Clustering (Unsupervised)

Self Organizing Maps (SOM)

CS 6316 – Machine Learning
Fall 2017

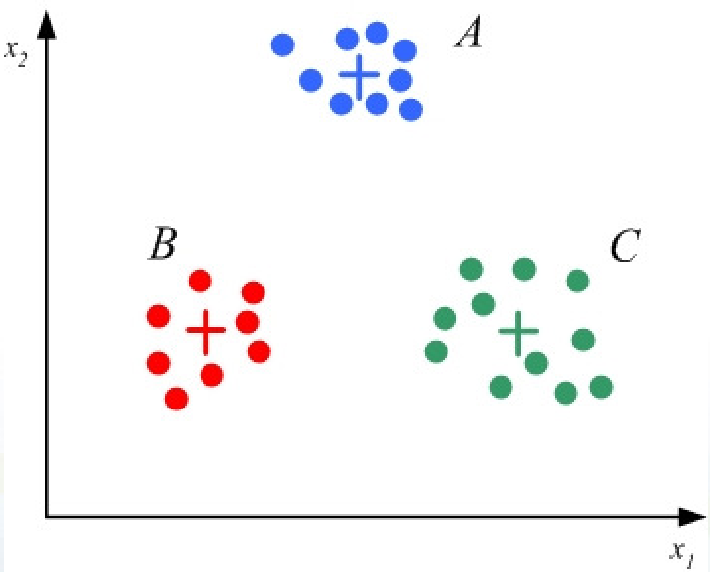
OUTLINE

- Recall Unsupervised Learning
- Clustering and Vector Quantization
- Self-Organizing Maps (SOM)

Recall Unsupervised Learning

- Recall from Ch. 2
  - Unsupervised learning
  - Data reduction approach

- Example: training data represented by three (3) "centers"



Two Types of Problems

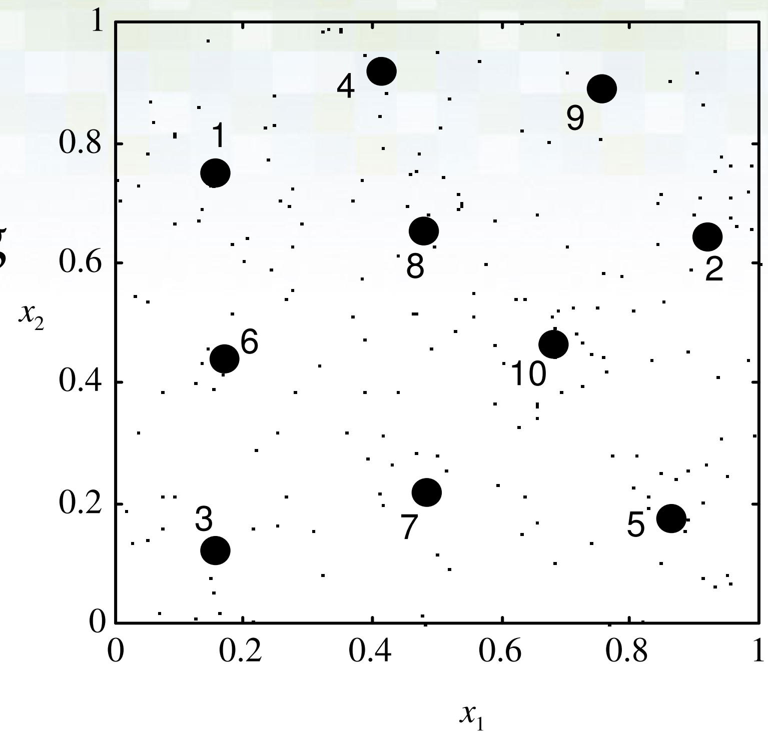
(1) Data reduction

Vector Quantization (VQ)+Clustering

"Model" ~ m points

Vector Quantizer Q:

$$f(\mathbf{x},\omega) = Q(\mathbf{x}) = \sum_{j=1}^{m} \mathbf{c}_j I(\mathbf{x} \in \mathbf{R}_j)$$



VQ setting: given n training samples $X=\{x_1, x_2, \ldots, x_n\}$

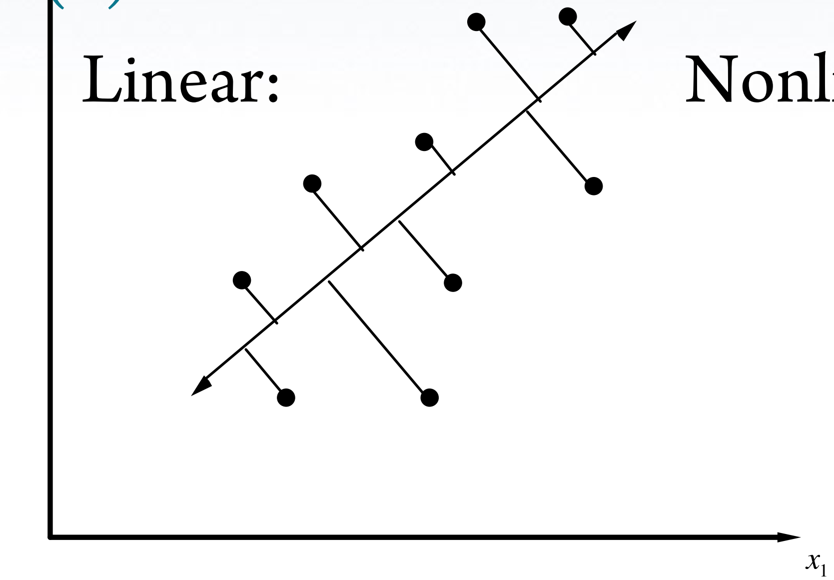
Find the coordinates $\mathbf{c}_j$ of m centers (prototypes) such that the total squared error distortion is minimized

$$R(\omega) = \int \|\mathbf{x} - f(\mathbf{x},\omega)\|^2 p(\mathbf{x}) d\mathbf{x}$$

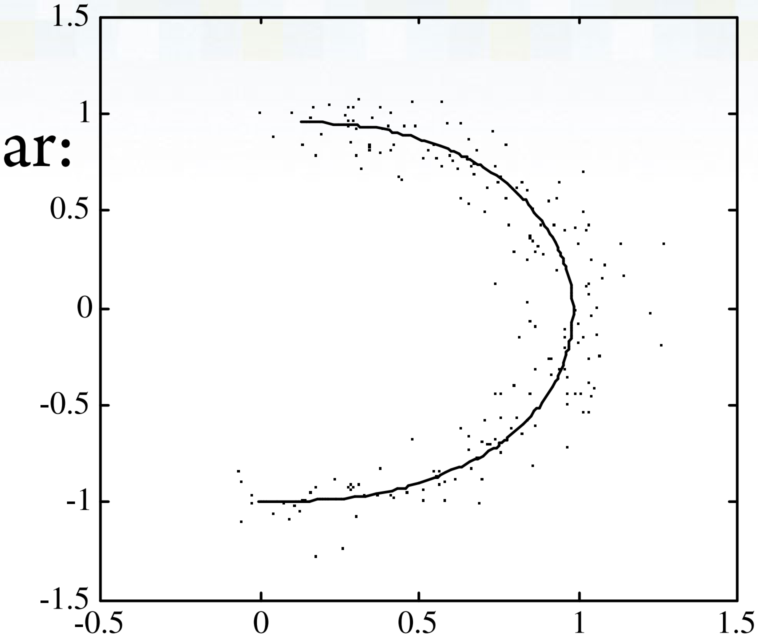
Two Types of Problems

(2) Data reduction

Linear:



Nonlinear:



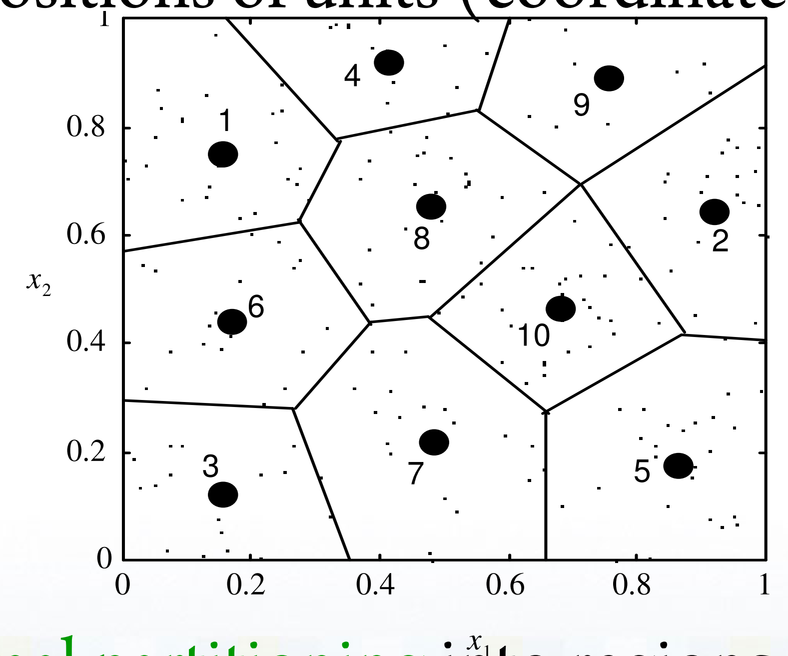
"Model" ~ projection of high-dim data onto low-dim space

Note: the goal is to estimate a mapping from d-dimensional input space (d=2) to low-dimensional feature space

$$R(\omega) = \int \|\mathbf{x} - f(\mathbf{x}, \omega)\|^2 \, p(\mathbf{x}) d\mathbf{x}$$

Vector Quantization and Clustering

- Two complementary goals of VQ:
  1. Partition the input space into disjoint regions
  2. Find positions of units (coordinates of prototypes)



- Note: optimal partitioning into regions
  - Nearest-neighbor rule (Voronoi regions)

