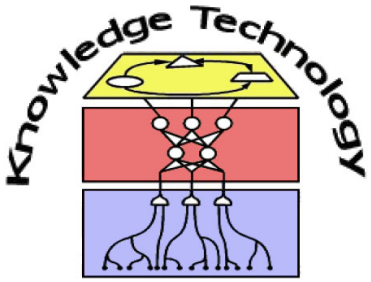
Knowledge Processing with Neural Networks

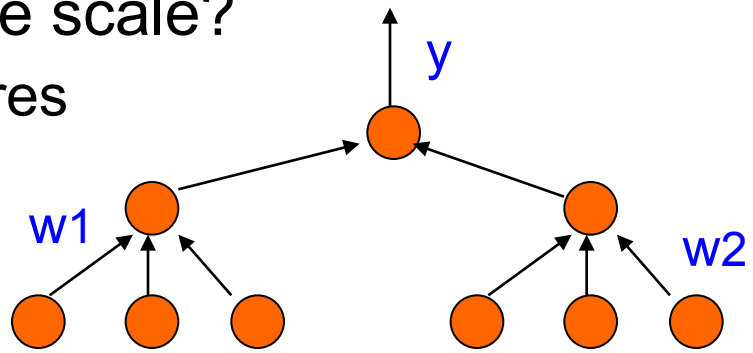
Lecture 6: Neural Clustering



http://www.informatik.uni-hamburg.de/WTM/

Some issues with backpropagation

- Where does the supervision come from?
  - Most data is unlabelled
    - …The vestibular-ocular reflex is an exception….
- How well does the learning time scale?
  - Its is impossible to learn features for different parts of an image independently if they all use the same error signal.



- Can neurons implement backpropagation?
  - Not in the obvious way but perhaps at a higher level cognitive view (to some extent)
- However, also current successess with deep learning…

Three kinds of learning

- Supervised Learning: this models p(y|x)
  - Learn to predict a real valued output or a class label from an input.
- Reinforcement learning: tries to be successful at the end
  - Choose actions that maximize payoff
- Unsupervised Learning: this models p(x)
  - Build a causal generative model that explains why some data vectors occur and not others

or

  - Discover interesting features; separate sources that have been mixed together; find temporal invariants etc. etc.

The Goals of Unsupervised Learning

- The general goal of unsupervised learning is to ***discover*** useful ***structure*** in large data sets, without requiring a desired target output or reinforcement signal.
  - It is not obvious how to turn this general goal into a specific objective function that can be used to drive the learning.

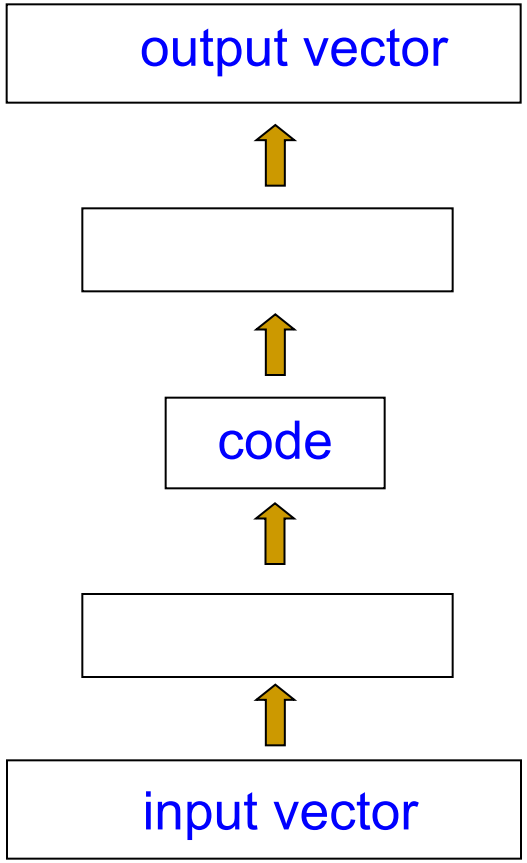
- A more specific goal:
  - Create representations that are better for subsequent supervised or reinforcement learning.

Another Goal of Unsupervised Learning

- Build a density model of the data vectors.
  - This assigns a "score" or probability to each possible data vector
- There are many ways to use a good density model.
  - ***Classify*** by seeing which model likes the test case data most
  - ***Monitor*** a complex system by noticing improbable states
  - ***Extract*** interpretable factors (causes or constraints).

