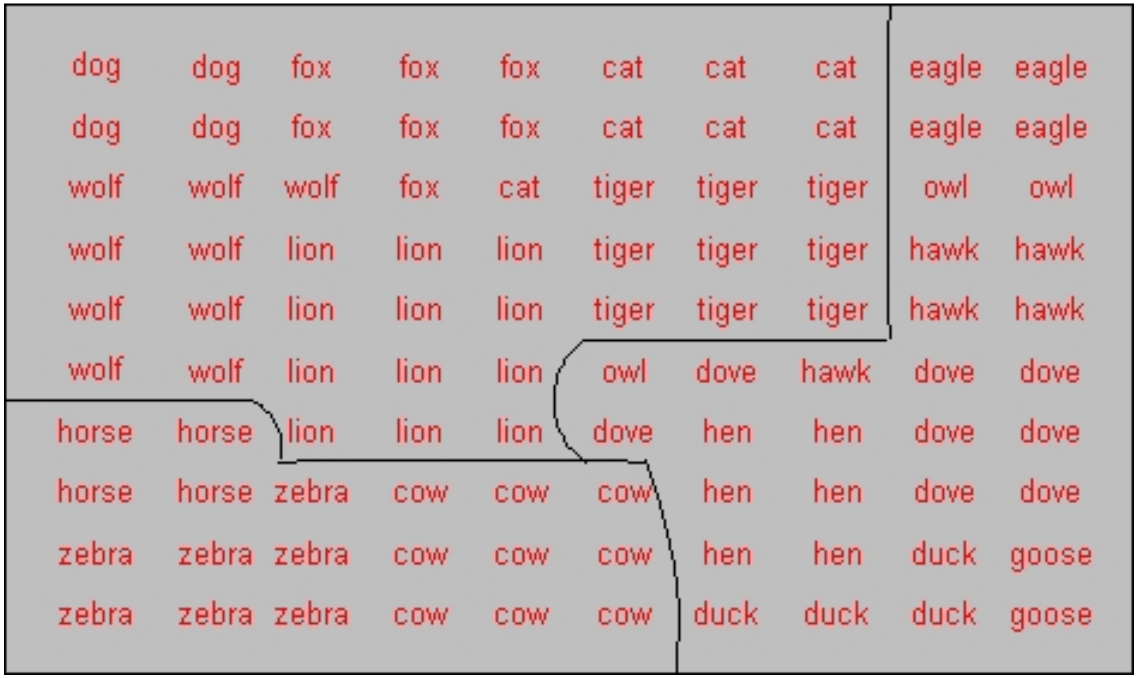
  - The toroid topology avoids "border effects" by connecting the top of the map to the bottom and the left to the right:



The Self-Organizing Map (SOM) Parameterization – Size

- The map size determines the degree of precision and the ability to quantize the input space.
  - If one chooses the number of neurons to be equal to the desired number of clusters, then the SOM behaves like a very robust k-means algorithm.
  - Using large maps allows for the emergence of salient features in the input space. With the use of visualizations techniques one can obtain insight on the number of clusters in the data and correlations between features in the training set.

Another Example



We observe that there are three distinct clusters of animals: “birds”, “peaceful species” and “hunters”.

