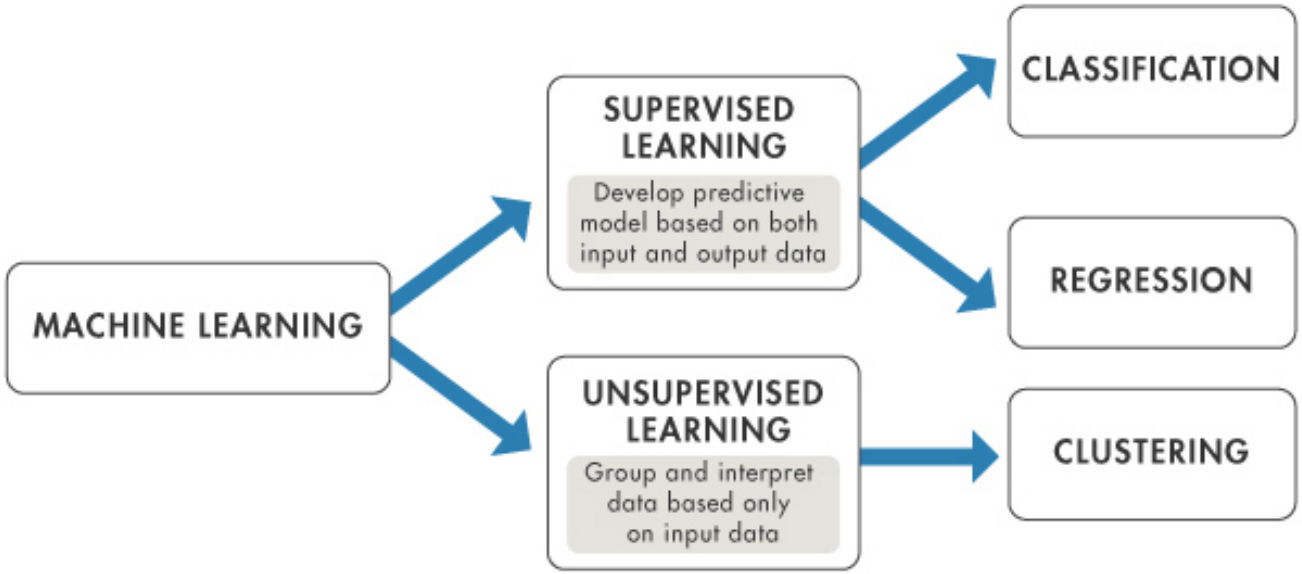
Introduction to Machine Learning (for PhD)

Marco Piangerelli – Computer Science Division & Math Division
marco.piangerelli@unicam.it
Sebastiano Pilati – Physics Division
sebastiano.pilati@unicam.it

27 June 2019

Overview


MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

???

???

The Chimera of Arezzo

"... a thing of immortal make, not human, lion-fronted and snake behind, a goat in the middle, and snorting out the breath of the terrible flame of bright fire."

Homer , Iliad

Similarity

Similarity has to be quantified and measured (otherwise it is useless)

Similarity

Similarity has to be quantified and measured (otherwise it is useless)

Similarity is measured in the range 0 to 1 $[0, 1]$.

Similarity

Similarity has to be quantified and measured (otherwise it is useless)

Similarity is measured in the range 0 to 1 $[0, 1]$.

Two main consideration about similarity:

Given two objects X and Y

- Similarity $= 1$ *if* $X = Y$
- Similarity $= 0$ *if* $X \neq Y$

Similarity

The fastest way to measure Similarity is to think if it as a “ distance"

$$Similarity\ between\ X\ and\ Y\ = d(X, Y) = ||X - Y||$$

Similarity

The fastest way to measure Similarity is to think if it as a “ distance"

$$Similarity\ \ between\ X\ \ and\ \ Y\ \ = d(X, Y) = ||X - Y||$$

$$SIMILARITY \approx DISTANCE$$

Similarity

The fastest way to measure Similarity is to think if it as a " distance"

$$Similarity\ between\ X\ and\ Y\ = d(X, Y) = ||X - Y||$$

$$SIMILARITY \approx DISTANCE$$

Which Distance?

Distances

- Hamming Distance
- Euclidean Distance
- Manhattan Distance
- Minkowski Distance
- Chebyshev Distance
- Cosine Distance
- Kullback-Leiber Distance
- Jaccard Distance
- Mahalanobis Distance

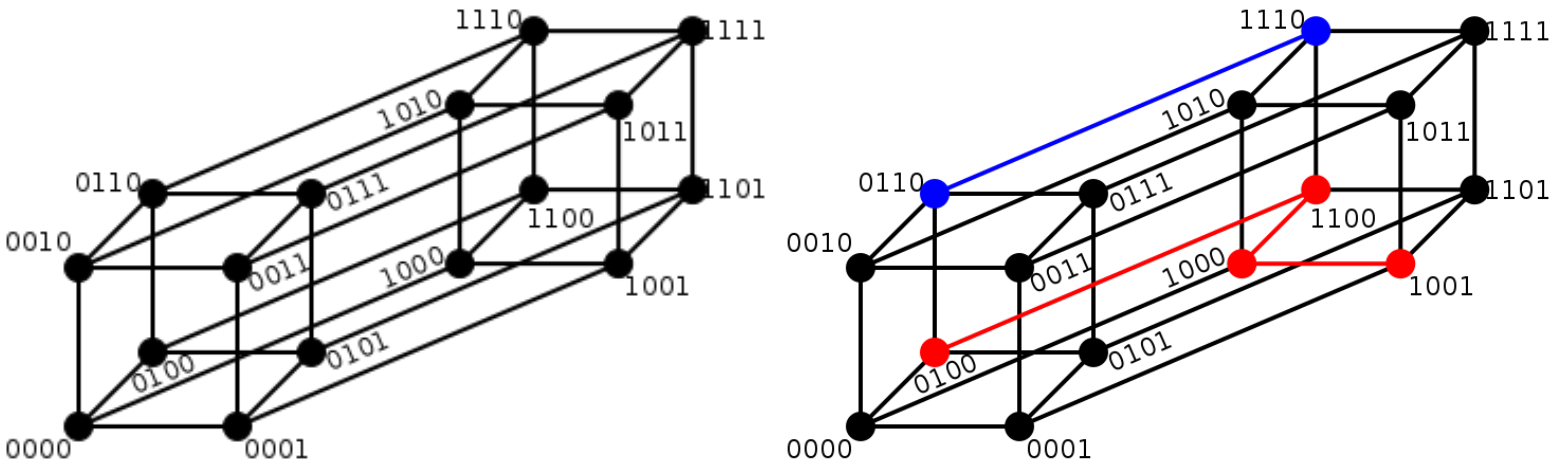
The Hamming Distance

The Hamming distance between two strings (or vectors) of equal length is the number of positions at which the corresponding symbols are different.