Using backprop for unsupervised learning

- Try to make the output the same as the input in a network with a central bottleneck.
  - The activities of the hidden units in the bottleneck form an efficient code.
    - The bottleneck does not have room for redundant features.
  - Good for extracting independent features (as in the family trees)
  - Autoassociator (same input and output)



Self-supervised backprop in a linear network

- If the hidden and output layers are linear, it will learn hidden units that are a linear function of the data and minimize the squared reconstruction error.
  - This is what PCA (Principal Components Analysis) does.

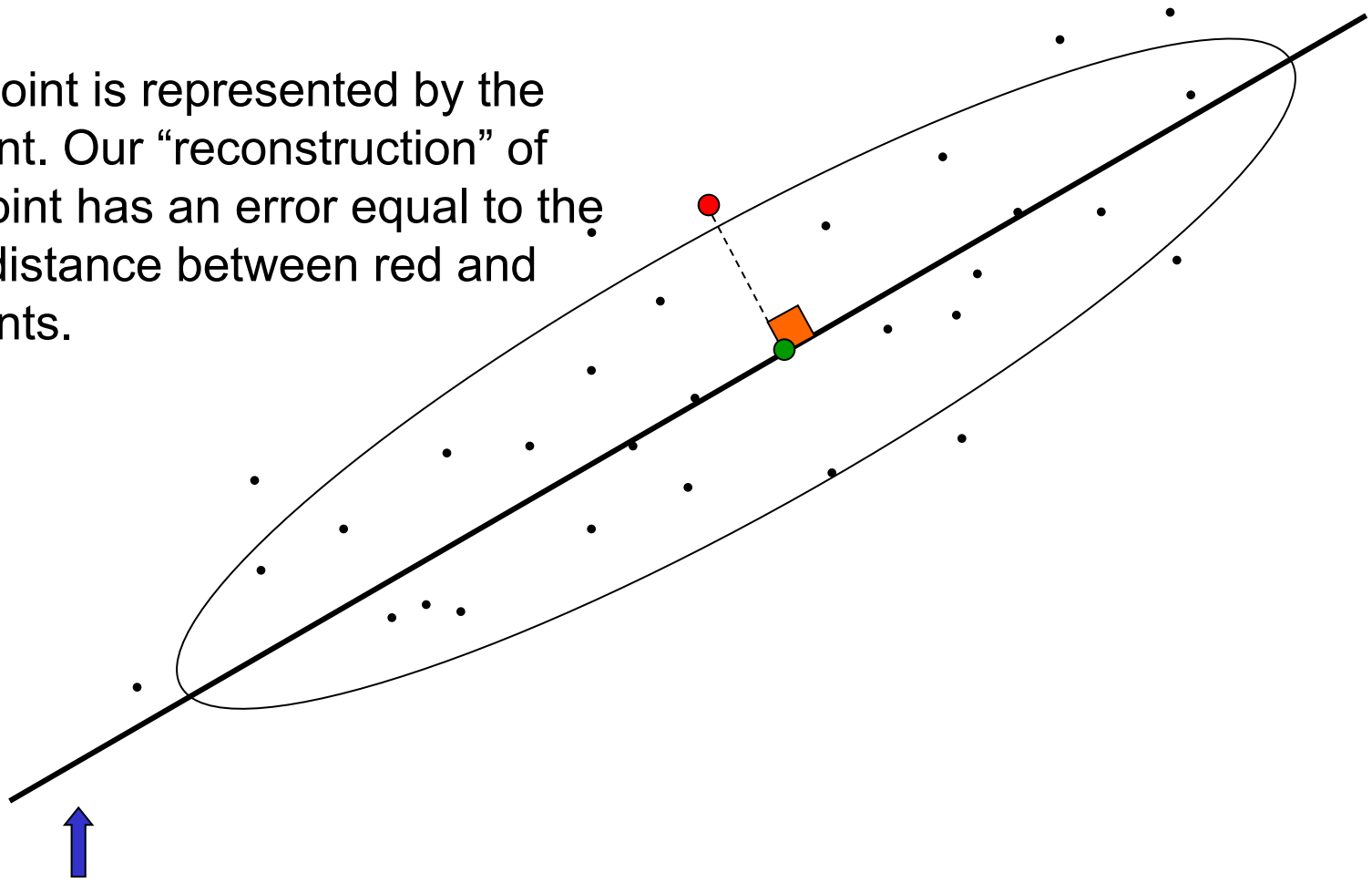
- The M hidden units will span the same space as the first M principal components found by PCA
  - Their weight vectors may not be orthogonal
  - They will tend to have equal variances

Principal Components Analysis

- This takes N-dimensional data and finds the M orthogonal directions in which the data have the most variance
- These M principal directions form a subspace.
- We can represent an N-dimensional datapoint by its projections onto the M principal directions
  - This loses information about where the datapoint is located in the remaining orthogonal directions.
- We reconstruct by using the mean value (over all the data) on the N-M directions that are not represented.
  - The reconstruction error is the sum over all these unrepresented directions of the squared differences from the mean.