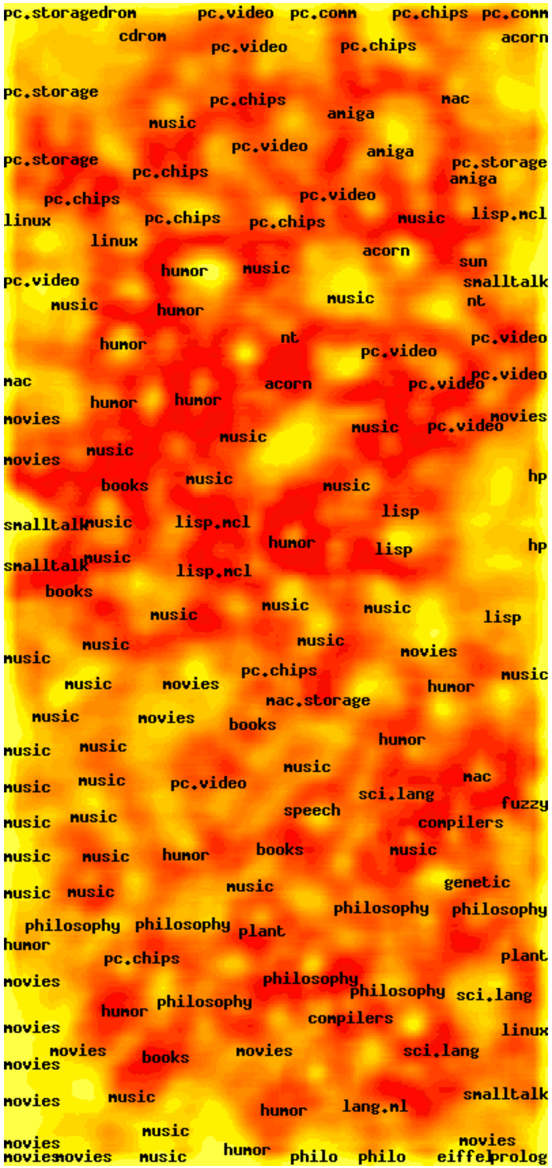
The SOM for Data Mining

- The SOM is a good method for obtaining an initial understanding of a set of data about which the analyst does not have any opinion (e.g. no need to estimate number of clusters)
- The map can be used as an initial unbiased starting point for further analysis. Once the clusters are selected from the map, they are analyzed to find out the reasons for such clustering
  - It may be possible to determine which attributes were responsible for the clusters
  - It may also be possible to identify some attributes which do not contribute to the clustering

Example: Text Mining with a SOM

- This example comes from the WEBSOM project in Finland: http://websom.hut.fi/websom/
- WEBSOM is a method for organizing miscellaneous text documents onto meaningful maps for exploration and search. WEBSOM automatically organizes the documents onto a two-dimensional grid so that **related documents appear close to each other**

Example: Text Mining with a SOM

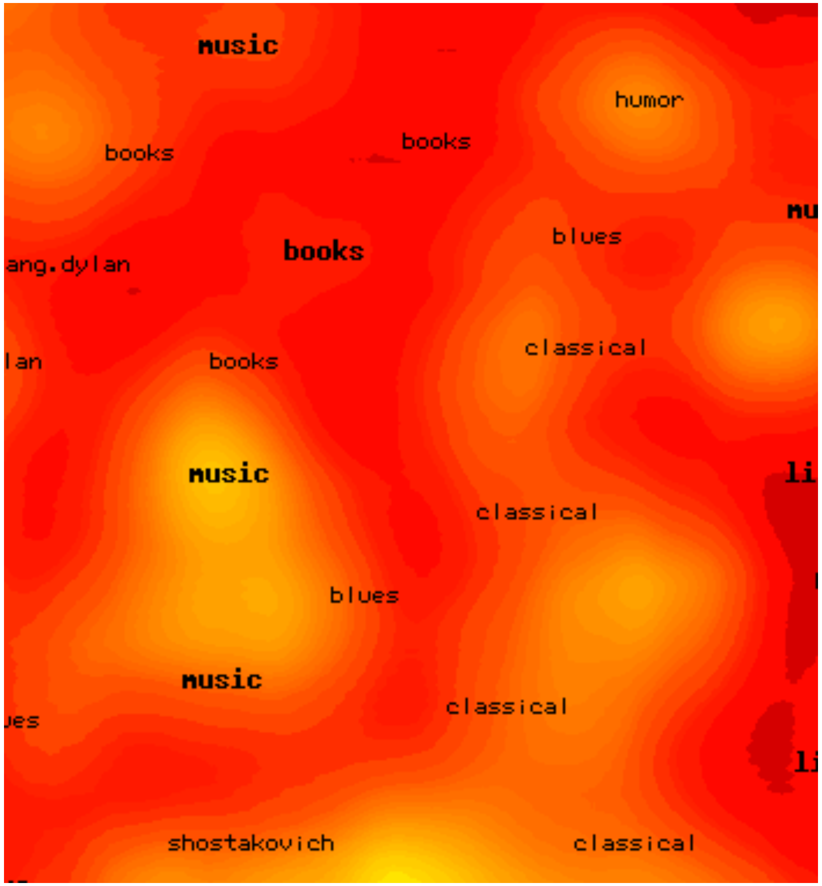


This map was constructed using more than one million documents from 83 USENET newsgroups:

