Neural Networks: Kohonen Self Organizing Maps

Andrzej Kordecki

Neural Networks (ML.ANK385 and ML.EM05): Lecture 10
Division of Theory of Machines and Robots
Institute of Aeronautics and Applied Mechanics
Faculty of Power and Aeronautical Engineering
Warsaw University of Technology

Table of Contents

Self Organizing Map

Introduction to Self Organizing Map

In unsupervised (self-organized) learning, there is no external teacher to oversee the learning process:

- NN learn to form their own classifications of the unlabeled training data (clustering),
- NN training assume that class membership is defined by the common features of input patterns - we want to identify those features,

The result of direct implementation of this idea on network is that the neurons are forced to organize themselves - Self Organizing Map.

T. Kohonen, "Self-Organized Formation of Topologically Correct Feature Maps", Biological Cybernetics, 43(1):59–69, 1982.

Introduction to Self-Organizing Map

The unsupervised learning may be pursued from different perspectives:

- Self-organized learning, the formulation of which is motivated by neurobiological considerations. The algorithm set a number of rules of local behavior, and the requirement is to use the rules to compute an input-to-output mapping with desirable properties.
- Statistical learning theory, which is the approach that is traditionally used in machine learning. The algorithm is focused on well-established statistical tools.

Introduction to Self-Organizing Map

The development of SOM as a neural model is motivated by a distinct feature of the human brain:

- The brain is organized in many places in such a way that different sensory inputs are represented by topologically ordered computational maps.
- Biological studies indicate that different sensory inputs (visual, auditory, etc.) are mapped onto corresponding areas of the cerebral cortex in an orderly fashion.

A computational map is defined by an array of neurons representing slightly differently tuned processors or filters, which operate on the sensory information bearing signals in parallel.

Topographic Maps

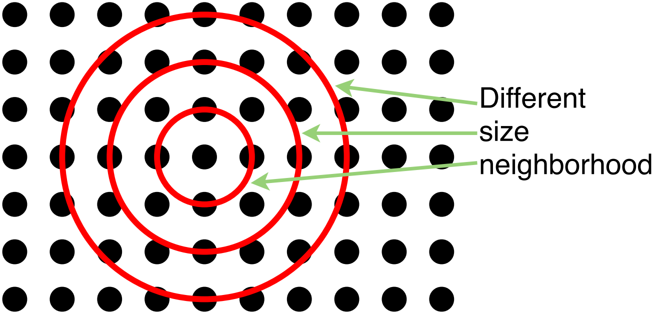
This form of map known as a topographic map:

- At each stage of processing each piece of incoming information is kept in its proper neighborhood.
- Neurons dealing with closely related pieces of information are kept close together so that they can interact via short synaptic connections.

The principle of topographic map formation: “The spatial location of an output neuron in a topographic map corresponds to a particular domain or feature drawn from the input space”.

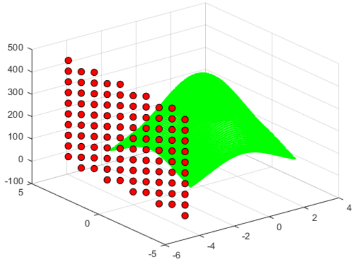
Topographic Maps

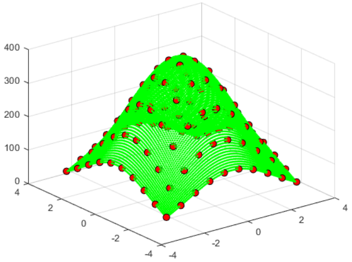
Topographic map represented as 2-dimensional rectangular grid filled with neurons.



Topographic Maps

Topographic map represented as 2-dimensional grid of neurons mapped onto corresponding input data.





Topographic Maps

The self-organizing map (SOM) use a neighborhood function to preserve the topological properties of the input space.

That is, a property of spaces is a topological property if whenever a space X possesses that property every space homeomorphic to X possesses that property. Topological properties examples:

- Separation: if for every pair of distinct points x and y in the space, there is at least either an open set containing x but not y, or an open set containing y but not x.
- Connected: A space is connected if it is not the union of a pair of disjoint non-empty open sets.

Topographic Maps

The SOM is a NN based on competitive learning of Rosenblatt rule:

- The output neurons of the network compete among themselves to be activated, with the result that only one output neuron (or one neuron per group) is on at any one time.
- An output neuron that wins the competition is called a winner-takesall neuron or a winning neuron. One way of inducing a winner-takes-all competition among the output neurons is to use lateral inhibitory connections between them.