Example

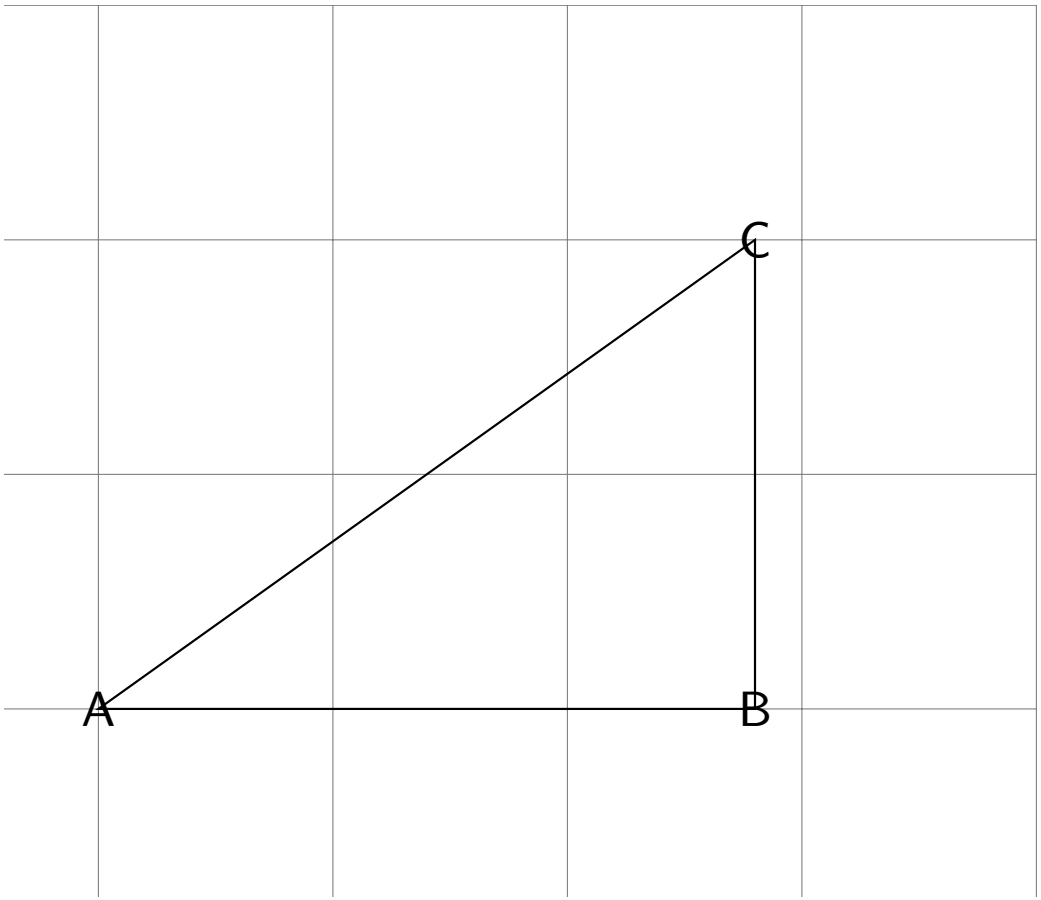
$X = \{0, 1, 0, 0, 1\}, \quad Y = \{1, 0, 0, 0, 1\}$