Generalized Lloyd Algorithm (GLA) for VQ

Given data points $\mathbf{x}(k) \quad k = 1, 2, \ldots$, loss function L (i.e., squared loss) and initial centers $\mathbf{c}_j(0) \quad j = 1, \ldots, m$

Perform the following updates upon presentation of $\mathbf{x}(k)$

1. **Find the nearest center** to the data point (the winning unit):

$$j = \underset{i}{\boldsymbol{arg\,min}} \left\| \mathbf{x}(k) - \mathbf{c}_i(k) \right\|$$

2. **Update the winning unit** coordinates (only) via

$$\mathbf{c}_j(k+1) = \mathbf{c}_j(k) + \gamma(k)\left[\mathbf{x}(k) - \mathbf{c}_j(k)\right]$$

Increment k and iterate steps (1) – (2) above

Note: - the learning rate decreases with iteration number k

- biological interpretations of steps (1)-(2) exist

Batch version of GLA

Given data points $\mathbf{x}_i \quad i=1,\ldots,n$, loss function L (i.e., squared loss) and initial centers $\mathbf{c}_j(0) \quad j=1,\ldots,m$

Iterate the following two steps

1. **Partition the data** (assign sample $\mathbf{x}_i$ to unit j) using the nearest neighbor rule. Partitioning matrix Q:

$$q_{ij} = \begin{cases} 1 \text{ if } L(\mathbf{x}_i, \mathbf{c}_j(k)) = \min\limits_{l} L(\mathbf{x}_i, \mathbf{c}_l(k)) \\ 0 \text{ otherwise} \end{cases}$$

2. **Update unit coordinates** as centroids of the data:

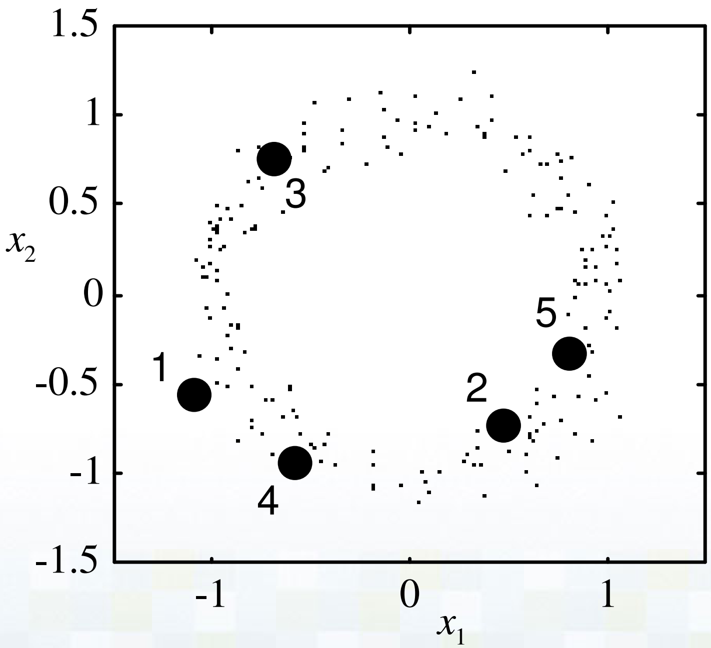
$$\mathbf{c}_j(k+1) = \frac{\sum_{i=1}^{n} q_{ij}\mathbf{x}_i}{\sum_{i=1}^{n} q_{ij}}, \; j = 1,\ldots,m$$

Note: final solution may depend on initialization (local min) – potential problem for both on-line and batch GLA

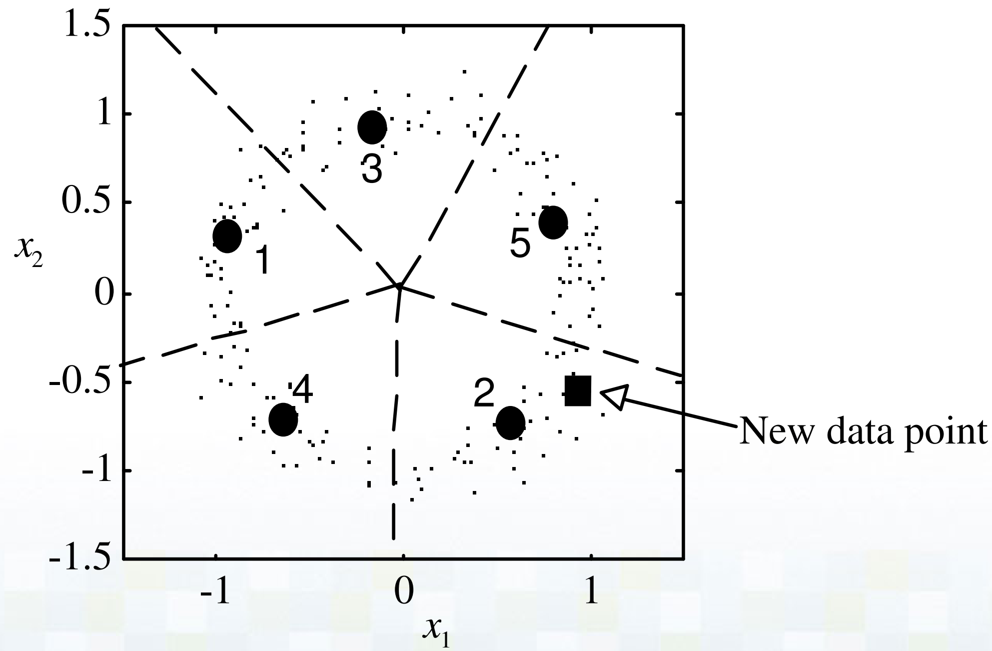
GLA Example 1

- Modeling doughnut distribution using 5 units

(a) initialization

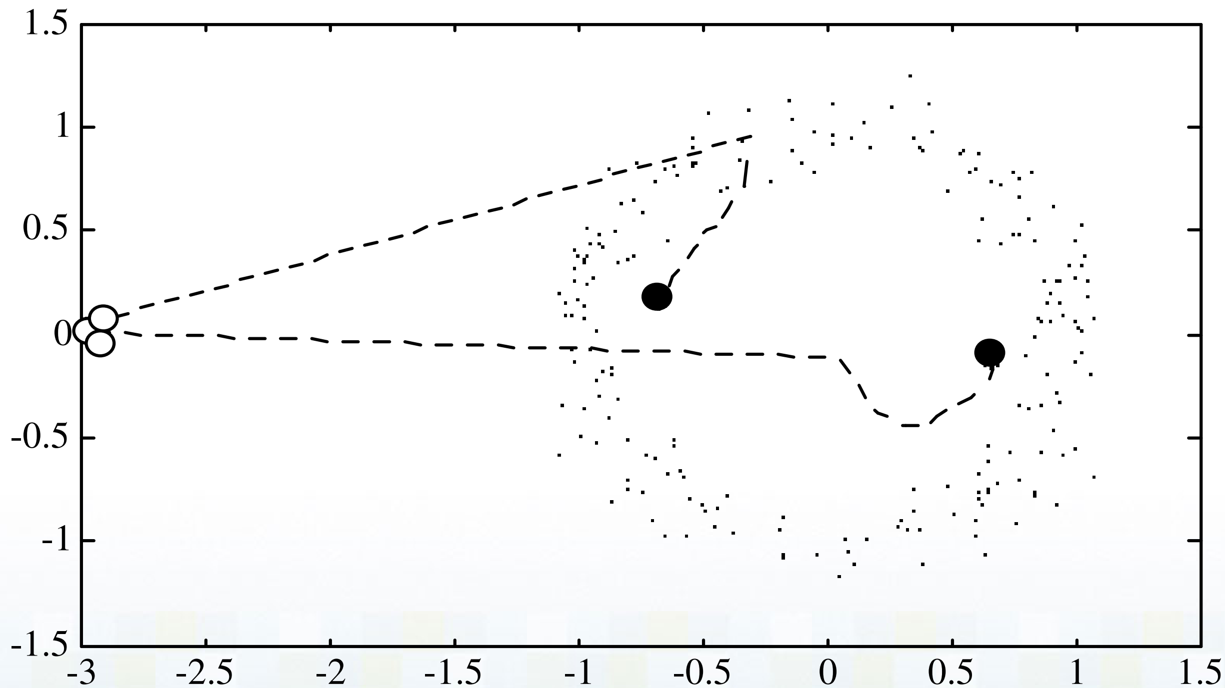


(b) final position (of units)



GLA Example 2

- Modeling doughnut distribution using 3 units:
  Bad initialization → poor local minimum



Avoiding Local Minima

- Starting with many random initializations
  - Then choosing best GLA solution

… there are others ways

… but one method is to employ the use of

- Self-Organizing Map:
  - Introduce topological relationship (map), thus forcing the *neighbors* of the winning unit to move towards the data

Clustering Methods

- Clustering: separating a data set into several groups (clusters) according to some measure of similarity
- Goals of clustering:
  - interpretation (of resulting clusters)
  - exploratory data analysis
  - preprocessing for supervised learning
  - often the goal is not formally stated
- VQ-style methods (GLA) often used for clustering, i.e. k-means or c-means
- Many other clustering methods as well

Clustering

- Clustering:
  - Partitioning a set of n objects (samples) into k disjoint groups, based on some similarity measure.

    Assumptions:
    - Similarity: distance metric dist(i, j)
    - Usually k given a priori (but not always!)
- Intuitive Motivation:
  - Similar objects ~ into one cluster
  - Dissimilar objects ~ into different clusters
  - Distance needs to be defined for different types of input

Self-Organizing Map

Specifically ~ "Kohonen Network"

History of Kohonen SOM

- Developed in 1982 by Tuevo Kohonen, a professor of the Academy of Finland

- Professor Kohonen worked on auto-associative memory during the 70s and 80s and in 1982 he presented his self-organizing map algorithm

Biological Motivation of SOM

- Brain changes its internal structure to reflect life experiences
  - Interaction with environment is critical at early stages of brain development (first 1-2 years of life)

- Existence of various regions (maps) in the brain

- How are these maps formed? i.e. information-processing model leading to map formation

  → Kohonen proposed SOM

What is a Self-Organizing Map?