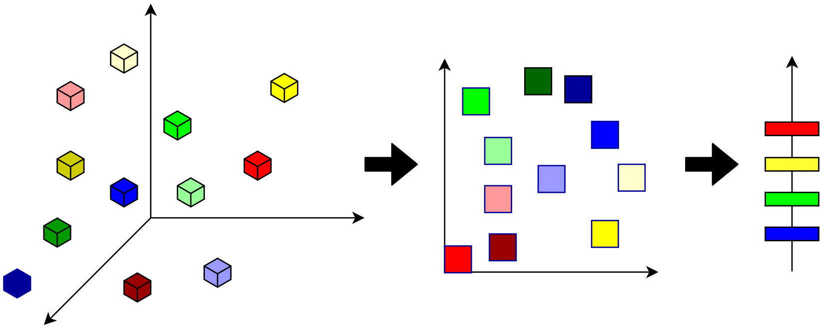
The goal of the SOM

A common problem in statistical pattern recognition is:

- Feature selection refers to a process whereby a data space is transformed into a feature space that has exactly the same dimension as the input data space,
- Feature extraction refers to transformation designed in such a way that the data set may be represented by a reduced number of "effective" features, yet retain most of the intrinsic information content of the original data. In other words, the data set undergoes a dimensionality reduction.

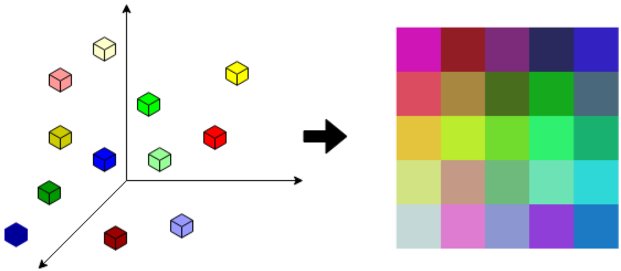
The goal of the SOM

Dimensionality reduction



The goal of the SOM

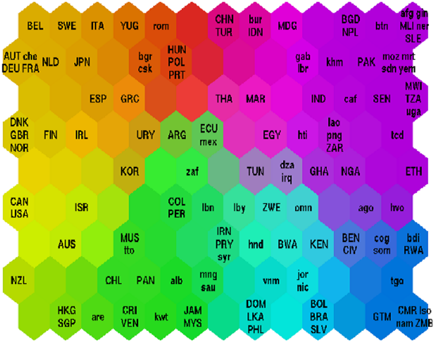
SOM provides a data visualization technique which helps to understand high dimensional data by reducing the dimensions of data to a map.

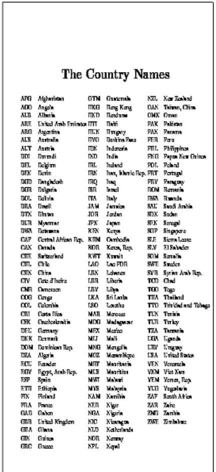


SOM also represents clustering concept by grouping similar data together.

The goal of the SOM

Example of data visualization: The World Poverty Classifier is a SOM that maps countries based on 39 "quality-of-life factors" including health, nutrition, education, etc.