$d_{Hamming}(X, Y) = 2$



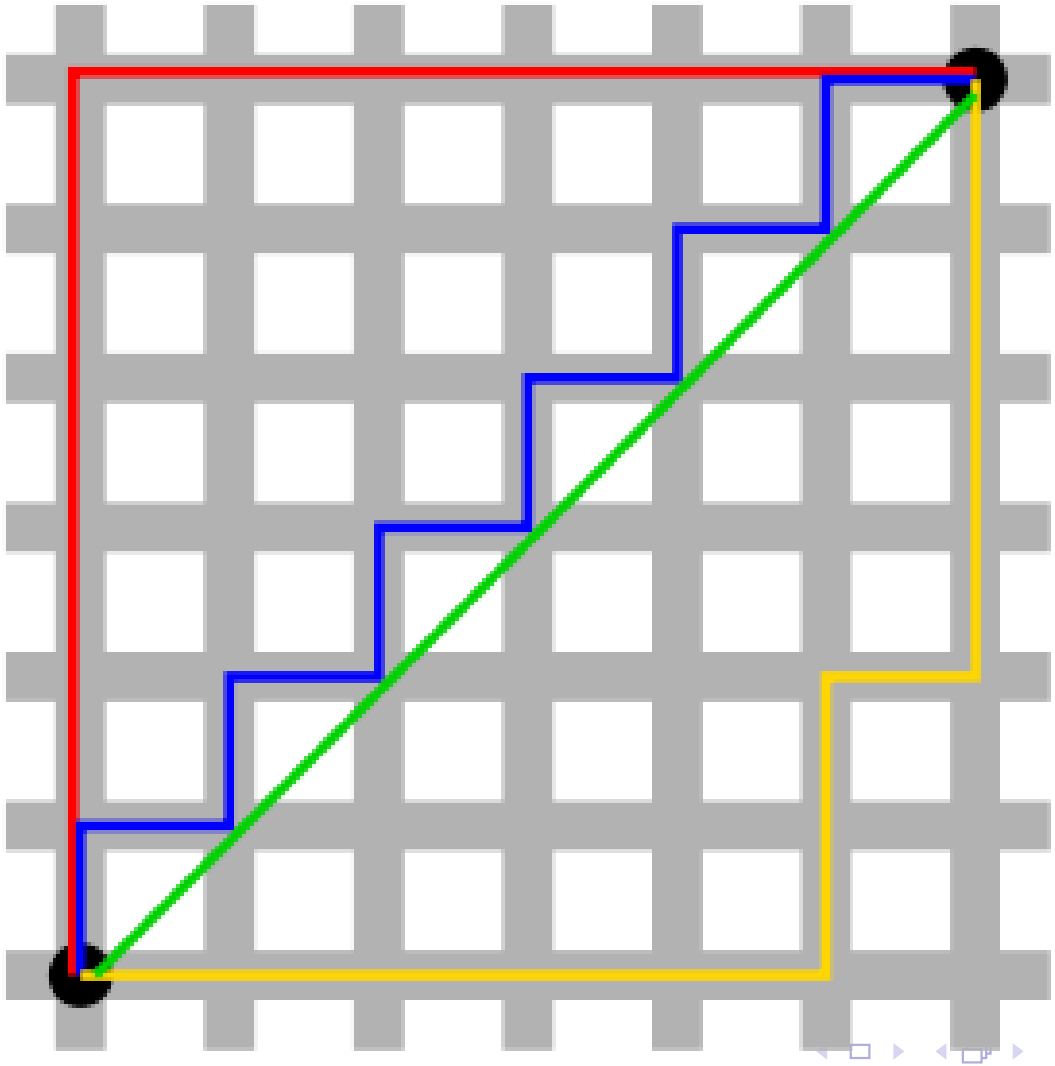
The Euclidean Distance

$$d_{Euclidean}(X, Y) = \sqrt{\sum_{i=0}^{N}(x_i - y_i^2)}$$



The Manhattan Distance

$$d_{Manhattan}(X, Y) = \sum_{i=0}^{N} |x_i - y_i|$$

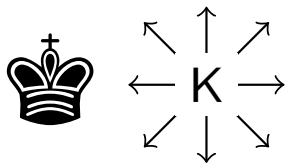


The Chebyshev Distance

$$d_{Chebyshev}(X, Y) = \sup_{0 \leq i \leq N} (|x_i - y_i|)$$

The Chebyshev Distance

$$d_{Chebyshev}(X, Y) = \sup_{0 \leq i \leq N} (|x_i - y_i|)$$



The Minkowski Distance

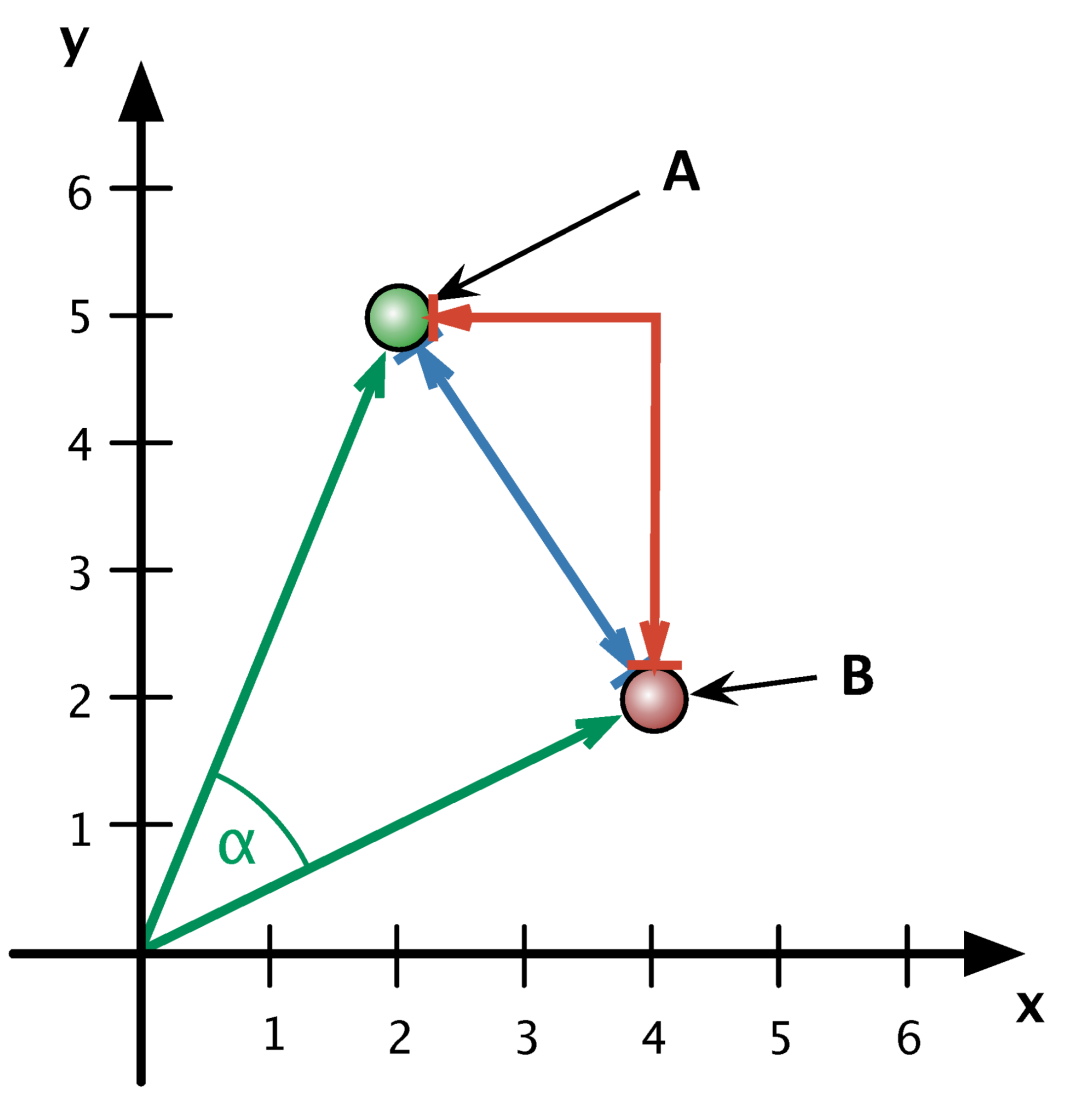
$$d_{Minkowsky}(X, Y) = \sqrt[p]{\sum_{i=0}^{N} |x_i - y_i|^p}$$

The KL Distance

Kullback-Leibler (KL) divergence:

- for histograms (x_d>0, $\Sigma_d x_d = 1$): $D(x,x') = -\sum_d x_d \log \frac{x_d}{x'_d}$

The Cosine Distance



What is clustering?

An informal definition for *Clustering* is: a procedure that aims to find "homogeneous" groups in a set of elements.

What is clustering?

Clustering is part of **unsupervised learning** that allows to discover hidden structures in data when we do not know anything in advance. The goal of clustering analysis is to group data according to their "similarity". The main idea is that similar items belong to the same cluster.

What is clustering?

There different approaches for performing a cluster analysis:

- prototype-based clustering
- hierarchical clustering
- density-based clustering
- neural model-based clustering
- ...

Prototype-based clustering

Prototype based clustering means that each cluster is represented by a prototype, for example a **centroid** of similar points. The well know **K-means** clustering belong to this family of clustering algorithms.
It is computationally very efficient and it is very at identifying "round" clusters.
Its drawbacks is the necessity of defining *a priori* the number, k, of clusters thus affecting the clustering performance.

K-means

- Randomly pick k centroids from the samples point as initial cluster centers.
- Assign each sample to the nearest centroid $\mu^{(j)}, j \in 1, ..., k$
- Move the centroids to the center of the samples that were assigned to it
- Repeat previous steps until the clusters do not change or a user's defined stop criterium is reached

K-means

How do we measure the "similarity" between two points?
In a n-dimensional space, the **squared Euclidean distance** can be used

$$d(\bar{x}, \bar{y})^2 = \sum_{j=1}^{n} (x_j - y_j)^2 = \|\bar{x} - \bar{y}\|_2^2$$

K-means

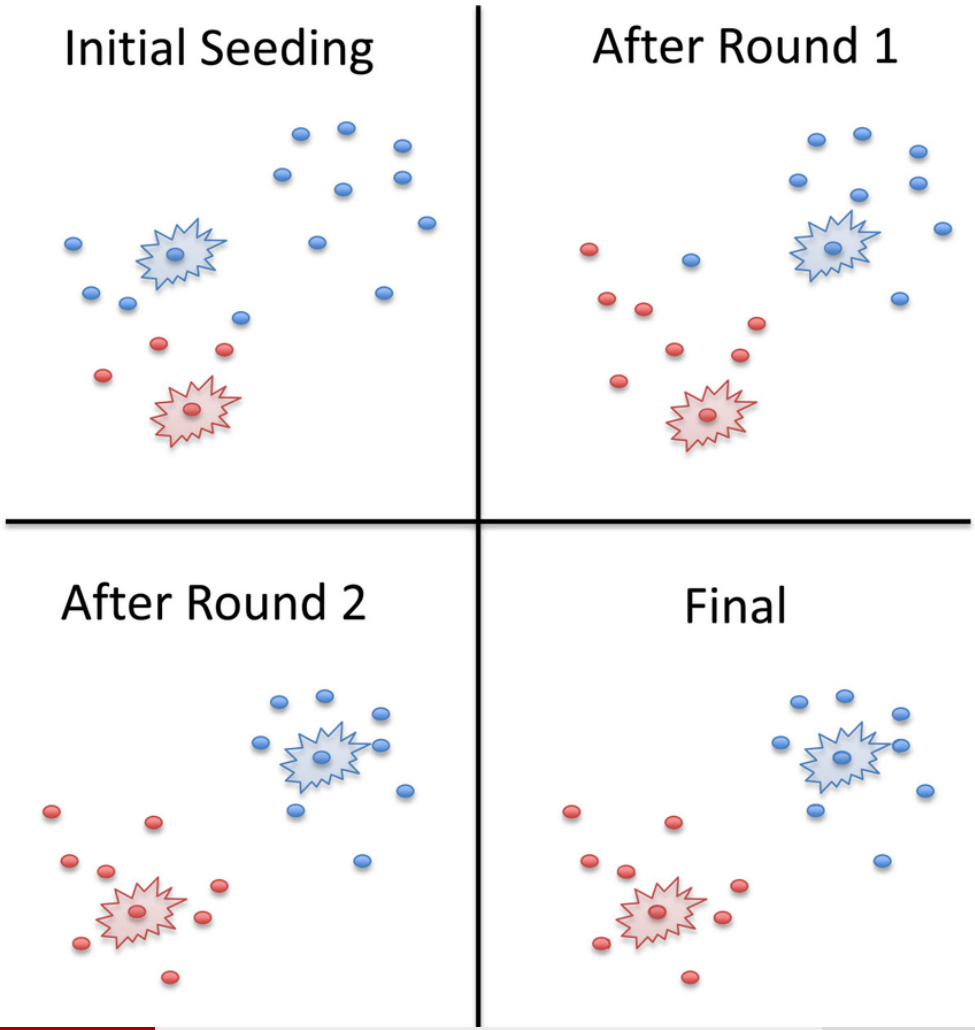
Based on the previous definition we can think the k-means cluster algorithm as an optimization problem.
K-means is partitional approach (breaking the dataset up into groups) and attempt to minimize the distance between points labeled to be in a cluster and a point designated as the center of that cluster.
We can define an iterative approach for minimizing the within cluster **Sum of Squared Errors (SSE)**