A picture of PCA with N=2 and M=1

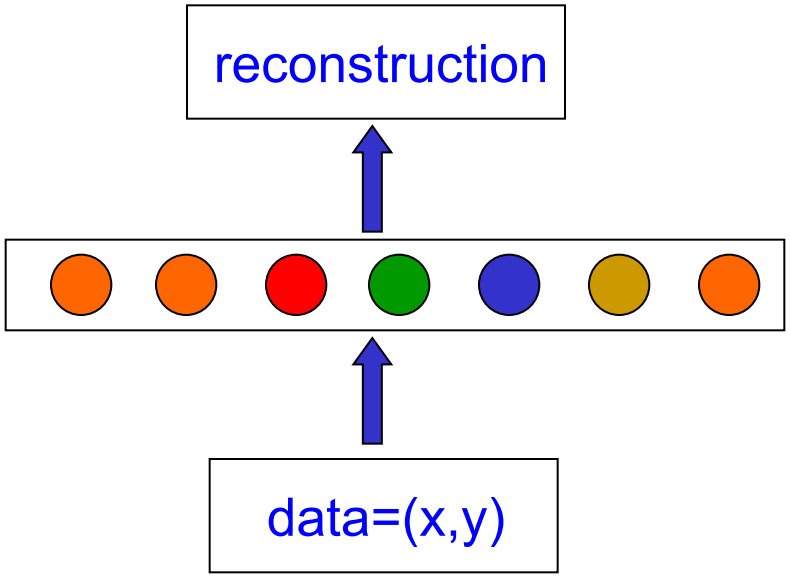
The red point is represented by the green point. Our “reconstruction” of the red point has an error equal to the squared distance between red and green points.

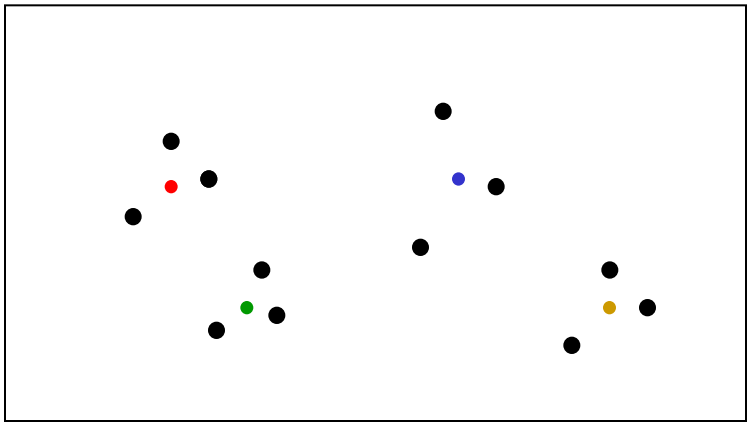


First principal component:
Direction of greatest variance

Self-supervised backprop and clustering

- If we force the hidden unit whose weight vector is closest to the input vector to have an activity of 1 and the rest to have activities of 0, we get clustering.
- The weight vector of each hidden unit represents the ***center*** of a cluster.
- Input vectors are reconstructed as the nearest cluster center.