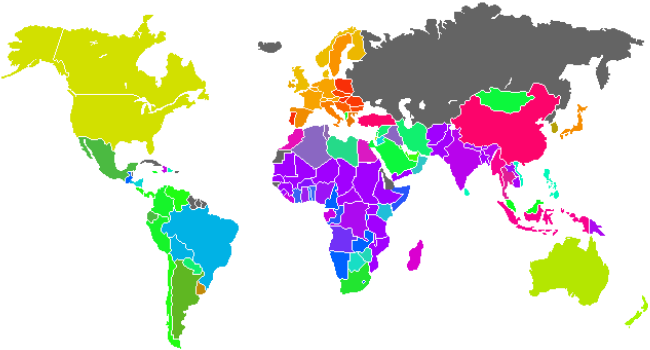


The Country Names

AFG	Afghanistan	GTM	Guatemala	NZL	New Zealand
AGO	Angola	HKG	Hong Kong	OAN	Taiwan, China
ALB	Albania	HND	Honduras	OMN	Oman
ARE	United Arab Emirates	HTI	Haiti	PAK	Pakistan
ARG	Argentina	HUN	Hungary	PAN	Panama
AUS	Australia	HVO	Burkina Faso	PER	Peru
AUT	Austria	IDN	Indonesia	PHL	Philippines
BDI	Burundi	IND	India	PNG	Papua New Guinea
BEL	Belgium	IRL	Ireland	POL	Poland
BEN	Benin	IRN	Iran, Islamic Rep.	PRT	Portugal
BGD	Bangladesh	IRQ	Iraq	PRY	Paraguay
BGR	Bulgaria	ISR	Israel	ROM	Romania
BOL	Bolivia	ITA	Italy	RWA	Rwanda
BRA	Brazil	JAM	Jamaica	SAU	Saudi Arabia
BTN	Bhutan	JOR	Jordan	SDN	Sudan
BUR	Myanmar	JPN	Japan	SEN	Senegal
BWA	Botswana	KEN	Kenya	SGP	Singapore
CAF	Central African Rep.	KHM	Cambodia	SLE	Sierra Leone
CAN	Canada	KOR	Korea, Rep.	SLV	El Salvador
CHE	Switzerland	KWT	Kuwait	SOM	Somalia
CHL	Chile	LAO	Lao PDR	SWE	Sweden
CHN	China	LBN	Lebanon	SYR	Syrian Arab Rep.
CIV	Cote d'Ivoire	LBR	Liberia	TCD	Chad
CMR	Cameroon	LBY	Libya	TGO	Togo
COG	Congo	LKA	Sri Lanka	THA	Thailand
COL	Colombia	LSO	Lesotho	TTO	Trinidad and Tobago
CRI	Costa Rica	MAR	Morocco	TUN	Tunisia
CSK	Czechoslovakia	MDG	Madagascar	TUR	Turkey
DEU	Germany	MEX	Mexico	TZA	Tanzania
DNK	Denmark	MLI	Mali	UGA	Uganda
DOM	Dominican Rep.	MNG	Mongolia	URY	Uruguay
DZA	Algeria	MOZ	Mozambique	USA	United States
ECU	Ecuador	MRT	Mauritania	VEN	Venezuela
EGY	Egypt, Arab Rep.	MUS	Mauritius	VNM	Viet Nam
ESP	Spain	MWI	Malawi	YEM	Yemen, Rep.
ETH	Ethiopia	MYS	Malaysia	YUG	Yugoslavia
FIN	Finland	NAM	Namibia	ZAF	South Africa
FRA	France	NER	Niger	ZAR	Zaire
GAB	Gabon	NGA	Nigeria	ZMB	Zambia
GBR	United Kingdom	NIC	Nicaragua	ZWE	Zimbabwe
GHA	Ghana	NLD	Netherlands		
GIN	Guinea	NOR	Norway		
GRC	Greece	NPL	Nepal		

The goal of the SOM

Example of data visualization: The map of the world where the countries have been colored in by their corresponding locations in the SOM.



https://www.cs.hmc.edu/ kpang/nn/som.html

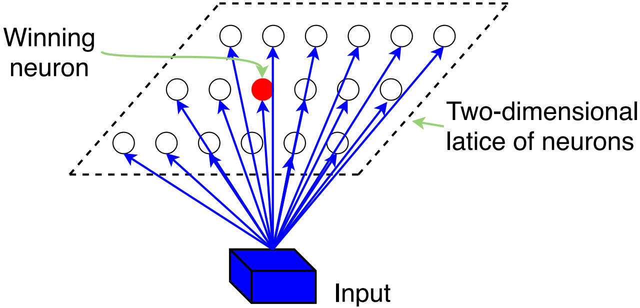
The goal of the SOM

The principal goal of the SOM is to transform an incoming input signal of arbitrary dimension into a one- or two-dimensional discrete map, and to perform this transformation adaptively in a topologically ordered fashion.
Definition:

> A self-organizing map is characterized by the formation of 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 patterns - hence, the name "self-organizing map."

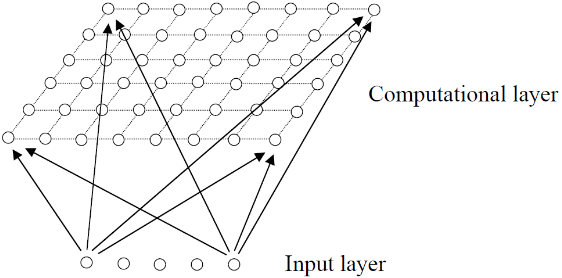
Setting up SOM

The Kohonen SOM has a feed-forward structure with a single computational layer arranged in rows and columns.



Setting up SOM

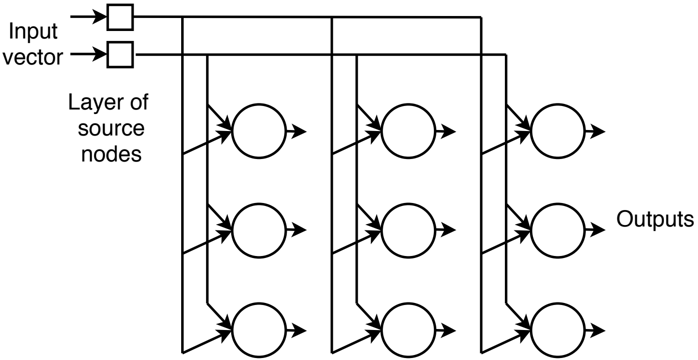
Each neuron is fully connected to all the source nodes in the input layer:



The one dimensional map will just have a single row (or a single column) in the computational layer.