- A kind of unsupervised training
  - In which networks learn to form their own classifications of the training data without extra help

- In order for the network to be unsupervised, we have to assume class membership is broadly defined by the input patterns sharing *common features*
  - The network will be able to identify those features across the range of input patterns
  - A SOM learns to classify the training data without any external supervision – thus requiring no target vector

What is a Self-Organizing Map?

- One interesting class of **unsupervised** systems is based on *competitive learning*
  - Output neurons compete amongst themselves to be activated: only one is activated at any given time
  - Activated neuron: "winning neuron"
  - Such competition can be induced/implemented by having ***lateral inhibition connections*** (negative feedback paths) between the neurons
  - The result is that neurons are forced to organize themselves
  - Therefore, such a network is called a **Self-Organizing Map**

What is a Self-Organizing Map?

- SOMs attempt to "*map*" their weights to conform to the given input data

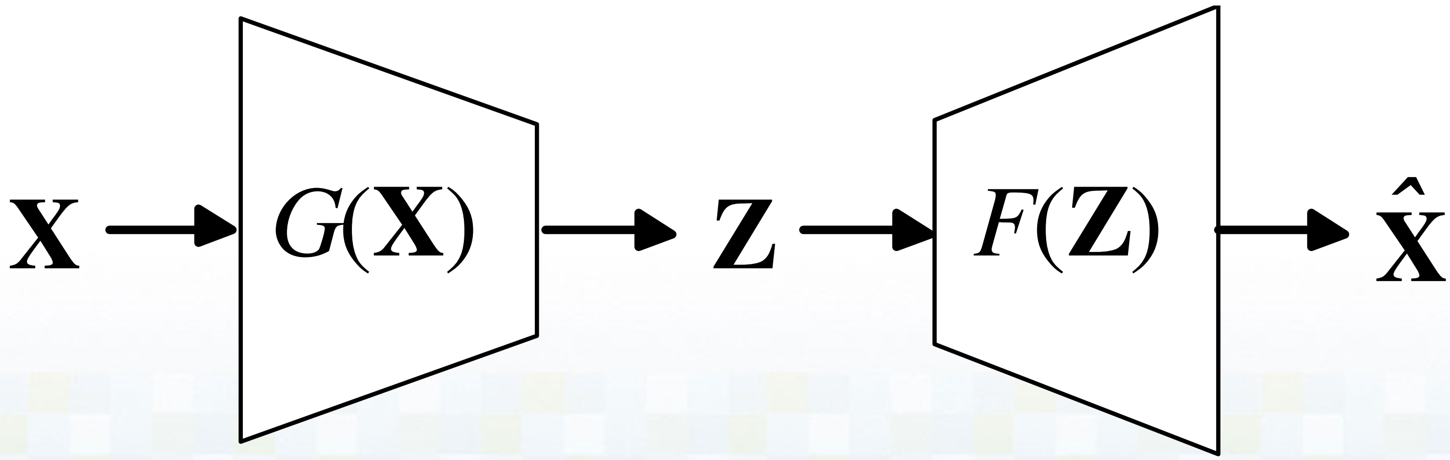
- *Thus SOMs are neural networks that employ unsupervised learning methods, mapping their weights to conform to the given input data with a goal of representing multidimensional data in an easier and understandable form for the human eye. (pragmatic value of representing complex data)*

A Goal of SOM

Dimensionality reduction!!

HERE'S ANOTHER TECHNIQUE TO CONVERT

HIGH DIMENSIONAL DATA INTO A LOWER DIMENSIONAL SPACE



Goal of SOM

- Dimensionality reduction
  - Project given (high-dim) data onto low-dim space (called a map)
  - Feature space (Z-space) is 1D or 2D
    - Is discretized as a number of units, e.g. 10x10 map
  - Z-space has distance metric → ordering of units

- Transform incoming signal pattern of arbitrary dimension into a 1 or 2 dim discrete map
  - And to perform this transformation adaptively in a topologically ordered fashion

Topographic Maps

- Neurobiological studies indicate that different sensory inputs (motor, visual, auditory, etc.) are mapped onto corresponding areas of the cerebral cortex in an *orderly fashion*

- Our interest is in building artificial topographic maps that learn through self-organization in a neurobiologically inspired manner

Topographic Maps

- This form of **map**, known as a topographic map, has two important properties:

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

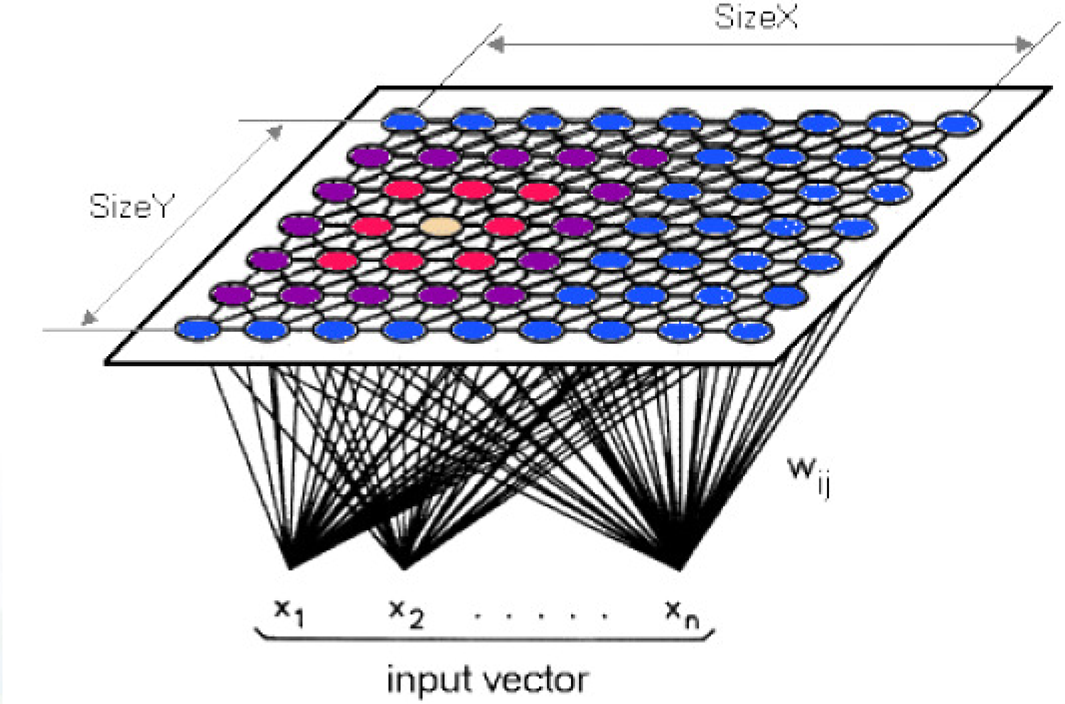
Topographic Maps

- We shall follow the ***principle of topographic map formation***:

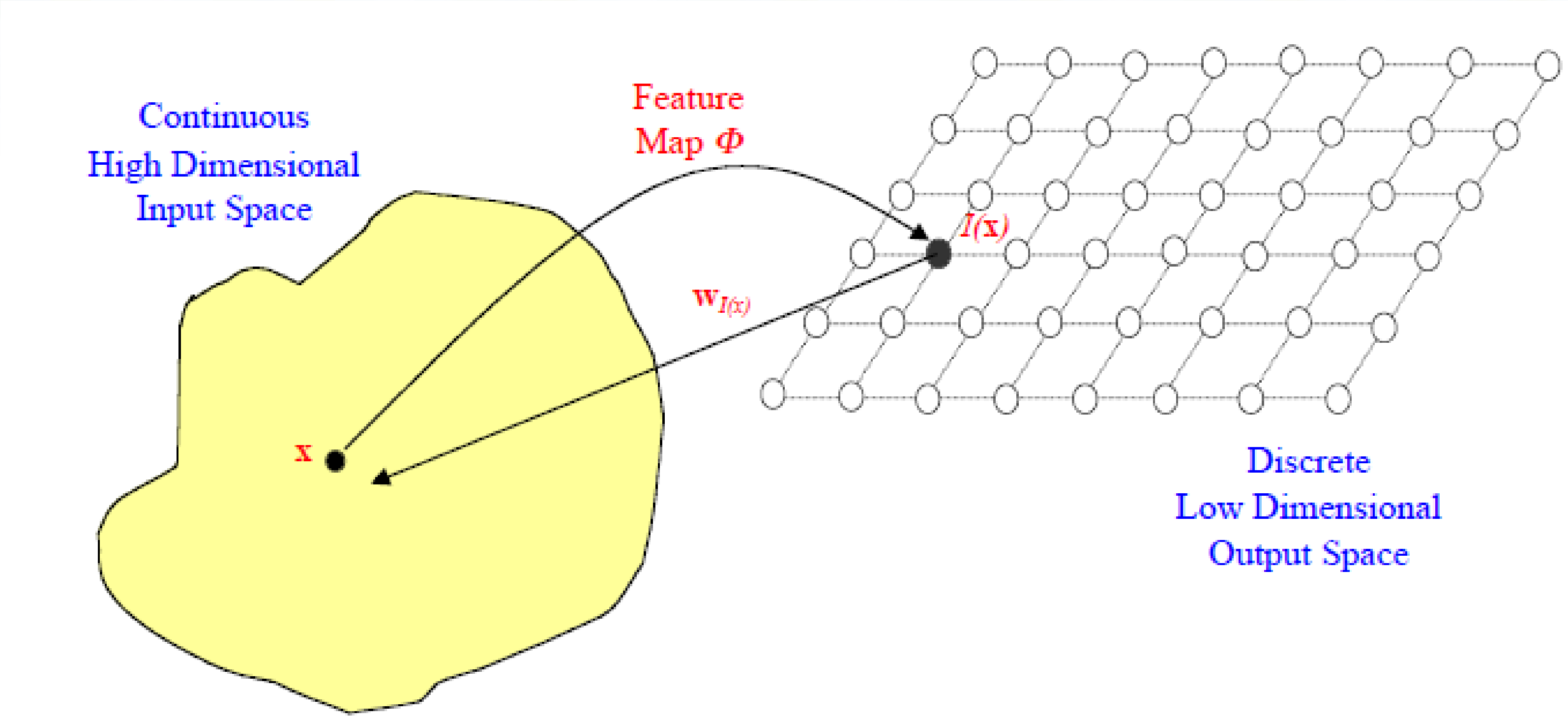
"The spatial location of an output neuron in a topographic map corresponds to a particular domain or feature drawn from the input space"