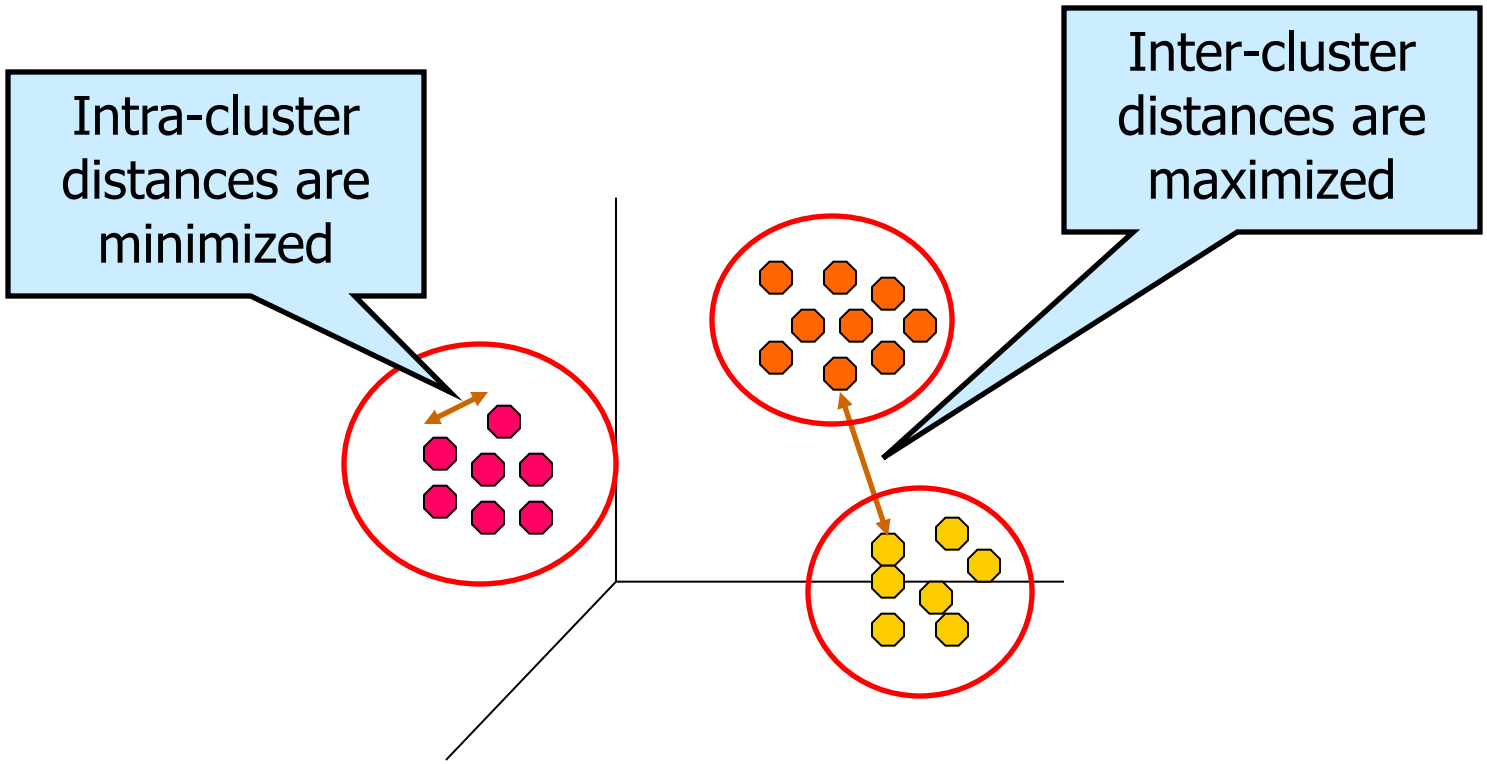


Clustering questions and motivation

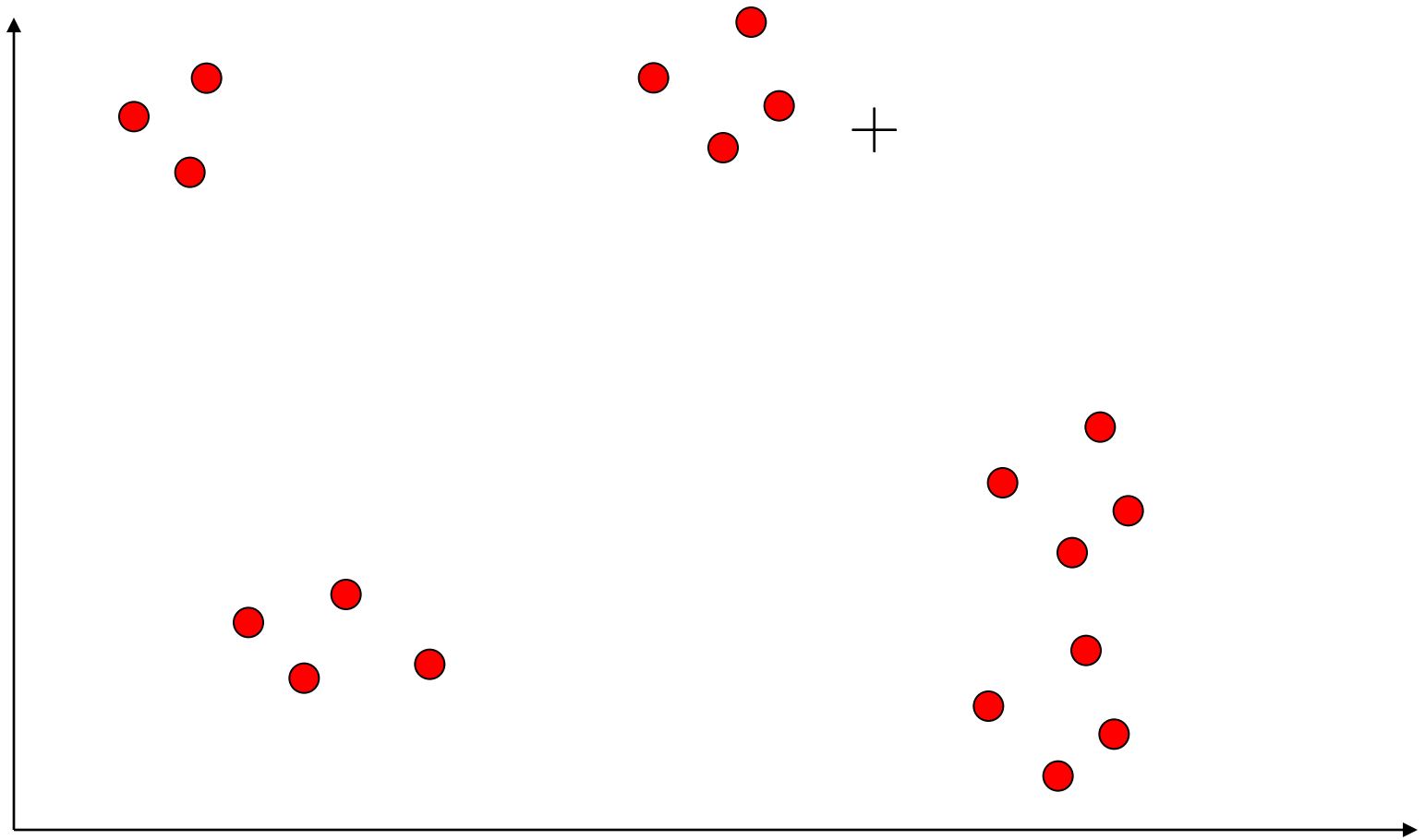
- We assume that the data was generated from a number of different classes. The aim is to cluster data from the same class together.
- How do we decide the number of classes?
  - Why not put each datapoint into a separate class?
  - What is the payoff for clustering things together?
- What if the classes are hierarchical?
- What if each datavector can be classified in many different ways? A one-out-of-N classification is not nearly as informative as a distributed code.

What is Cluster Analysis?