$$SSE = \sum_{i=1}^{m} \sum_{j=1}^{k} w^{(i,j)} \left\| \bar{x}^{(i)} - \bar{\mu}^{(j)} \right\|_2^2$$

$\bar{\mu}^{(j)}$ is the centroid for cluster j and $w^{(i,j)} = 1$ if sample $\bar{x}^{(j)}$ belong to the cluster; otherwise is 0.

Overview

In k-means the centre of a cluster is not necessarily one of the input data points



K-medoids

K-Means requires all the variables to be "Quantitative Variables" and using squared Euclidean Distance places the highest influence on largest distances.
$\rightarrow$ Lack of Robustness with respect to the Outliers (producing very large distances)

- Remember to scale your data before use k-means
- how to choose the number of clusters?
  - Instead of using centroids we use "medoids", i.e. the centers are among the data points
  - can be generalized for many distances in the minimization steps
  - the most famous algorithm is called PAM

K-means , K-medoids

- Remember to scale your data before use k-means
- how to choose the number of clusters?
  - Elbow rule $\rightarrow$ SSE for different values of k
  - Silhouette plots

Hierarchical Clustering

In hierarchical clustering (HC) we can choose two way for performing the cluster analysis:

- agglomerative cluster analysis (bottom-up)
- divisive cluster analysis (top-down)

Hierarchical Clustering

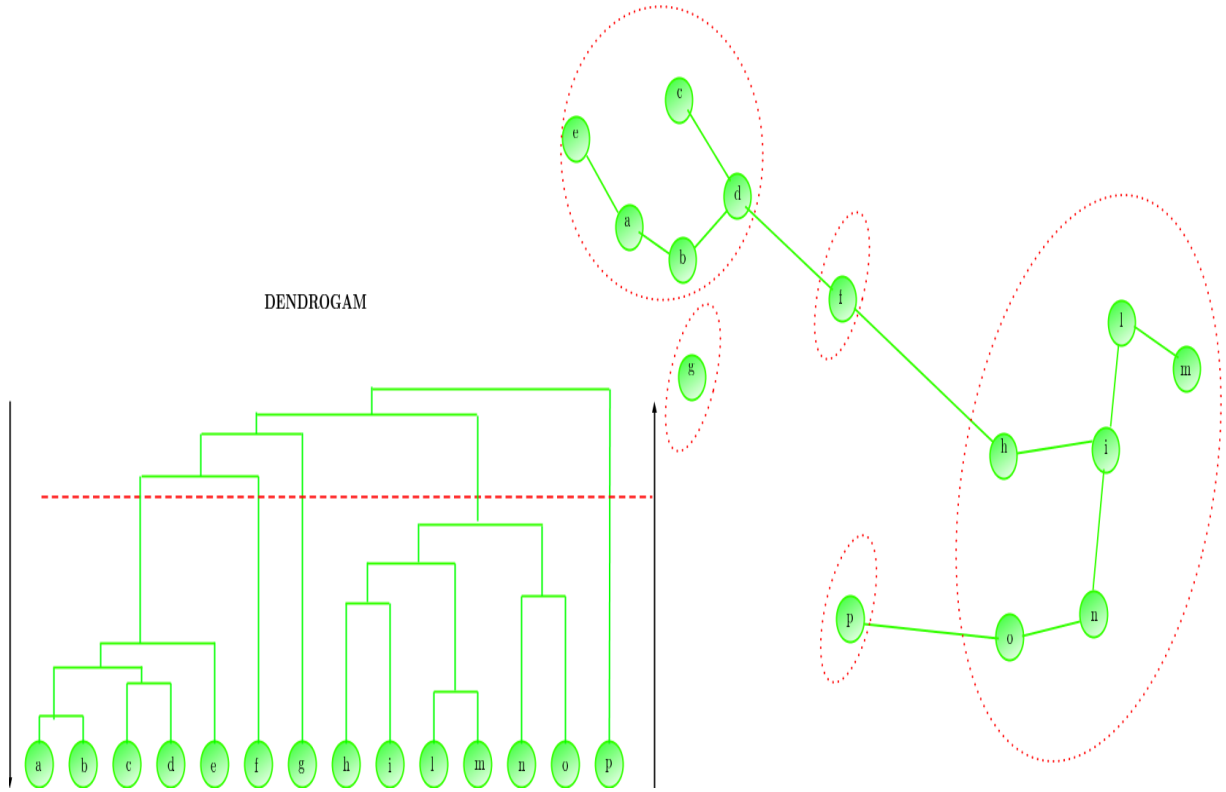
The two standard algorithms for agglomerative HC are:

- **Single Linkage**
- **Complete Linkage**

Single linkage allows us to compute the distance between the most similar members of each pair of clusters and merge the two cluster for which the distance between the most similar members is the smallest one; on the other hand complete linkage compare the most dissimilar members.