The Architecture

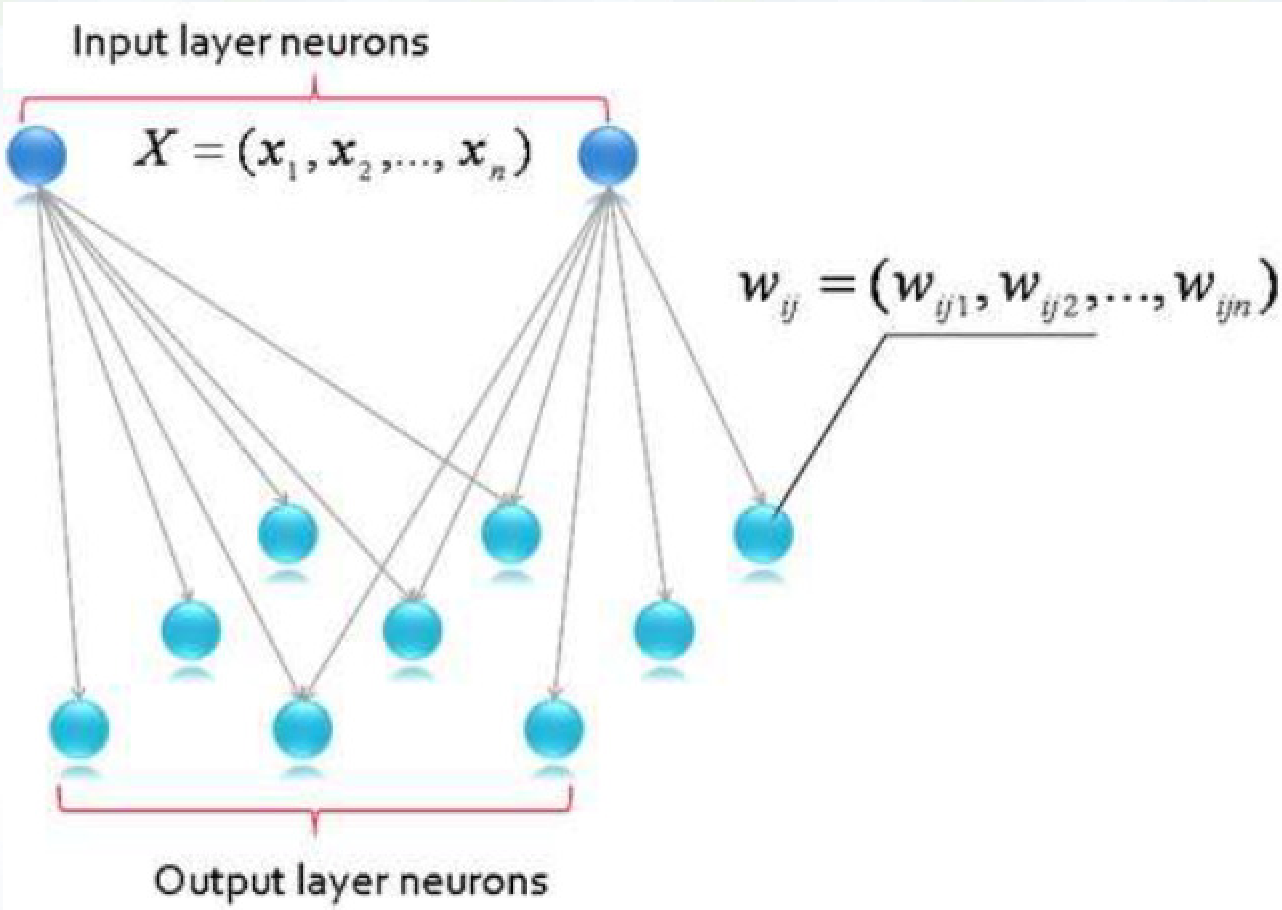
- Made up of input nodes and computational nodes "*neurons*"
- Each computational node is connected to each input node to form a lattice



- We have points **x** in the input space mapping to points $I(\mathbf{x})$ in the output space:



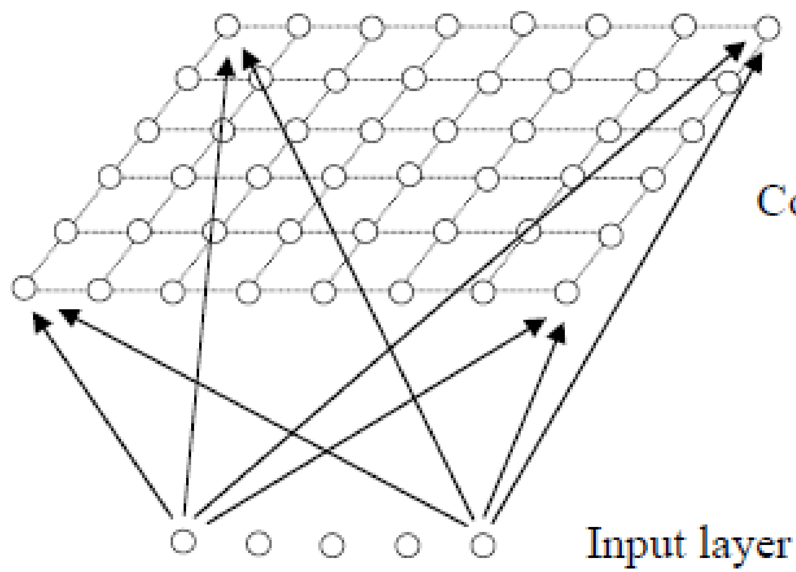
- Each point I in the output space will map to a corresponding point s(I) in the input space



Input layer neurons
$X = (x_1, x_2, \ldots, x_n)$
$w_{ij} = (w_{ij1}, w_{ij2}, \ldots, w_{ijn})$
Output layer neurons

Kohonen Networks

SOM has a feed-forward structure with a single computational layer arranged in rows and columns. Each neuron is fully connected to all the source nodes in the input layer:



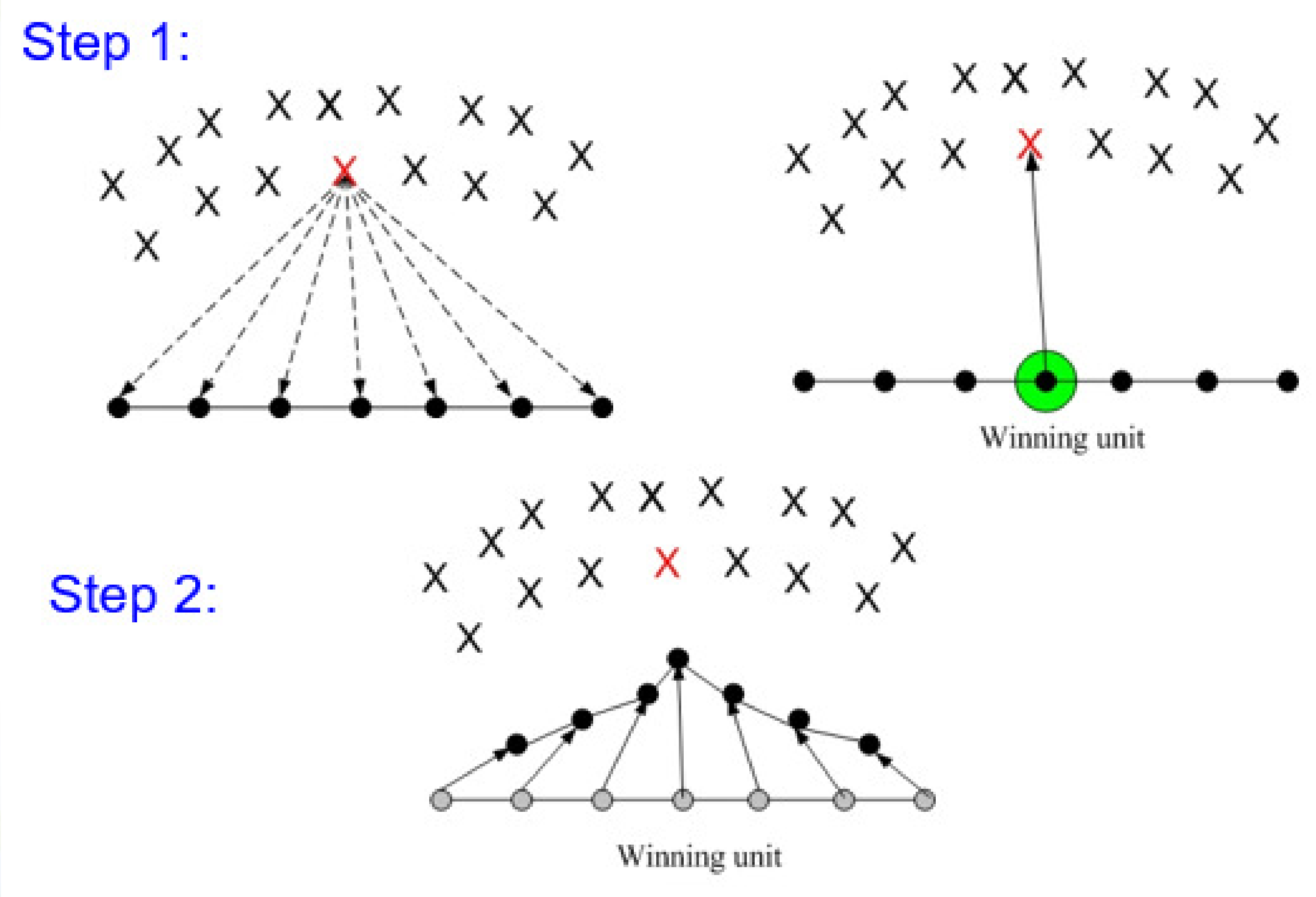
Clearly, a 1-dim map will just have a single row (or col) in the computational layer