The feature-mapping model

The 2-dimensional lattice of neurons, illustrated for a 2-dimensional input and 3-by-3 dimensional output.



Two-dimensional lattice of neurons,

Setting up SOM

Neuron connection:

- Each input sample typically consists of a localized lattice region of activity against a quiet background. The location and nature of such a spot usually varies from one realization of the input sample to another.
- All the neurons in the network should therefore be exposed to a sufficient number of different realizations of the input pattern in order to ensure that the self-organization process has a chance to develop properly.

The feature-mapping model

The principle of topographic map formation, stated as follows (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.

This principle has provided the neurobiological motivation for different feature-mapping models.

Training SOM

Training SOM

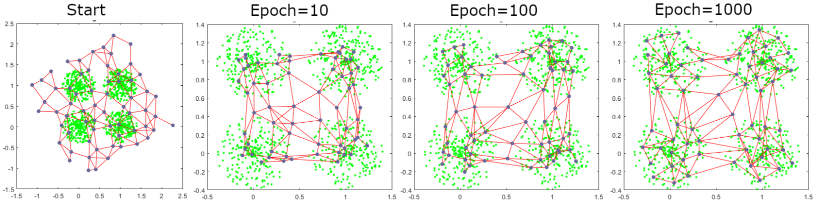
Training of SOM differs from other NN, because it applies competitive learning as opposed to error-correction learning (e.g. backpropagation).

The self-organization process involves 3 major steps:

- Competition: For each input sample, the neurons compute their respective values of a discriminant function which provides the basis for competition. The smallest value of the discriminant function indicate the winner neuron.
- Cooperation: The winning neuron determines the spatial location of a topological neighborhood of excited neurons, thereby providing the basis for cooperation among neighboring neurons.

Training SOM

- Adaptation: The excited neurons decrease their individual values of the discriminant function in relation to the input pattern through suitable adjustment of the associated connection weights, such that the response of the winning neuron to the subsequent application of a similar input pattern is enhanced.



The goal is: dimension reduction or clusterization.

Definitions

The SOM network is a feedforward network with 2 layers: an input layer and an output layer.

- Let an input pattern (vector) selected at random is M dimensional be denoted by $x = \{x_i : i = 1, ..., M\}$,
- The connection weights between the input units i and the neurons j in the computation layer can be written $w_j = \{w_{j,i} : j = 1, ..., N; i = 1, ..., M\}$, where N is the total number of neurons in the network.

The Competitive Process

The Competitive Process:

- The highest output value of the neuron will determine whether a given sample belongs to a given cluster.
- The neuron with the largest inner product $w_j^T x$ determine the location where the topological neighborhood of excited neurons is to be centered (largest neuron output value).
- The maximization of the inner product $w_j^T x$ is mathematically equivalent to minimizing the distance between the vectors x and w_j, provided that w_j has unit length for all j.

The Competitive Process

We can identify index $i(x)$ of the wining neuron that best matches the input vector x by applying the following discriminant function:

$$i(x) = \arg\min_j ||x - w_j|| \quad j \in D$$

where D denotes the lattice of neurons.

The Competitive Process

The best-matching criterion of discriminant function is the squared Euclidean distance between the input vector x and the weight vector w_j for each neuron j:

$$d_j(x) = \sum_{i=1}^{M} (x_i - w_{j,i})^2$$

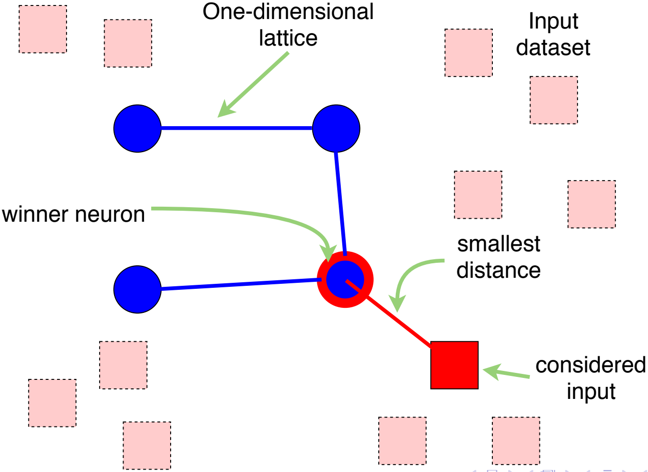
The particular neuron i that satisfies this condition is called the winning neuron for the input vector x:

$$i(x) = \arg\min_j d_j(x)$$

In other words, the neuron whose weight vector comes closest to the input vector (i.e., is most similar to it) is declared the winner.