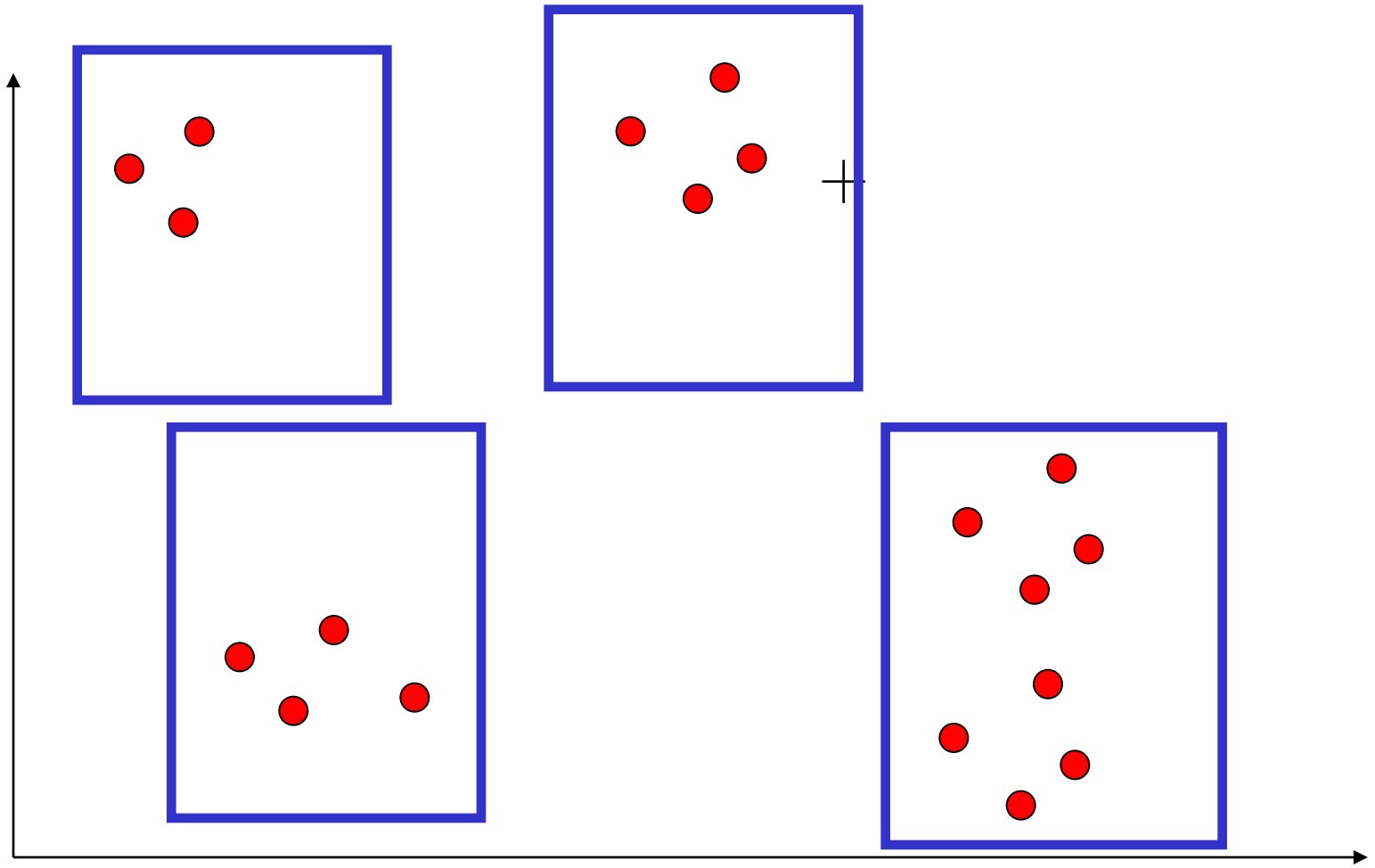
- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