Hierarchical Clustering

The result of performing a HC procedure is a dendrogram

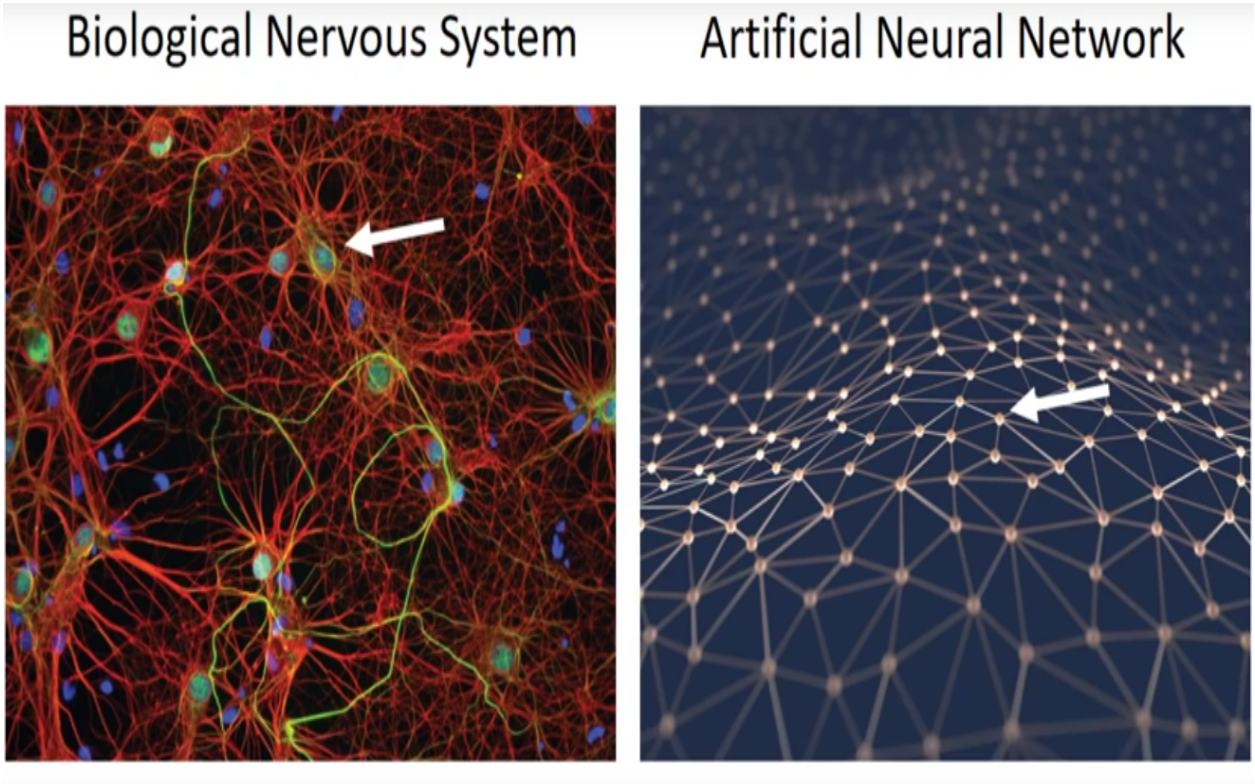


Cophenetic Correlation Coefficient → measure of how faithfully a dendrogram preserves the pairwise distances between the original unmodeled data points

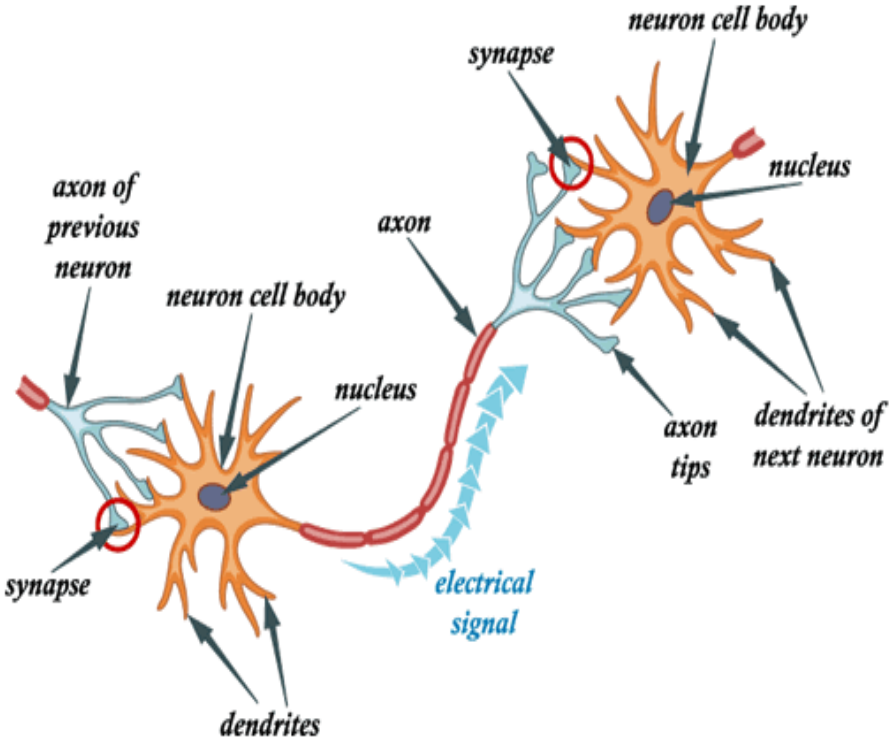
Neural Networks

Neural networks are computation systems inspired by the nervous system: both topologically and in terms of information processing.