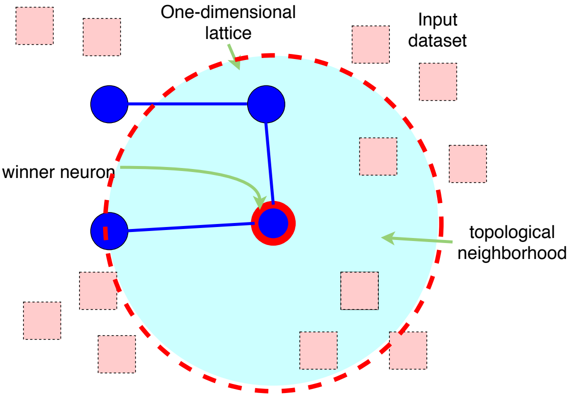
The Competitive Process

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



The Cooperative Process

The topological neighborhood $H_{j,i}$ centered on winning neuron i is encompassing a set of excited (cooperating) neurons, a typical ones of which are denoted by j.



The Cooperative Process

We assume that the topological neighborhood $H_{j,i}$ is a unimodal function of the lateral distance $d_{j,i}$, such:

- The topological neighborhood $H_{j,i}$ is symmetric about the maximum point defined by $d_{j,i} = 0$, which is equivalently attains its maximum value at the winning neuron i for which the distance $d_{j,i} = 0$.
- The amplitude of the topological neighborhood $H_{j,i}$ decreases monotonically with increasing lateral distance $d_{j,i}$, decaying to zero for $d_{j,i} \to \infty$. This is a necessary condition for convergence.

The Cooperative Process

The concept of neighborhood is taken into account through kernel functions H that satisfies these requirements:

- Threshold neighborhood kernel (bubble):

$$H_{j,i(x)} = \begin{cases} 1 & d_{j,i(x)} < 1 \\ 0 & otherwise \end{cases}$$

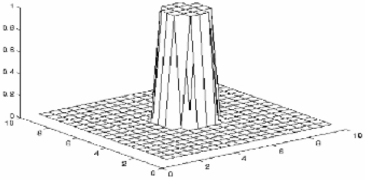
- Gaussian neighborhood kernel:

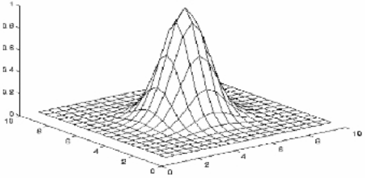
$$H_{j,i(x)} = \exp\left(- \frac{d^2_{j,i(x)}}{2\sigma^2} \right)$$

It is translation invariant, i.e., independent of the location of winning neuron in lattice.

The Cooperative Process

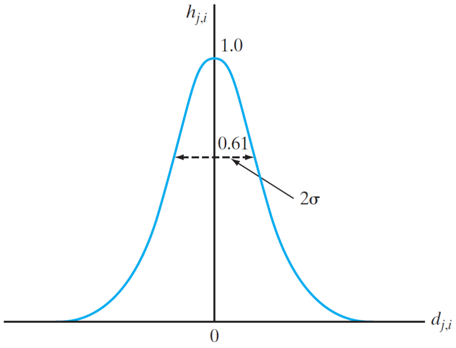
Threshold neighborhood kernel and Gaussian neighborhood kernel:





The Cooperative Process

In many cases a good choice is Gaussian function:



The use of a Gaussian topological neighborhood also makes the SOM algorithm converge more quickly than a rectangular topological neighborhood would.

The Cooperative Process

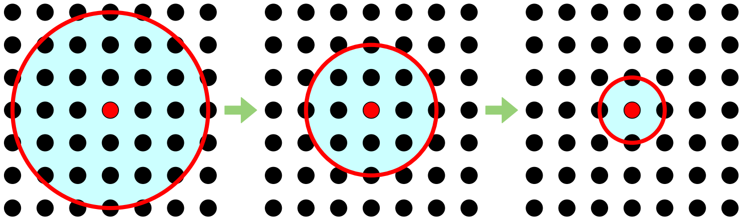
The size of the topological neighborhood can shrink with time by making the width of the topological neighborhood function $H_{j,i}$ decrease with time. The choice for the dependence of σ on discrete time t is the exponential decay described by:

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{\tau_\sigma}\right)$$

where σ_0 is the value of assumed σ at the initiation of the SOM algorithm and τ_σ is a assumed time constant. The topological neighborhood function assumes a time-varying due to decay form:

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

The Cooperative Process



The Adaptive Process

In the adaptive process, the outputs become self-organised and the feature map between inputs and outputs is formed:

- the winning neuron i gets its weights updated,
- the neighbors of winning neuron j (excited neurons) will have their weights updated as well, although by not as much as the winner itself.