Self-Organizing Map

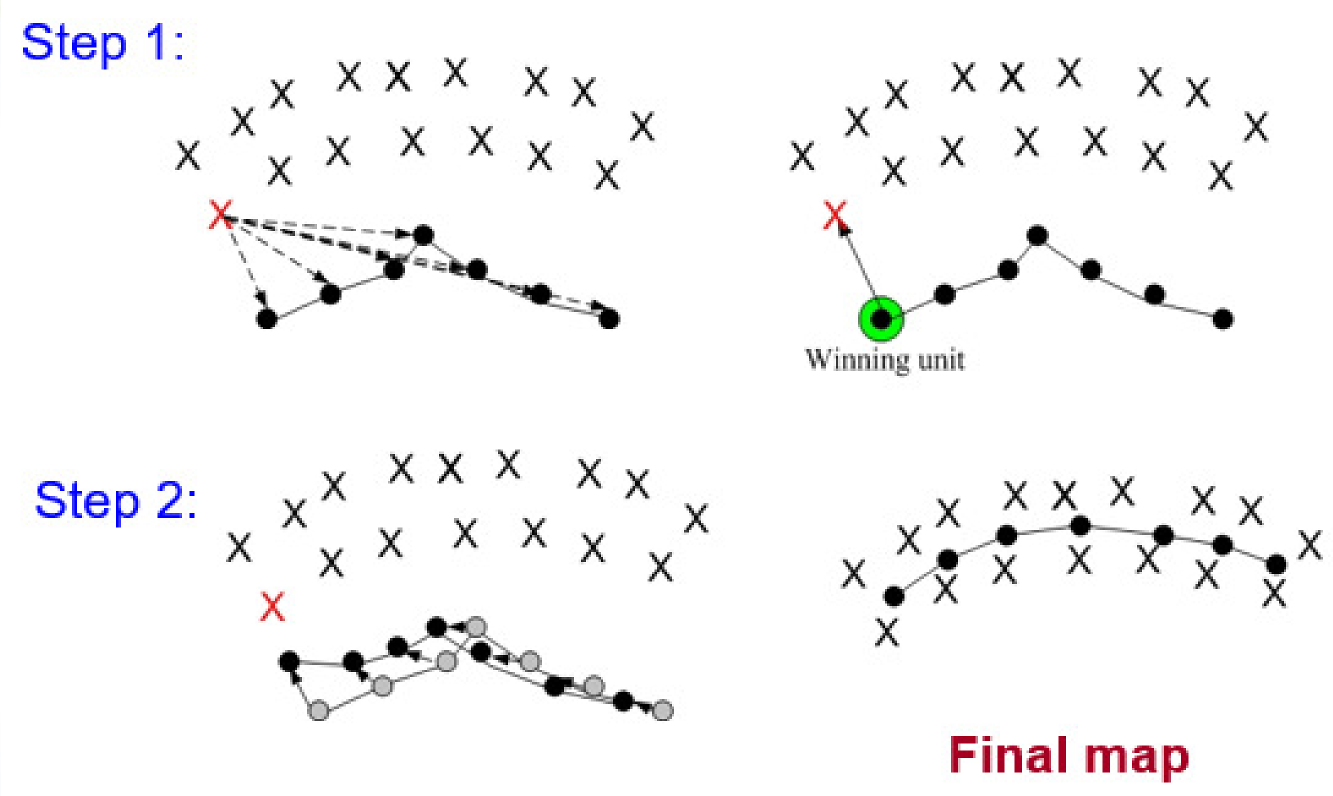
- Place neurons at the nodes of a 1 or 2 dimensional lattice
- The neurons become selectively tuned to various input patterns (stimuli) or classes of input patterns during the course of the competitive learning

- The locations of the neurons so tuned (i.e. the winning neurons) become ordered and a meaningful ***coordinate system*** for the ***input features*** is created on the lattice
  - The SOM thus forms the required topographic map of the input patterns
  - **CAN VIEW THIS AS A NON-LINEAR GENERALIZATION OF PCA!**

SOM example (one iteration)



SOM example (next iteration)



SOM Algorithm (in short)

1. Each node's weights are initialized
2. A vector is chosen at random from the set of training data and presented to the network
3. Every node in the network (neuron) is examined to calculate which ones' weights are most like the input vector. The winning node: the "Best Matching Unit" (BMU)
4. The radius of the neighborhood of the BMU is calculated (radius diminishes each time-step)
5. Any nodes (neurons) found within the radius of BMU (#4) are adjusted to make them more like the input vector
   – Closer a neuron is to the BMU, more its weights are altered
6. Repeat starting at #2 for N iterations

Components of Self Organization [Four Major Components]

1. INITIALIZATION
2. COMPETITION
3. COOPERATION
4. ADAPTATION

Components of Self Organization
[Four Major Components]

- INITIALIZATION
  - All the connection weights are initialized with small random values

- COMPETITION
  - For each input pattern, the neurons compute their respective values of a *discriminant function* which provides the basis for competition
  - The particular neuron with the smallest value of the discriminant function is declared the winner

Components of Self Organization
[Four Major Components]

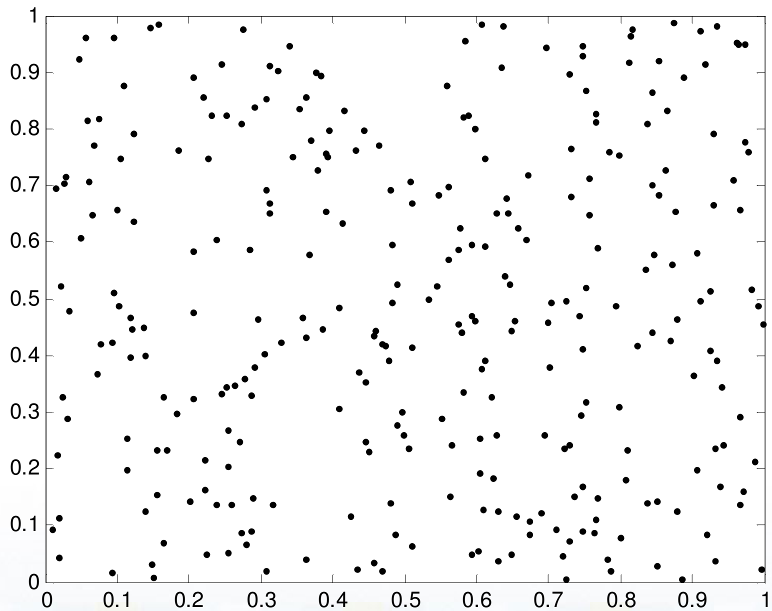
- COOPERATION
  - The winning neuron determines the spatial location of a topological neighborhood of excited neurons
  - Thereby providing the basis for cooperation among neighboring neurons

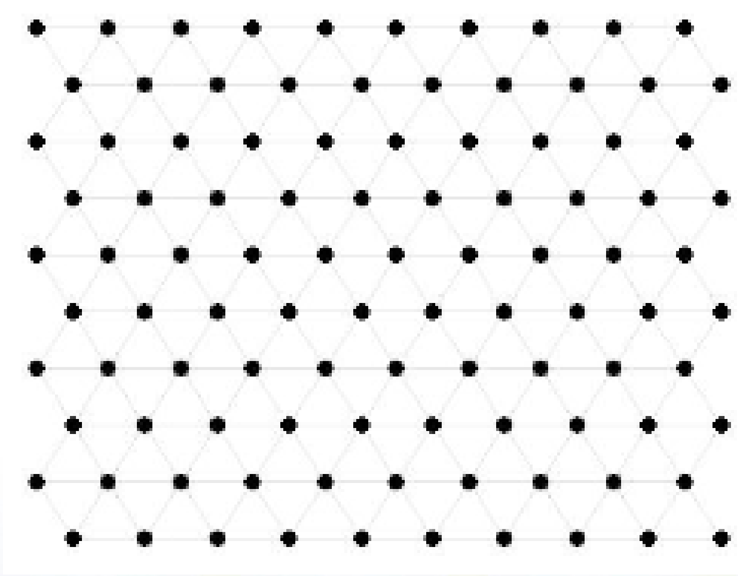
Components of Self Organization
[Four Major Components]

- 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

Does SOM work with Uniform Distribution??