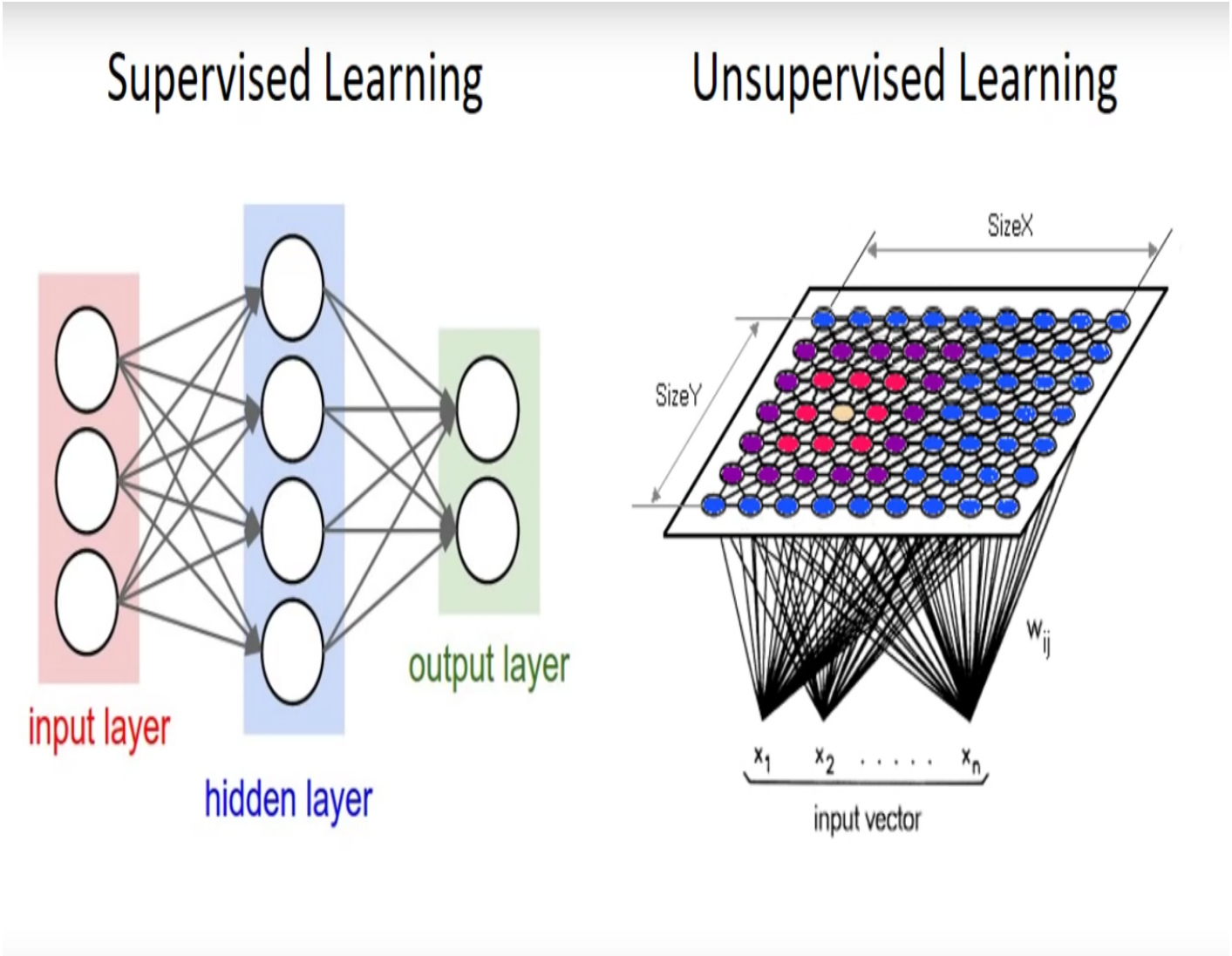
Neural Networks



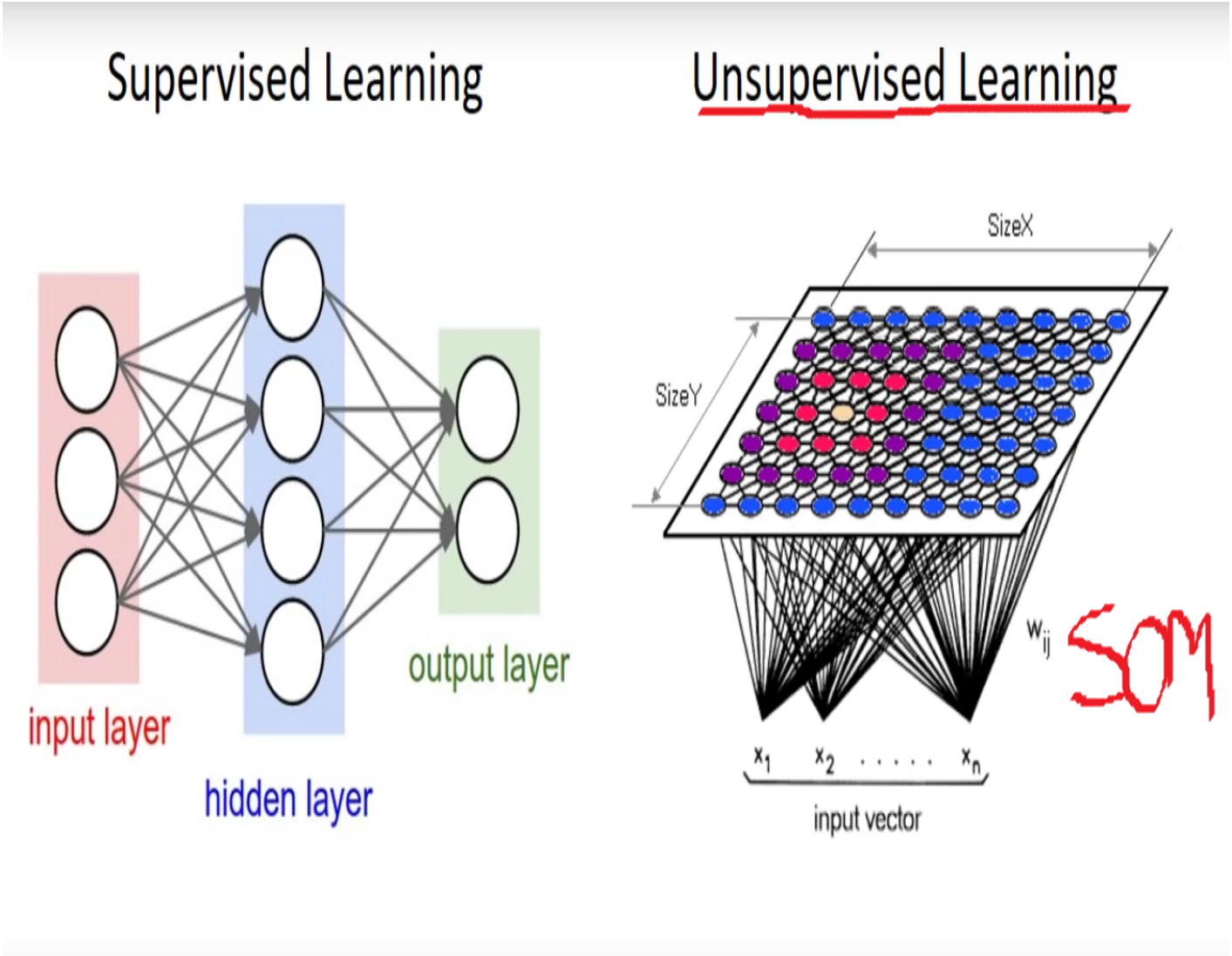
Neural Networks



Neural Networks



Neural Networks



Neural-based Clustering

The most popular approach belonging to this family is represented by **Self-organizing Map (SOM)**. SOMs are a particular neural network's architecture specifically designed for dealing with clustering or visualization or high dimensional dataset in lower dimensional space.

SOM

How does our brain store and recall the impressions it receives every day ?

SOM

How does our brain store and recall the impressions it receives every day ?
The brain does not have any training samples and therefore no "desired output".
There is no output in this sense at all, too

SOM

How does our brain store and recall the impressions it receives every day ?
The brain does not have any training samples and therefore no "desired output".
There is no output in this sense at all, too
Our brain responds to external input by changes in state. These are, so to speak, its output.

SOM

A paradigm of neural networks where the output is the state of the network, which learns completely unsupervised, i.e. without a teacher.

SOM

A paradigm of neural networks where the output is the state of the network, which learns completely unsupervised, i.e. without a teacher.

In SOM, we only ask which neuron is active at the moment. We are not interested in the exact output of the neuron but in knowing which neuron provides output.

SOM

A paradigm of neural networks where the output is the state of the network, which learns completely unsupervised, i.e. without a teacher.

In SOM, we only ask which neuron is active at the moment. We are not interested in the exact output of the neuron but in knowing which neuron provides output.

SOMs are considerably more related to biology than he feedforward networks, which are increasingly used for calculations.