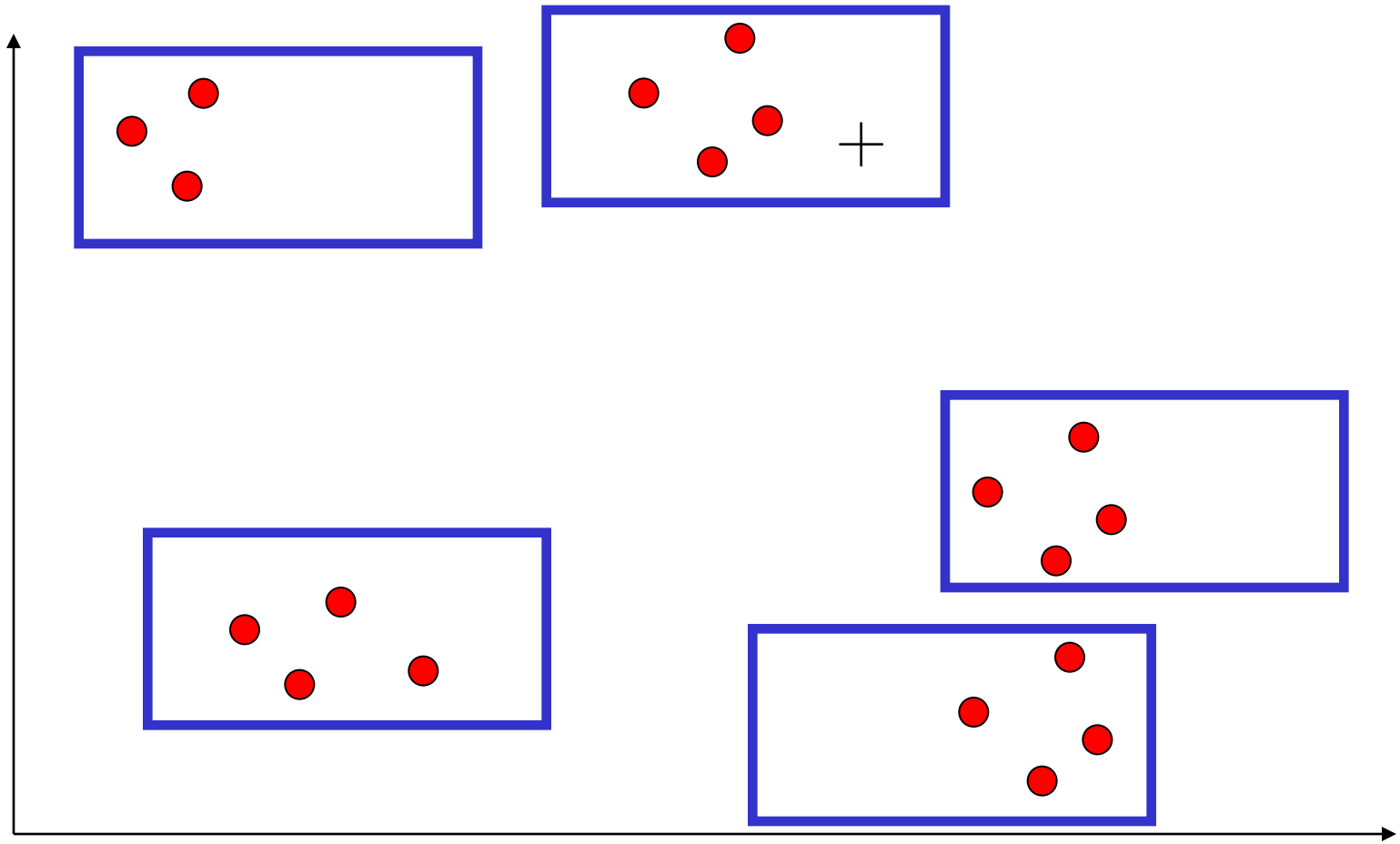
Clustering: Classes unknown



Clustering of previous objects



Clustering of previous objects



Clustering Problem

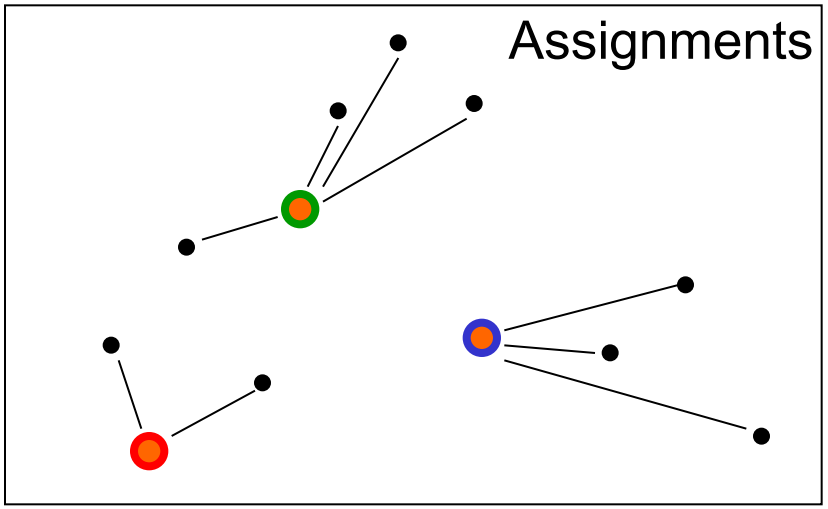
- Clustering differs from classification in that clusters for input data are not known a priori.

- Given a database $D=\{t_1,t_2,\ldots,t_n\}$ of input data (tuples) and an integer value k, the ***Clustering Problem*** is to define a mapping $f:D\rightarrow\{1,..,k\}$ where each t_i is assigned to one cluster K_j, $1<=j<=k$.

- A ***Cluster***, K_j, contains precisely those tuples mapped to it.

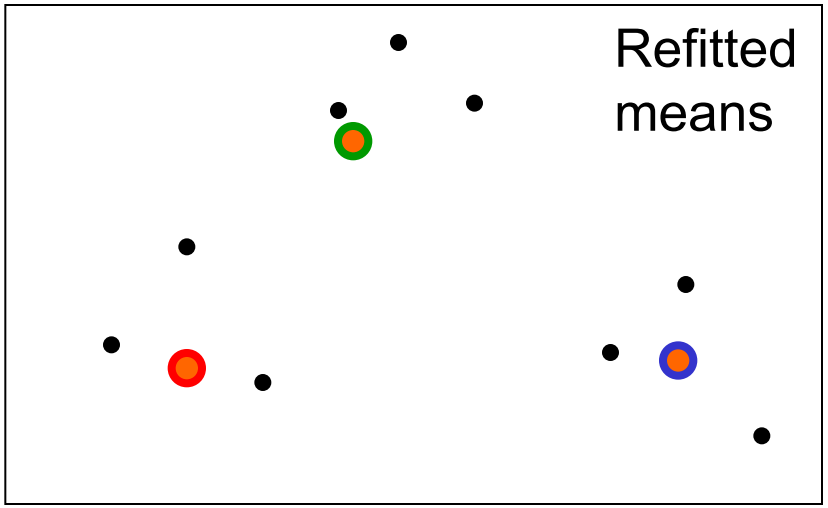
The k-means algorithm (reminder)

- Assume the data lives in a Euclidean space.
- Assume we want k classes.
- Assume we start with randomly located cluster centers



The algorithm alternates between two steps:

1. ***Assignment step***: Assign each datapoint to the closest cluster.
2. ***Refitting step***: Move each cluster center to the center of gravity of the data assigned to it.