We may then express the change to the weight vector of neuron in the lattice as:

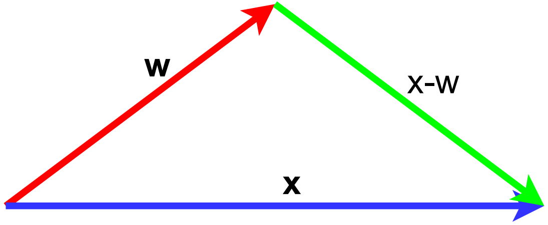
$$\Delta w_j = \eta H_{j,i}(x - w_j)$$

The discrete-time form:

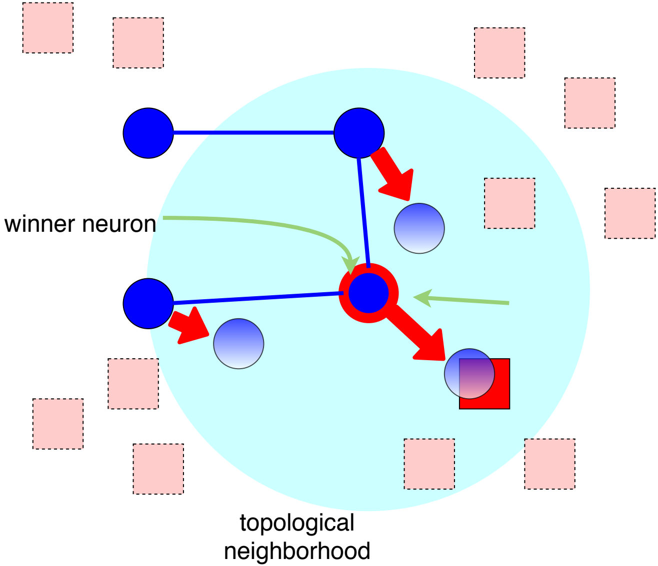
$$w_{j,i}(t+1) = w_{j,i}(t) + \eta(t) H_{j,i}(t)(x_i(t) - w_{j,i}(t))$$

The Adaptive Process

The maximization of the inner product $w_j^T x$ is mathematically equivalent to minimizing the distance between the vectors x and w_j.



The Adaptive Process



The Adaptive Process

The learning-rate parameter $\eta(t)$ should also be time varying.

- It should start at some initial value η_0 and then decrease gradually with increasing time t.
- This requirement can be satisfied by the following heuristic:

$$\eta(t) = \eta_0 \exp\left(-\frac{t}{\tau_\eta}\right)$$

where τ_η is assumed time constant.

Ordering and Convergence

There are two identifiable phases of this adaptive process:

- Self-organizing (ordering) phase – during which the topological ordering of the weight vectors takes place (can take 1000 iterations):
    - The learning-rate parameter $\eta(t)$ should begin with a value close to 0.1 and gradually decrease, but remain above 0.01.
    - The neighborhood function $H_{j,I(x)}$ should initially include almost all neurons in the network centered on the winning neuron $I(x)$ and then shrink slowly with time.

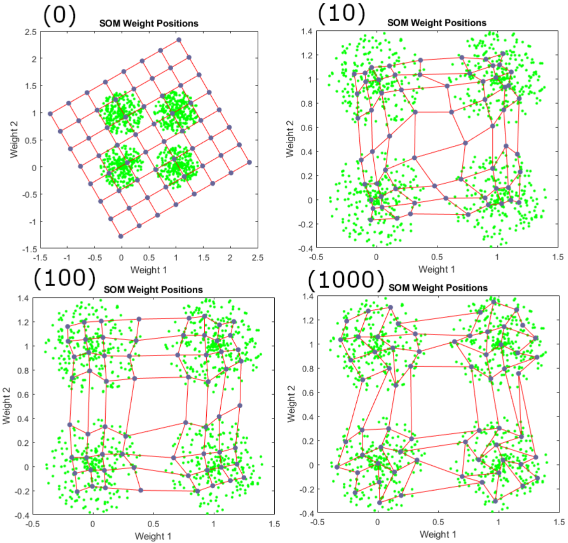
Ordering and Convergence

There are two identifiable phases of this adaptive process:

- Convergence phase – during which the feature map is fine tuned and comes to provide an accurate statistical quantification of the input space (can take at least 500 times the number of neurons):
    - For good statistical accuracy, the learning-rate parameter $\eta(t)$ should be maintained during the convergence phase at a small value, on the order of 0.01,
    - The neighborhood function $H_{j,I(x)}$ should contain only the nearest neighbors of a winning neuron, which may eventually reduce to one = zero neighboring neurons.