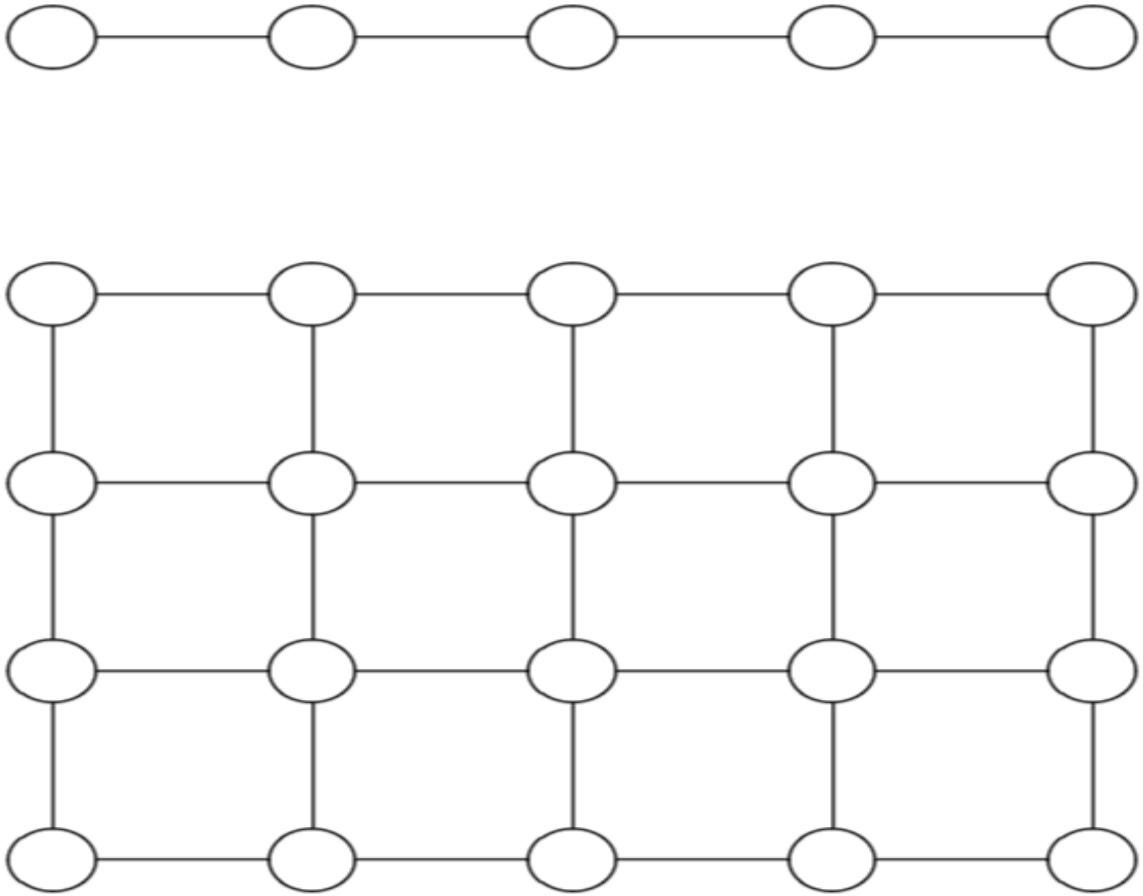
SOM

- Constrained version of K-means where the prototypes lie in 1d or 2D
- High-dimensional input (N dimensions) $\rightarrow$ low-dimensional grid of cells (G dimensions)

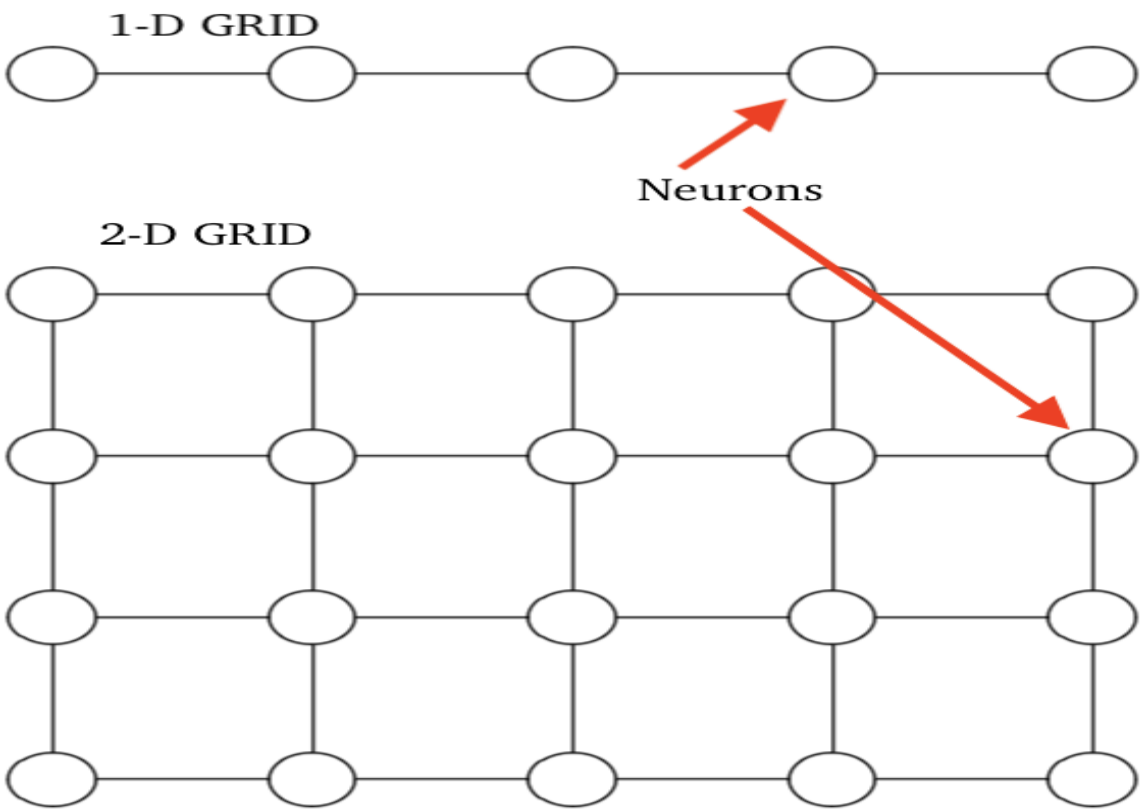
There are two spaces in which SOMs are working:

- The N-dimensional input space
- The G-dimensional grid that indicates the neighboring relationship, i.e. the *network topology or the constrained topological map*

SOM



SOM



Each neuron can be considered a Prototype, i.e. "centers" of cluster.

SOM

- Training
- Testing

SOM

- Training
- Testing

They are very similar and consists in the following steps:

SOM

- Training
- Testing

They are very similar and consists in the following steps:

- Input of an arbitrary value p of the input space $\mathcal{R}^n$
- Calculation of the distance between every neuron k and p by means of a norm, i.e. calculation of $\|(p - c_k)\|$.
- One neuron becomes active, namely such neuron i with the shortest calculated distance to the input. All other neurons remain inactive.This paradigm of activity is also called winner-takes-al l scheme. The output we expect due to the input of a SOM shows which neuron becomes active.

SOM

Training makes the SOM topology cover the input space

- Initialization: The network starts with random neuron centers $c_k \in \mathcal{R}^N$ from the input space
- Creating an input pattern: A stimulus, i.e. a point p, is selected from the input space $\mathcal{R}^N$
- Distance measurement: then the distance $\|(p - c_k)\|$ is determined for every neuron k in the network.
- Winner takes all: the winner neuron i is determined, which has the smallest distance to p, i.e. which fulfills the condition

$$\|(p - c_i)\| \leq \|(p - c_k)\| \ \forall k \neq i$$

SOM

- Adapting the centers: The neuron centers are moved within the input space according to the rule

$$\Delta c_k = \eta(t) * h(i, k, t) * (p - c_k)$$

where the values Δc_k are simply added to the existing centers. The last factor shows that the change in position of the neurons k is proportional to the distance to the input pattern p and, as usual, to a time- dependent learning rate $\eta(t)$. The above-mentioned network topology exerts its influence by means of the function $h(i, k, t)$, which will be discussed in the following.

SOM

A SOM does not need a target output to be specified unlike many other types of network. Instead, where the node weights match the input vector, that area of the lattice is selectively optimized to more closely resemble the data for the class the input vector is a member of. From an initial distribution of random weights, and over many iterations, the SOM eventually settles into a map of stable zones. Each zone is effectively a feature classifier, so you can think of the graphical output as a type of feature map of the input space.