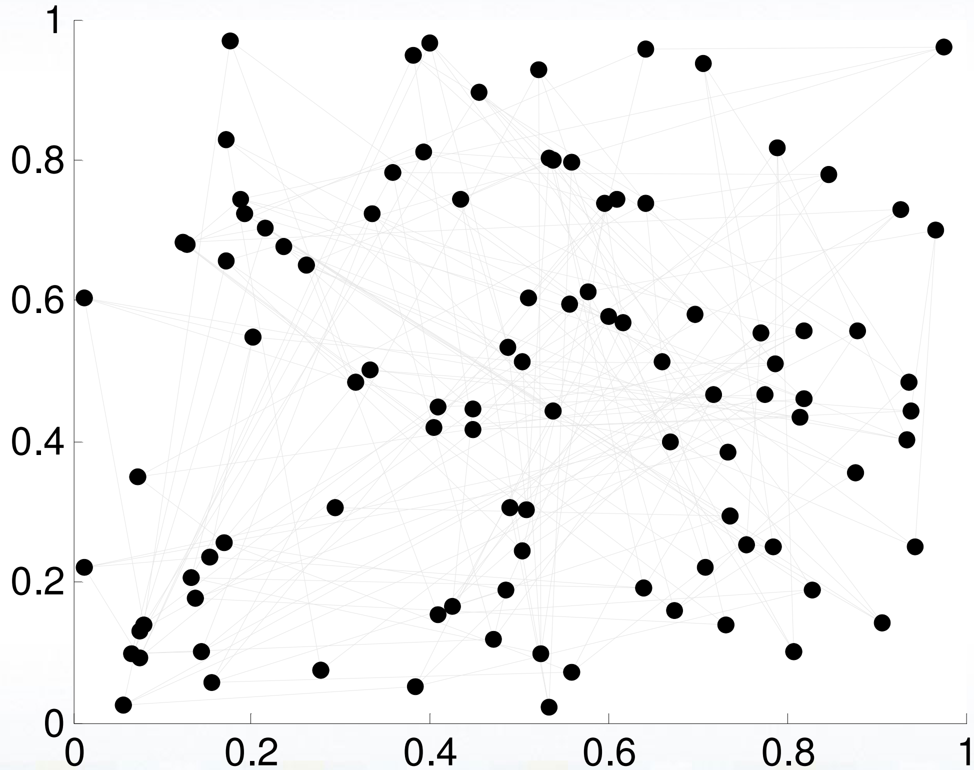
- Let's start with the training data that consists of 300 training samples uniformly distributed in [0,1] square as shown:



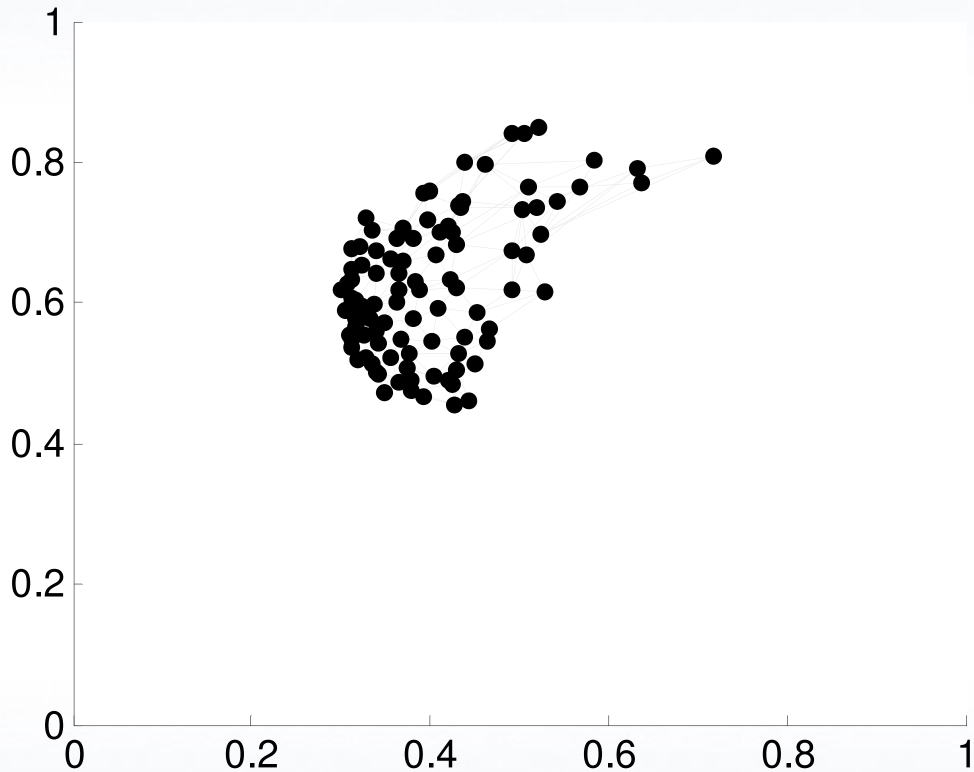


- The map uses 2D 10x10 grid topology shown on the right

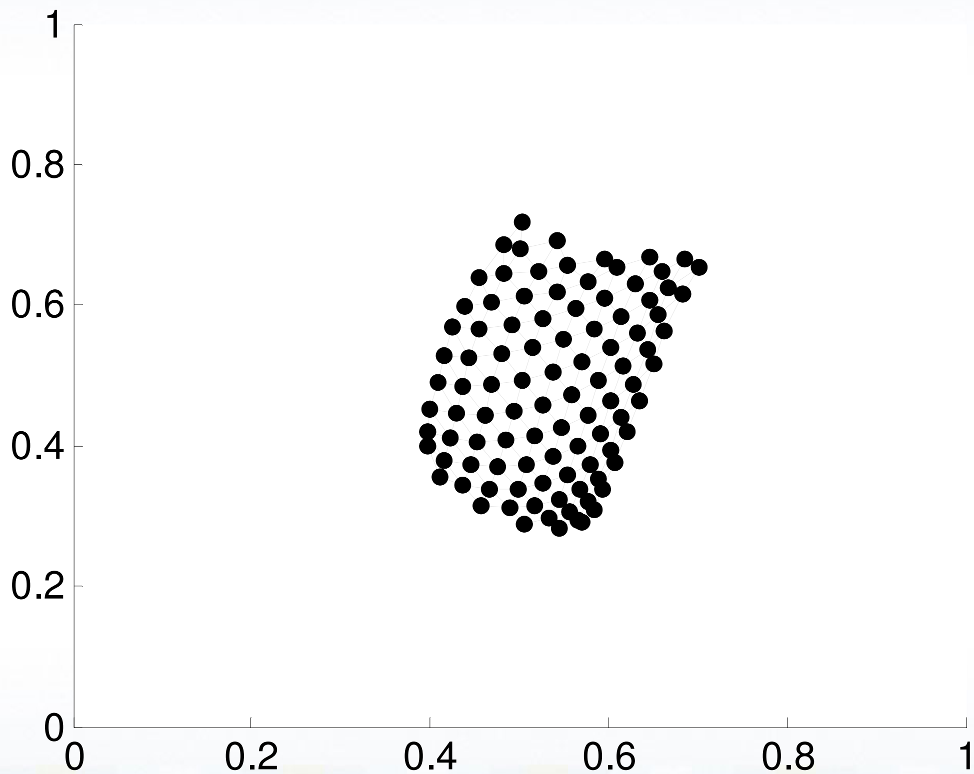
Initially neurons are placed randomly in [0,1] square



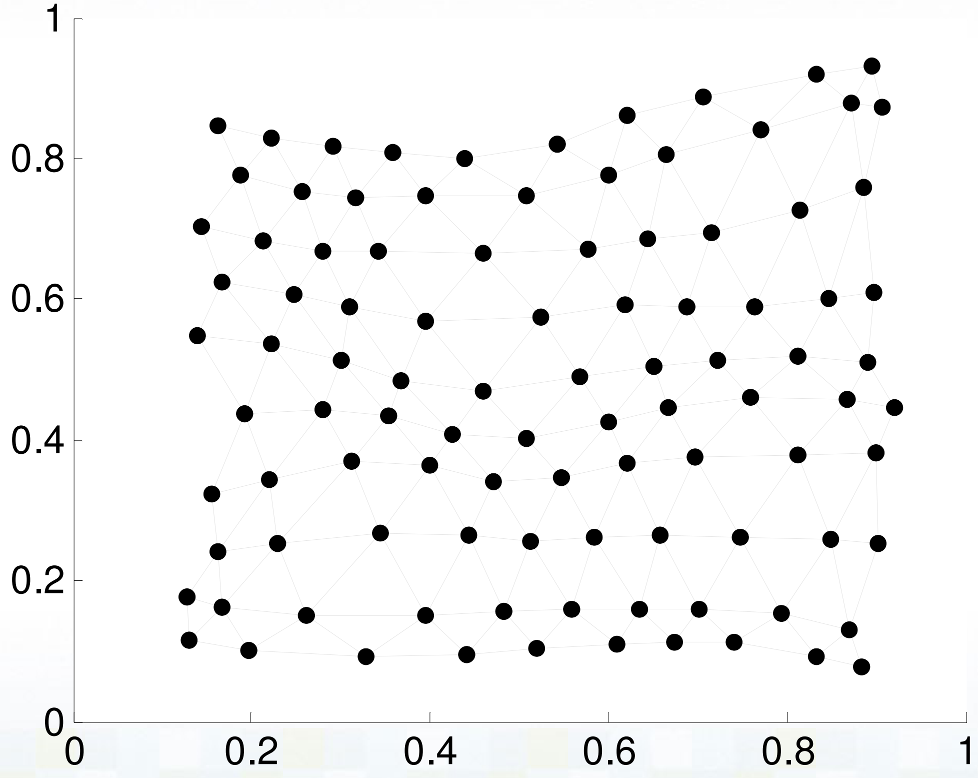
After 50 iterations: the units are clustered in the center area of the square



After 100 iterations: at this point, one can see that the map units start to exhibit correct *topological ordering* that reflect an underlying distribution



After 10,000 iterations [final SOM model]: provides a faithful approximation of the uniform distribution