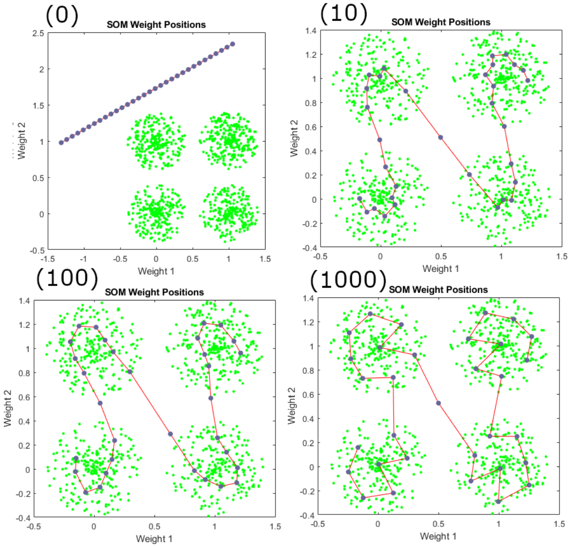
Results

Lattice:
2-dimensional
Input
vector:
2-dimensional.



Results

Lattice:
1-dimensional
Input
vector:
2-dimensional.



Summary of the SOM Algorithm

The stages of the SOM algorithm:

1. Initialization - Choose random values for the initial weight vectors $w_j(0)$ or select initial weight from input dataset. The only restriction here is that the weights should be different to each other.
2. Sampling - Draw a sample x from the input space with a certain probability. The vector x represents the activation pattern (input) that is applied to the lattice.
3. Matching - Find the best-matching (winning) neuron $i(x)$ at time-step t by using the minimum-distance criterion

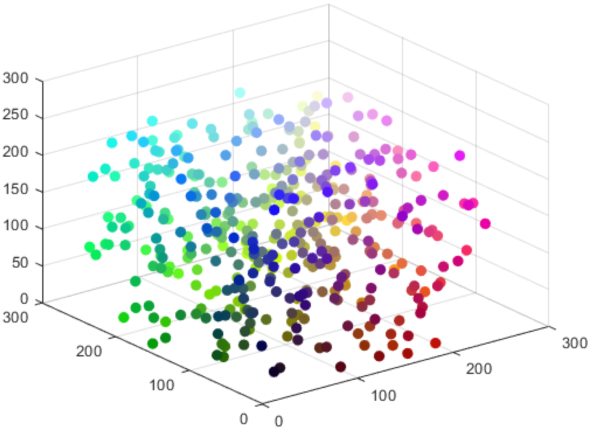
Summary of the SOM Algorithm

The stages of the SOM algorithm:

4. Updating - Apply the weight update equation to all excited neurons by using the update formula $\Delta w_{j,i}$, where learning-rate parameter η and neighborhood function H are varied dynamically during learning for best results.
5. Continuation - Continue with step 2 until no noticeable changes in the feature map are observed.

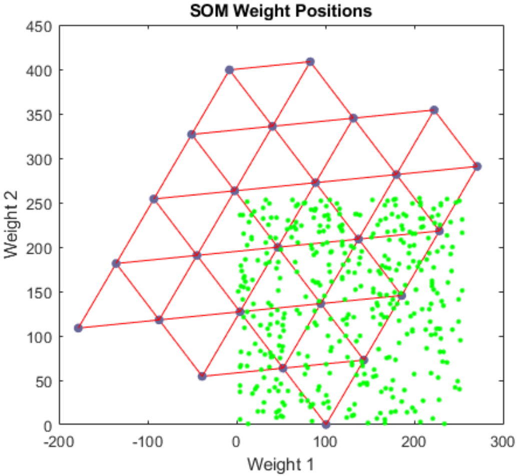
Results

Example - Color organization. Visualization of 400 color samples (RGB) on 5×5 lattice.



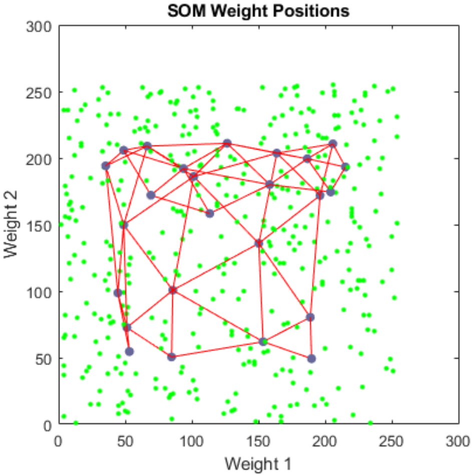
Results

Example - Color organization. Initial 5×5 lattice location (weights).