- Color denotes the density or the clustering tendency of the documents
- Light (yellow) areas are clusters and dark (red) areas empty space between the clusters
- This is a little difficult to read, but WEBSOM allows one to zoom in

Example: Text Mining with a SOM

- Zoomed view of the WEBSOM map:



Example: Image Clustering

http://www.codeproject.com/KB/graphics/som

The Self-Organizing Map (SOM) Visualization Techniques

- One of the greatest advantages of the SOM is that it is highly visual.
- There are two major visualizations for the SOM:
  - **Unified Distance Matrix**

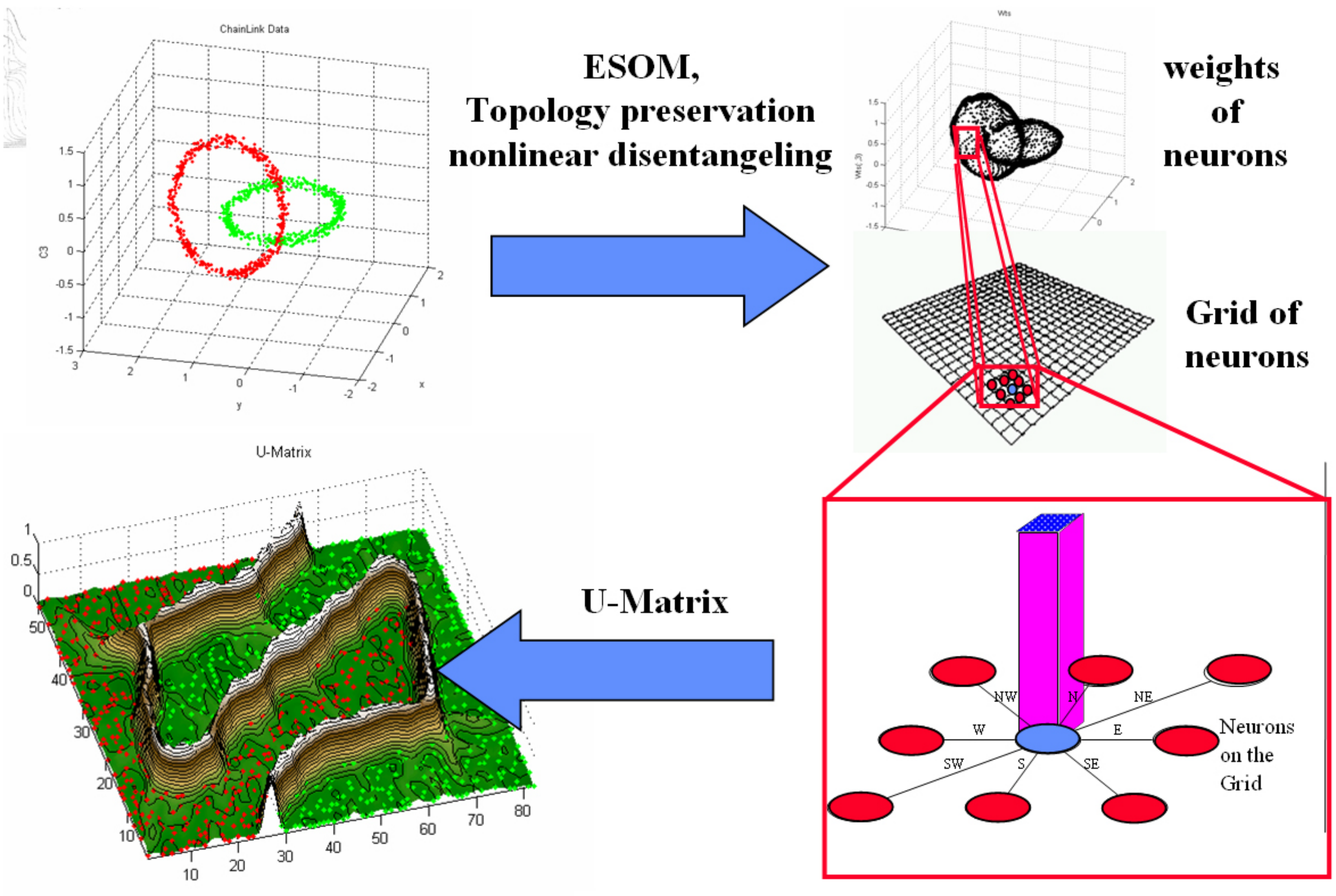
    or simply U-Matrix, visualizes the cluster structure of the SOM. A matrix of distances between the weight vectors of adjacent neurons on the map is formed, after which some representation for the matrix is chosen, for example a color scale. The lighter the color between two map units is, the smaller is the relative distance between their weight vectors. Given this, dark areas on the maps usually identify boundaries between clusters in the underlying data.