SOM Applications

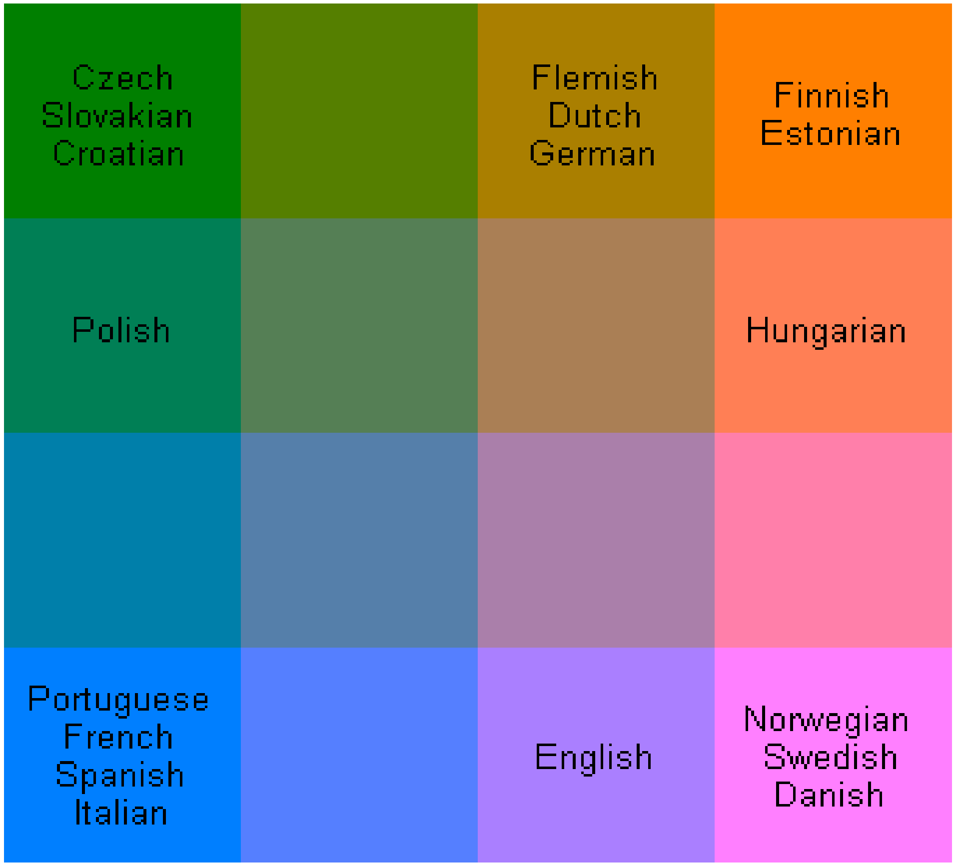
- Two types of applications:
  - Vector Quantization
  - Clustering of multivariate data
- SOM Research: http://www.cis.hut.fi/research/som-research
- Numerous Applications:
  - Marketing surveys / segmentation
  - Financial / stock market data
  - Text data / document map ~ WEBSOM
  - Image data / picture map ~ PicSOM

Example: Clustering European Languages

- Modeling approach: 18 European languages. Each language ~ 10 word set
  - If same alphabet, possible to perform clustering using some distance measure

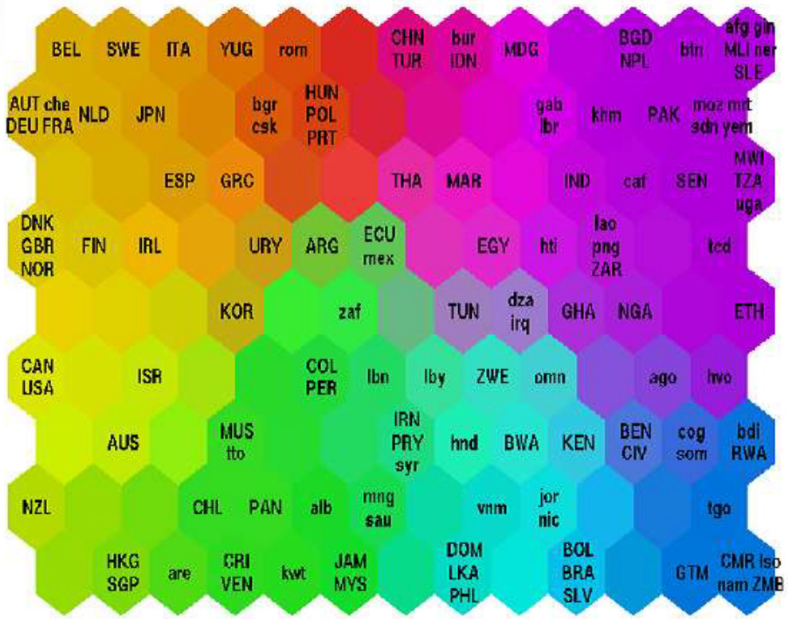
English	Norwegian	Polish	Czech	Slovakian	Flemish	Croatian	Portuguese	French	Spanish	Italian	Swedish	Danish	Finnish	Estonian	Dutch	German	Hungarian
one	en	jeden	jeden	jeden	ien	jedan	um	un	uno	uno	en	en	yksi	uks	een	erins	egy
two	to	dwa	dva	dva	twie	dva	dois	deux	dos	due	tva	to	kaksi	kaks	twee	zwei	ketto
three	tre	trzy	tri	tri	drie	tri	tres	trois	tres	tre	tre	tre	kolme	kolme	drie	drie	harom
four	fire	cztery	ctyri	styri	viere	cetiri	quarto	quatre	cuatro	quattro	fyra	fire	nelja	neli	vier	vier	negy
five	fem	piec	pet	pat	vuvve	pet	cinco	cinq	cinco	cinque	fem	fem	viisi	viis	vijf	funf	ot
six	seks	szesc	sest	sest	zesse	sest	seis	six	seis	sei	sex	seks	kuusi	kuus	zes	sechs	hat
seven	sju	sediem	sedm	sedem	zevne	sedam	sete	sept	siete	sette	sju	syv	seitseman	seitse	zeven	sieben	het
eight	atte	osiem	osm	osem	achte	osam	oito	huit	ocho	otto	atta	otte	kahdeksan	kaheksa	acht	acht	nyolc
nine	ni	dziewiec	devet	devat	negne	devet	nove	neuf	nueve	nove	nio	ni	yhdeksan	uheksa	negen	neun	kilenc
ten	ti	dziesiec	deset	desat	tiene	deset	dez	dix	dies	dieci	tio	ti	kymmenen	kumme	tien	zehn	tiz

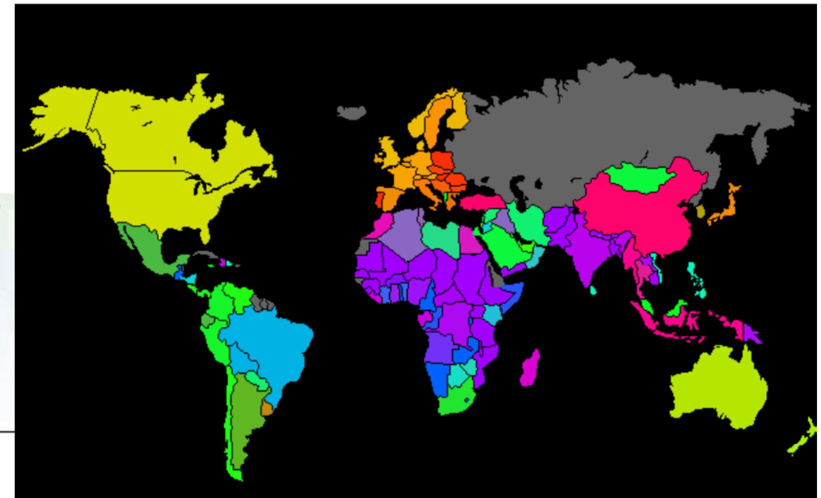
2D SOM (70 iterations)



- See section 6.7 in textbook for further details about "Similarity between European Languages"

SOM Example: Classify World Poverty





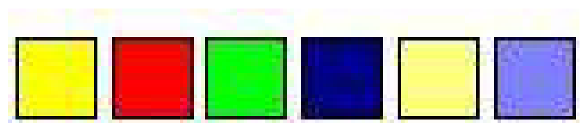
The Country Names

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		

Example: Color Classifier

- Problem:
  - Group and represent the primary colors and their corresponding shades on a two dimensional plane

- Colors are represented in their RGB values to form 3-dimensional vectors

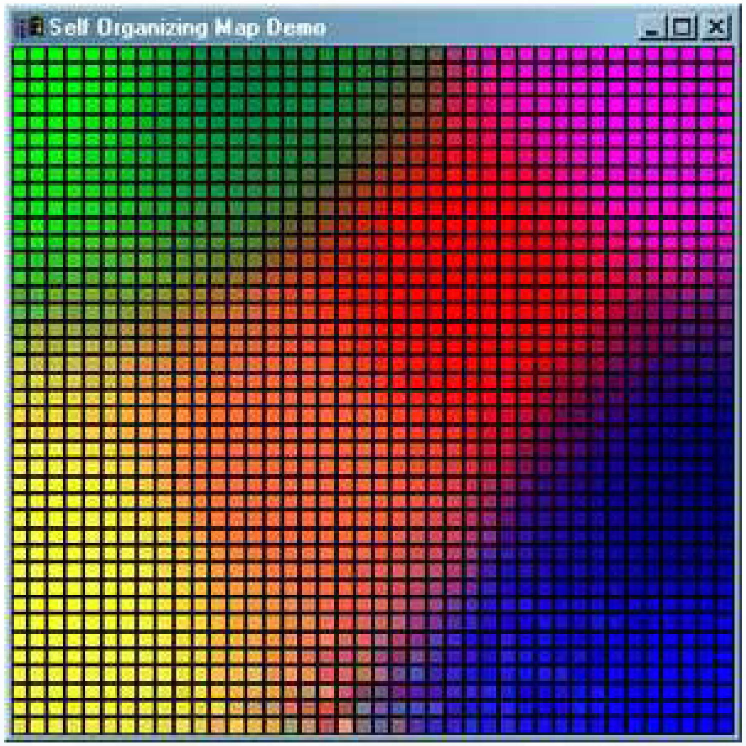


- Assume the RGB values are represented by the values 0 – 6 depending on their intensity

  i.e. Red = (6, 0, 0); Green = (0, 6, 0); Blue = (0, 0, 6)

Color Classifier: Example Output

- Classification on 40x40 SOM



Additional Examples

Clustering of European Languages

Background: historical linguistics studies relatedness btwn languages based on

phonology, morphology, syntax and lexicon

Difficulty of the problem: due to evolving nature of human languages and globalization.