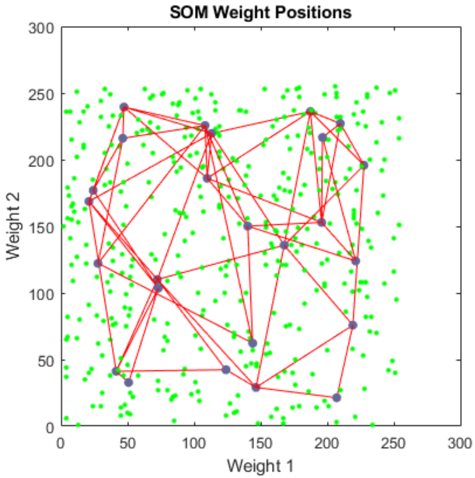
Results

Example - Color organization. Lattice location (weights) after 100 iteration.



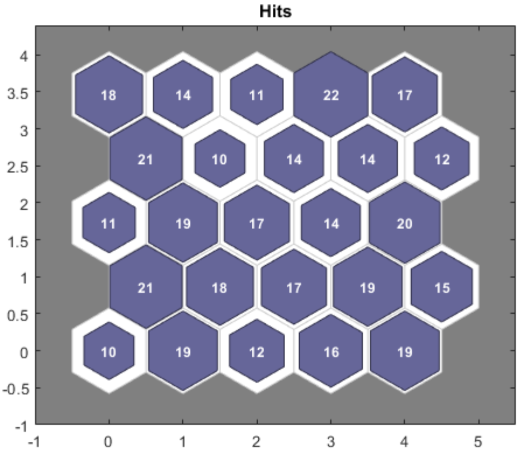
Results

Example - Color organization. Lattice location (weights) after 1000 iteration.



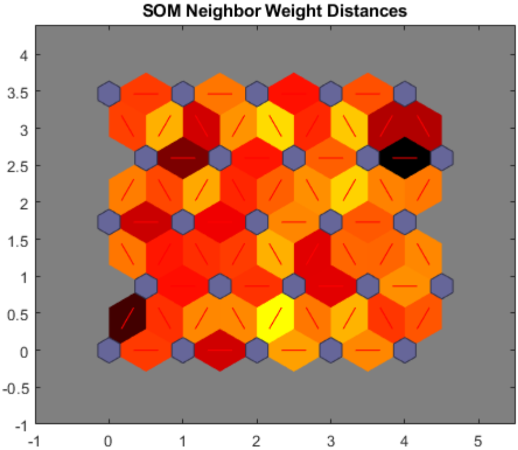
Results

Example - Color organization. Each SOM neuron showing the number of input vectors that it classifies.



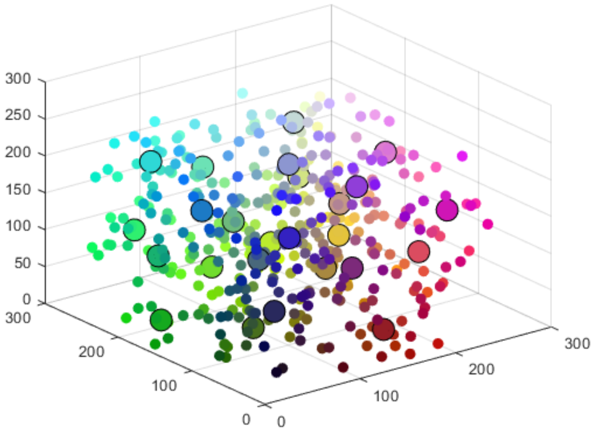
Results

Example - Color organization. The SOM layer showing neurons as gray-blue patches and their direct neighbor relations with red lines.



Results

Example - Color organization. Visualization of 400 color samples and 5×5 neuron centers.



Results

Example - Color organization. Output lattice with colors indicating neuron weights.

Density Matching

The feature map Φ reflects variations in the statistics of the input distribution:

- regions in the input space from which the sample training vectors x are drawn with high probability of occurrence are mapped onto larger domains of the output space,
- therefore with better resolution than regions of input space from which training vectors are drawn with low probability.

Density Matching

We need to relate the input vector probability distribution $p(x)$ to the magnification factor $m(x)$ of the feature map.

- Generally, for two dimensional feature maps the relation cannot be expressed as a simple function, but in one dimension we can show that

$$m(x \to \inf) \approx p^{2/3}(x)$$

  So the SOM algorithm doesn't match the input density exactly, because of the power of $2/3$ rather than 1.
- Computer simulations indicate similar approximate density matching in general, always with the low input density regions slightly over-represented.

Questions