The Self-Organizing Map (SOM) Visualization Techniques

- **Component Planes**

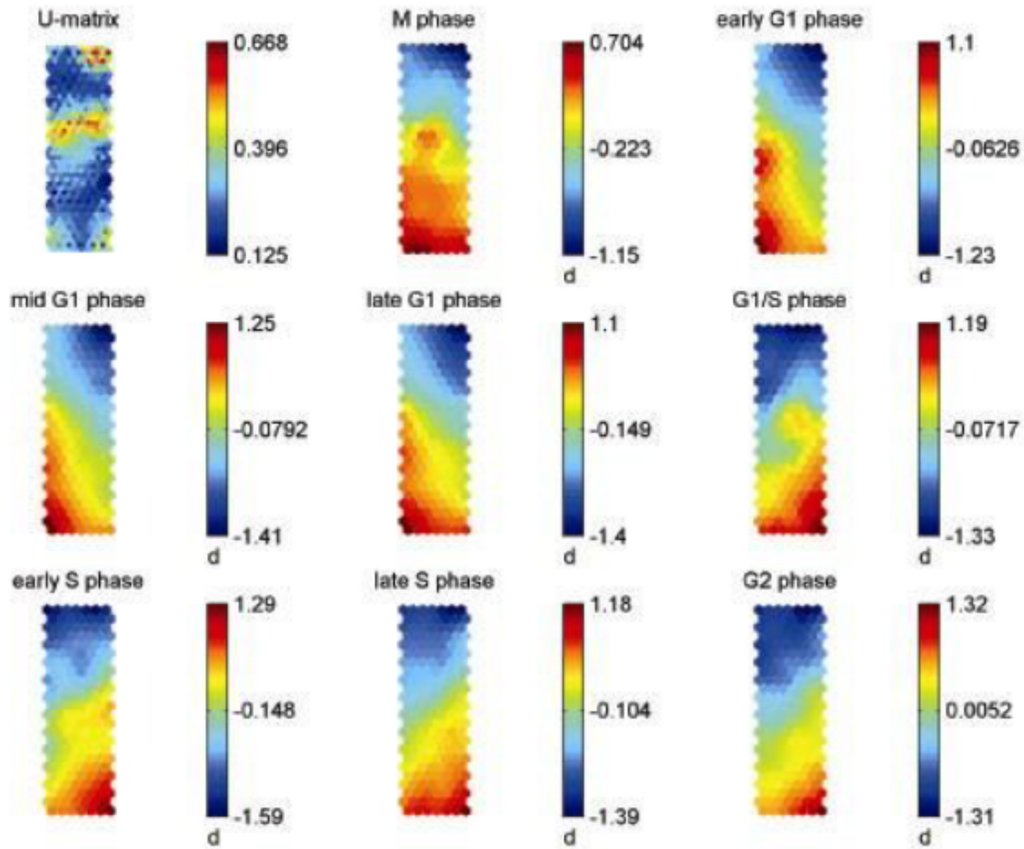
  is a representation that visualizes relative component values in the weight vectors of the SOM. The illustration can be considered as a sliced version of the SOM, where each plane shows the distribution of one weight vector component. Using the distributions, correlations between different components can be studied. Sometimes these component planes can be useful in interpreting the type of samples that belong to a cluster, comparing them to the U-Matrix.