SOM

SOM learning rule

(Definition). A SOM is trained by presenting an input pattern and determining the associated winner neuron. The winner neuron and its neighbor neurons, which are defined by the topology function, then adapt their centers according to the rule

$$\Delta c_k = \eta(t) * h(i,k,t) * (p - c_k)$$

$$\Delta c_k(t+1) = c_k(t) + Deltac_k(t)$$

SOM- topology function

The topology function h is not defined on the input space but on the grid and represents the neighborhood relationships between the neurons, i.e. the topology of the network. It can be time-dependent (which it often is) - which explains the parameter t. The parameter k is the index running through all neurons, and the parameter i is the index of the winner neuron. In principle, the function shall take a large value if k is the neighbor of the winner neuron or even the winner neuron itself, and small values if not. A more precise definition: The topology function must be unimodal, i.e. it must have exactly one maximum. This maximum must be next to the winner neuron i, for which the distance to itself certainly is 0. The function h needs some kind of *distance notion* on the grid because from somewhere it has to know how far i and k are apart from each other on the grid. There are different methods to calculate this distance. On a two-dimensional grid we could apply, for instance, the Euclidean distance or on a one-dimensional grid we could simply use the number of the connections between the neurons i and k.

SOM- topology function

Topology function

(Definition). The topology function $h(i, k, t)$ describes the neighborhood relationships in the topology. It can be any unimodal function that reaches its maximum when $i = k$. Time-dependence is optional, but often used.

Euclidean distance

$$h(i, k, t) = e^{-\frac{\|g_i - c_k\|^2}{2\sigma^2(t)}}, \sigma \textit{ is the radius that decays over time}$$

SOM

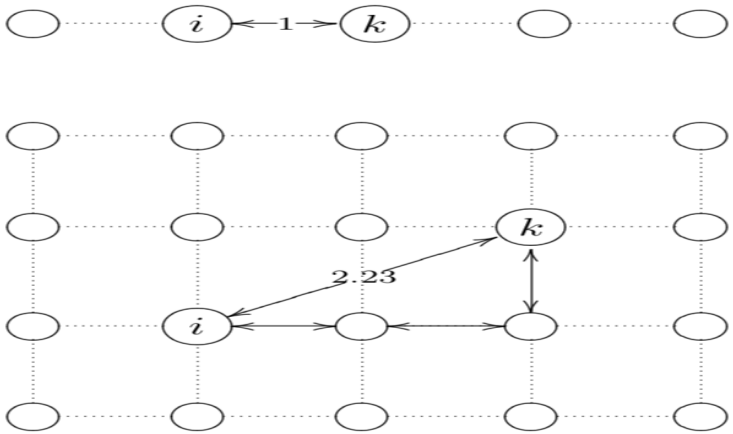
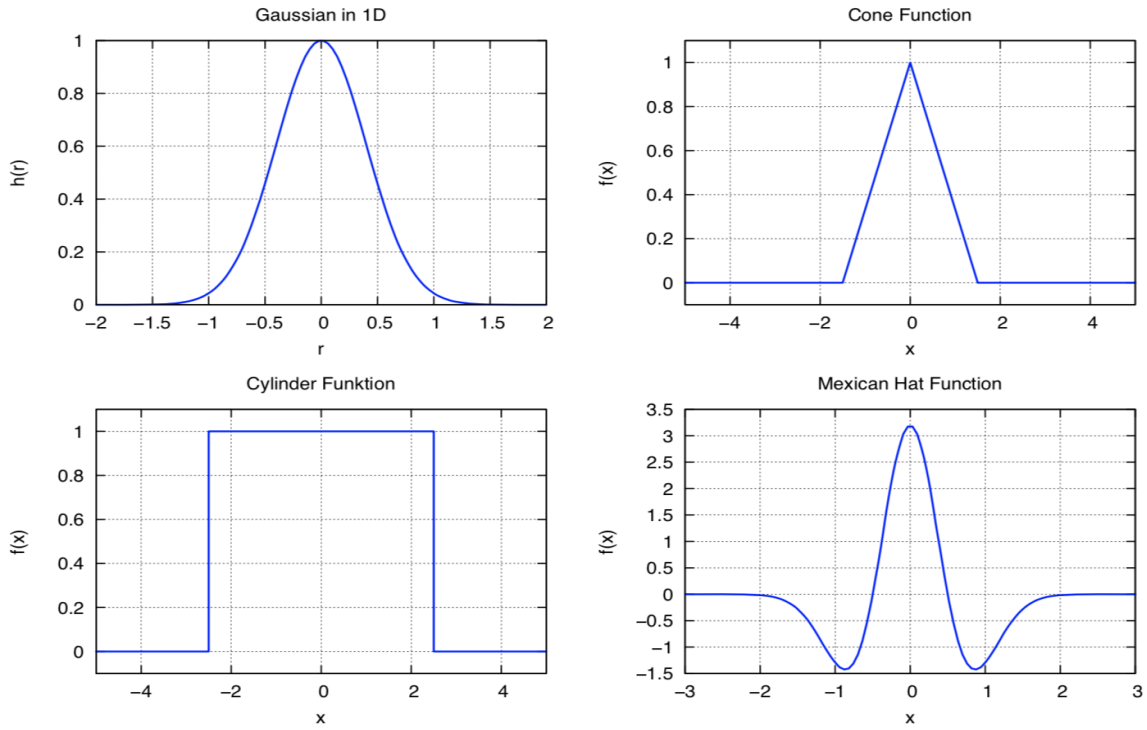


Figure : Example distances of a one-dimensional SOM topology (above) and a two-dimensional SOM topology (below) between two neurons i and k. In the lower case the Euclidean distance is determined (in two-dimensional space equivalent to the Pythagoream theorem). In the upper case we simply count the discrete path length between i and k. To simplify matters I required a fixed grid edge length of 1 in both cases.

SOM


Gaussian in 1D
h(r)
r
Cone Function
f(x)
x
Cylinder Funktion
f(x)
x
Mexican Hat Function
f(x)
x

SOM-Example

We use a two-dimensional input space, i.e. $N = 2$. Let the grid structure be one-dimensional ($G = 1$). Furthermore, our example SOM should consist of 7 neurons and the learning rate should be $\eta = 0.5$.
The neighborhood function is also kept simple so that we will be able to mentally comprehend the network:

$$h(i,k,t) = \begin{cases} 1 & k \text{ direct neighbor of } i, \\ 1 & k = i, \\ 0 & \text{otherw.} \end{cases}$$

p is our training sample

SOM-Example

- Learning direction: Remember that the neuron centers c_k are vectors in the input space, as well as the pattern p. Thus, the factor $(p - c_k)$ indicates the vector of the neuron k to the pattern p. This is now multiplied by different scalars:
- Our topology function h indicates that only the winner neuron and its two closest neighbors (here: 2 and 4) are allowed to learn by returning 0 for all other neurons. A time-dependence is not specified. Thus, our vector $(p - c_k)$ is multiplied by either 1 or 0.
- The learning rate indicates, as always, the strength of learning. As already mentioned, $\eta = 0.5$, i. e. all in all, the result is that the winner neuron and its neighbors (here: 2, 3 and 4) approximate the pattern p half the way (in the figure marked by arrows).

SOM-Example