Hypothesis: similarity based on analysis of a small 'stable' word set.

See glottochronology, Swadesh list, at

http://en.wikipedia.org/wiki/Glottochronology

SOM Clustering of European Languages

Modeling approach: language ~ 10 word set.

Assuming words in different languages are encoded in *the same alphabet*, it is possible to perform clustering using some distance measure.

Issues:

- selection of a *stable word set*
- data encoding + distance metric

Stable word set: numbers 1 to 10

Data encoding: Latin alphabet, use 3 first letters (in each word)

Numbers word set in 18 European languages

Each language is a feature vector encoding 10 words

English	Norwegian	Polish	Czech	Slovakian	Flemish	Croatian	Portuguese	French	Spanish	Italian	Swedish	Danish	Finnish	Estonian	Dutch	German	Hungarian
one	en	jeden	jeden	jeden	ien	jedan	um	un	uno	uno	en	en	yksi	uks	een	erins	egy
two	to	dwa	dva	dva	twie	dva	dois	deux	dos	due	tva	to	kaksi	kaks	twee	zwei	ketto
three	tre	trzy	tri	tri	drie	tri	tres	trois	tres	tre	tre	tre	kolme	kolme	drie	drie	harom
four	fire	cztery	ctyri	styri	viere	cetiri	quarto	quatre	cuatro	quattro	fyra	fire	nelja	neli	vier	vier	negy
five	fem	piec	pet	pat	vuvve	pet	cinco	cinq	cinco	cinque	fem	fem	viisi	viis	vijf	funf	ot
six	seks	szesc	sest	sest	zesse	sest	seis	six	seis	sei	sex	seks	kuusi	kuus	zes	sechs	hat
seven	sju	sediem	sedm	sedem	zevne	sedam	sete	sept	siete	sette	sju	syv	seitseman	seitse	zeven	sieben	het
eight	atte	osiem	osm	osem	achte	osam	oito	huit	ocho	otto	atta	otte	kahdeksan	kaheksa	acht	acht	nyolc
nine	ni	dziewiec	devet	devat	negne	devet	nove	neuf	nueve	nove	nio	ni	yhdeksan	uheksa	negen	neun	kilenc
ten	ti	dziesiec	deset	desat	tiene	deset	dez	dix	dies	dieci	tio	ti	kymmenen	kumme	tien	zehn	tiz

Data Encoding

Word ~ feature vector encoding 3 first letters

Alphabet ~ 26 letters + 1 symbol ‘BLANK’

vector encoding:

ALPHABET	INDEX
‘BLANK’	00
A	01
B	02
C	03
D	04
…	…
X	24
Y	25
Z	26

For example, ONE : ‘O’~14 ‘N’~15 ‘E’~05

Word Encoding (cont'd)

- Word → 27-dimensional feature vector

one (Word)

$\Downarrow$

15 14 05 (Indices)

$\Downarrow$

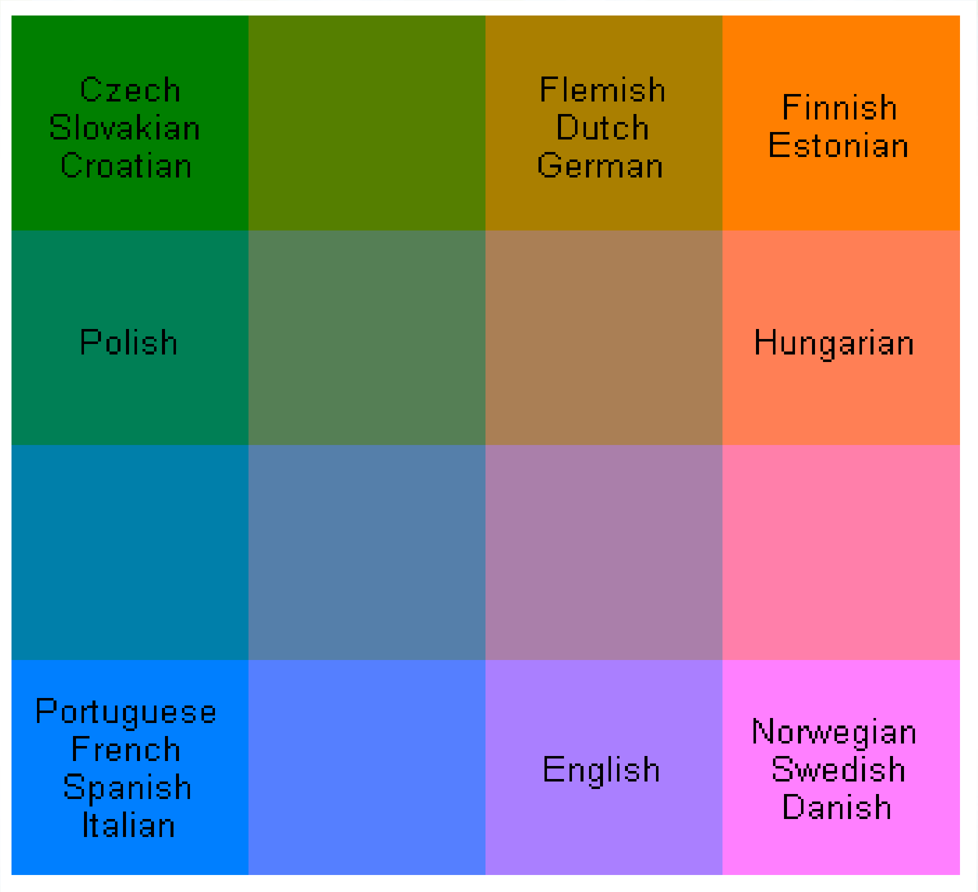
0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

- Encoding is insensitive to order (of 3 letters)
- Encoding of 10-word set: concatenate feature vectors of all words: 'one' + 'two' + … + 'ten'

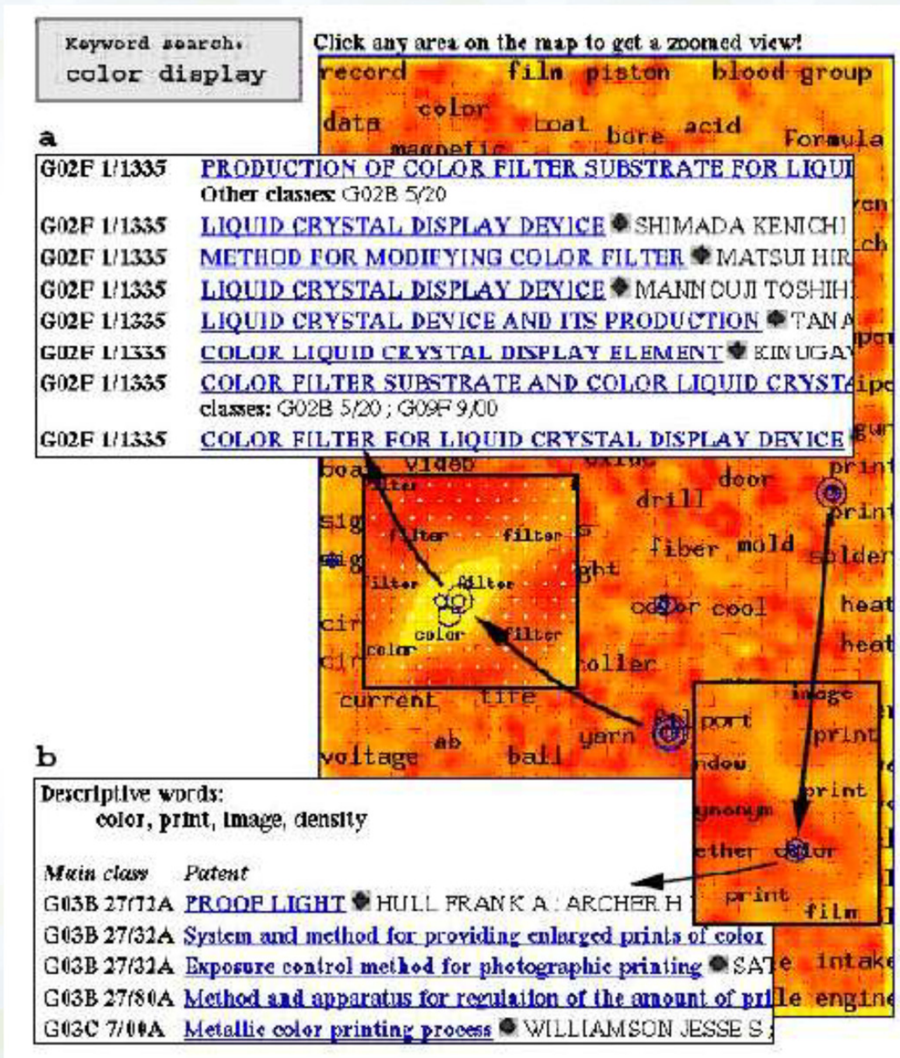
  → **word set** encoded as vector of dim. [1 X 270]

SOM Modeling Approach

2-Dimensional SOM



WEBSOM: Organization of a Massive Document Collection



References:

~Self Organizing Maps ~ Fundamentals ~ J.Bullinaria

~Self-Organizing Maps ~ R.Kiminya