SOM Visualization

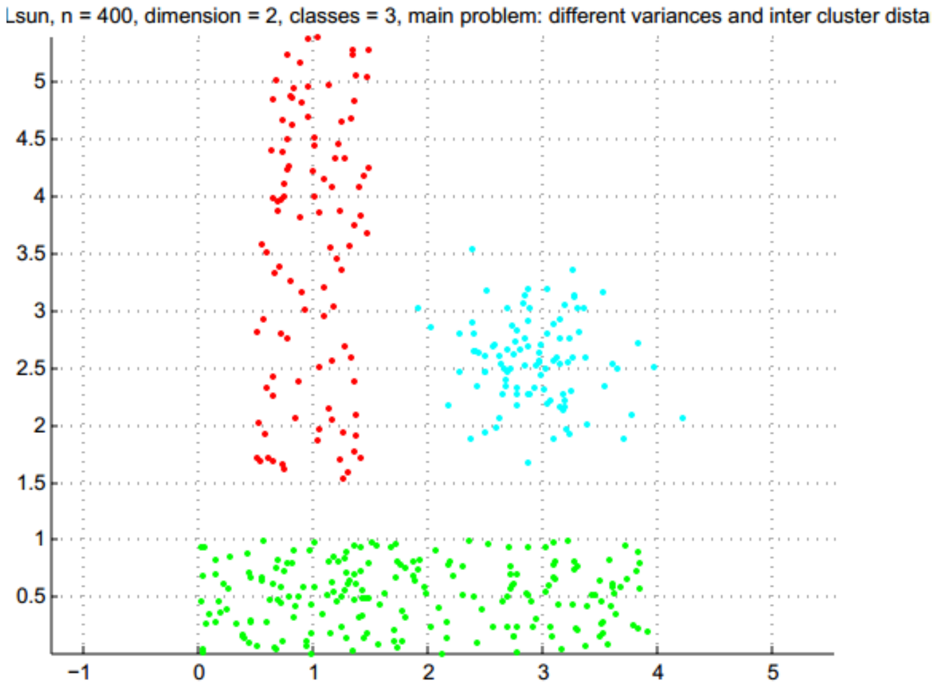


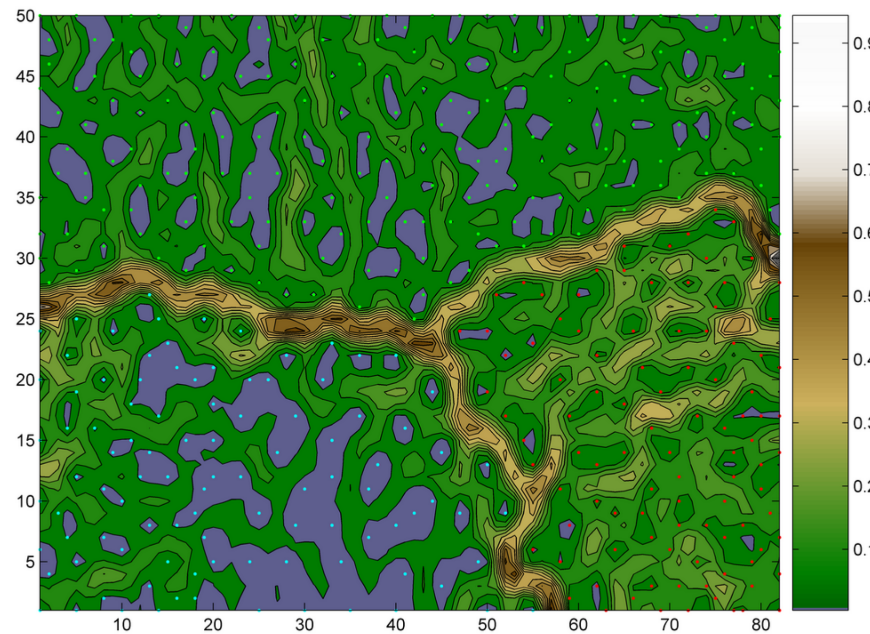
U-heigths are average local distances !

The Self-Organizing Map (SOM) Visualization Techniques - Example

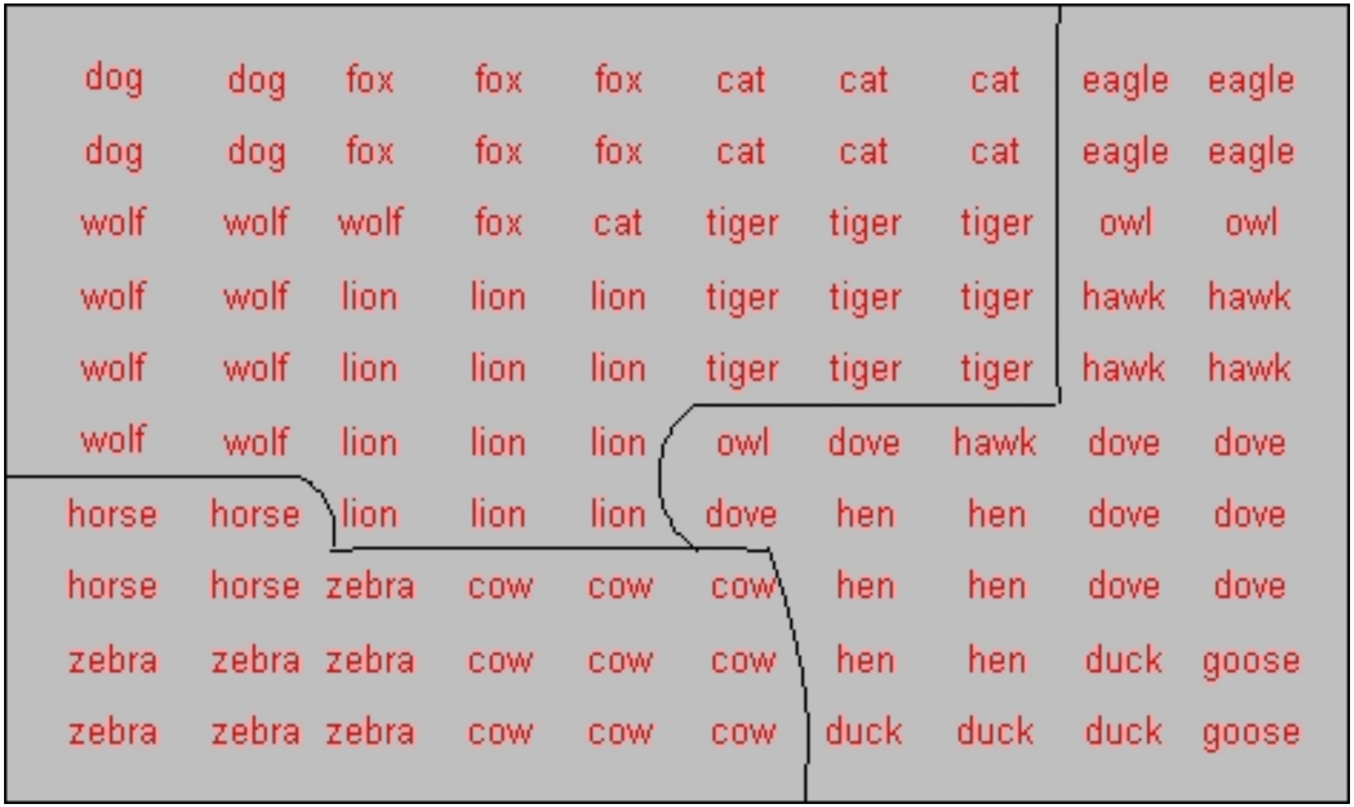


Example:
Lsun Clustering



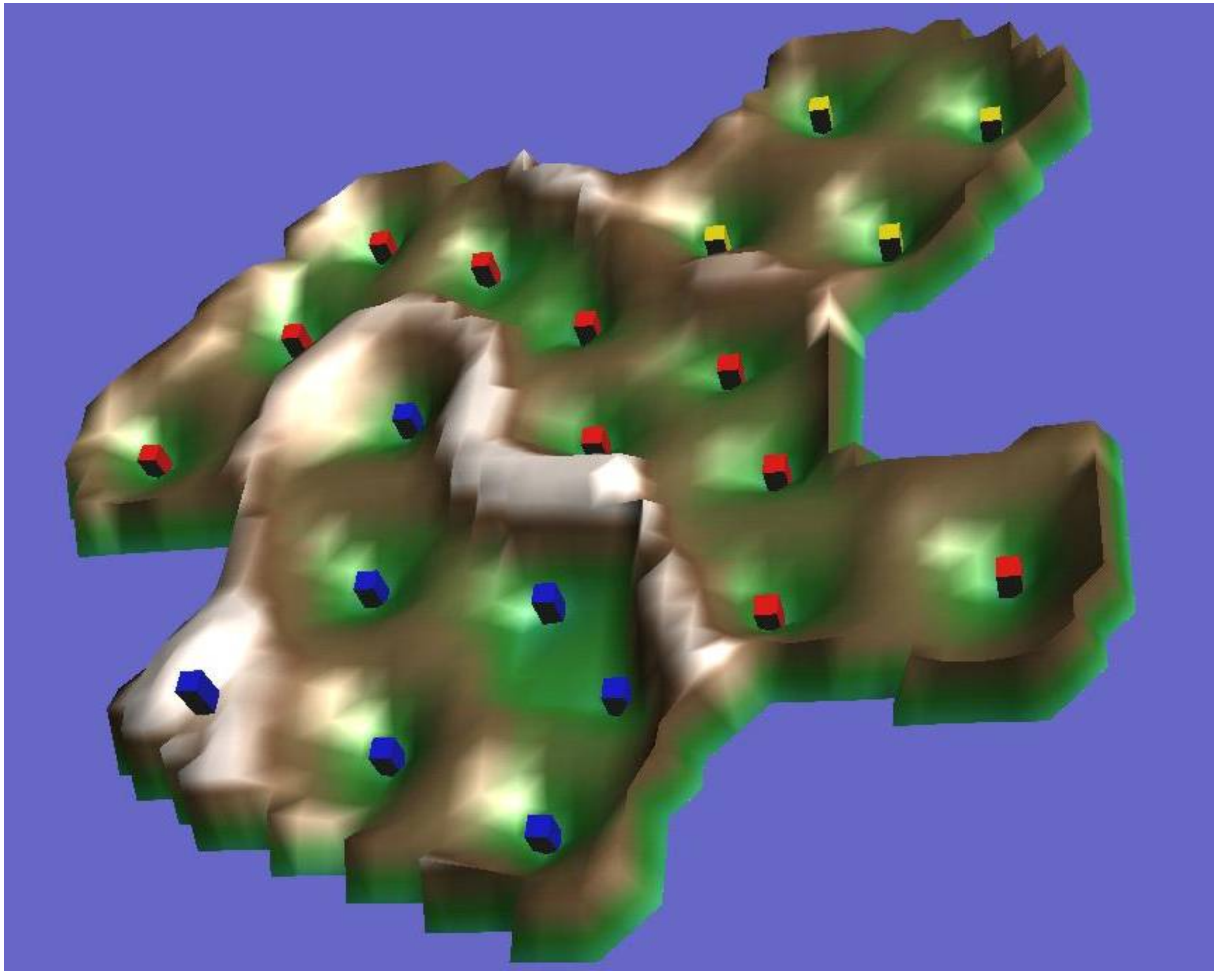


Another Example



We observe that there are three distinct clusters of animals: “birds”, “peaceful species” and “hunters”.

SOM DNA Microarray Clustering



http://www.uni-marburg.de/fb12/datenbionik/anwendungen

MusicMiner
http://musicminer.sourceforge.net/

The image cannot be displayed. Your computer may not have enough memory to open the image, or the image may have been corrupted. Restart your computer, and then open the file again. If the red x still appears, you may have to delete the image and then insert it again.

References

- [Koh1982] Teuvo Kohonen, *Self-organized formation of topologically correct feature maps*, Biological Cybernetics, 43:59-69, 1982
- [RMS1992] Helge Ritter, Thomas Martinetz and Klaus Schulten, *Neural computation and self-organizing maps: an introduction*, Addison-Wesley, 1992
- [YaZ2001] Jinsan Yang and Byoung-Tak Zhang, Customer Data Mining and Visualization by Generative Topographic Mapping Methods, In Simeon J. Simoff, Monique Noirhomme-Fraiture and Michael H. Böhlen eds., *Proceedings of the International Workshop on Visual Data Mining (VDM@ECML/PKDD2001)*, Freiburg, Germany, pp. 55-66, 4 September 2001