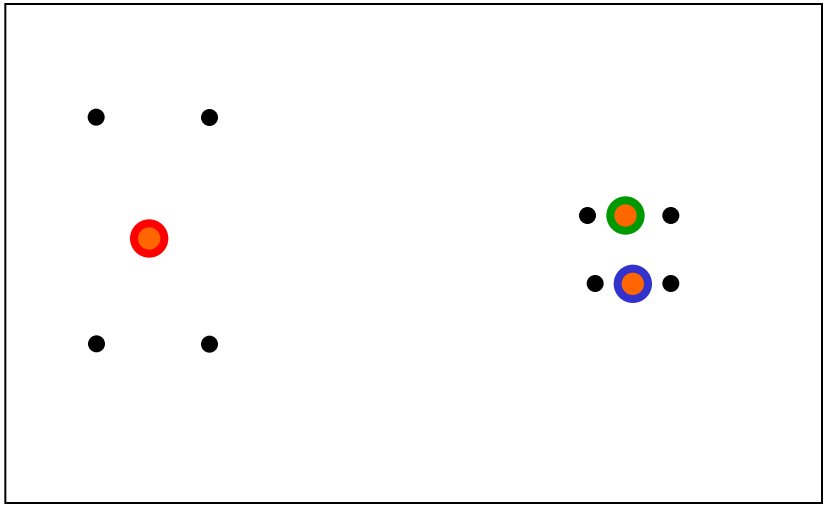
Why k-means converges

- Whenever an assignment is changed, the sum squared distances of datapoints from their assigned cluster centers is reduced.
- Whenever a cluster center is moved, the sum squared distances of the datapoints from their currently assigned cluster centers is reduced.
- If the assignments do not change in the assignment step, we have converged.

Local Minima

- There is nothing to prevent k-means getting stuck at local minima.
- We could try many random starting points
- We could try non-local split-and-merge moves: Simultaneously merge two nearby clusters and split a big cluster into two.

A bad local optimum



The Self-Organising Map

- One neural method of clustering was proposed by Kohonen
- Idea inspired by the way that mappings are learnt in the topographic feature maps found in many brain areas
- These ***feature maps*** consist of one- or two-dimensional sheets of neurons with lateral interactions

Feature Maps

- Patterns that are similar (‘close together’) excite neurons that are near to each other
- Patterns that are very different excite neurons far away
- This is known as ***topology preservation***
- The ordering of the inputs is preserved
  - if possible (perfectly topology-preserving)

The Self-Organizing Map Algorithm

- The weight vectors are randomly initialized
- Input vectors are presented to the network
  - Determine ***best matching neuron*** n_b with the minimal Euclidean distance between input x and the weight vector w

$$n_b = \min_j \left\| x - w_j^T \right\|$$

  - The winning node (and neighbors) have the weight vector moved ***closer to the input*** (with a learning rate $\eta(t)$)

$$w_j^T \leftarrow w_j^T + \eta(t) \cdot \left(x - w_j^T \right)$$

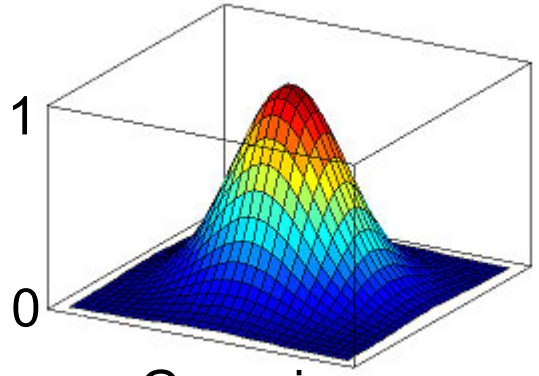
- Over time, the network ***self-organizes*** so that the input topology is preserved
- (so far, this is like on-line version of k-means)

Neighborhood Function Preserves Topology

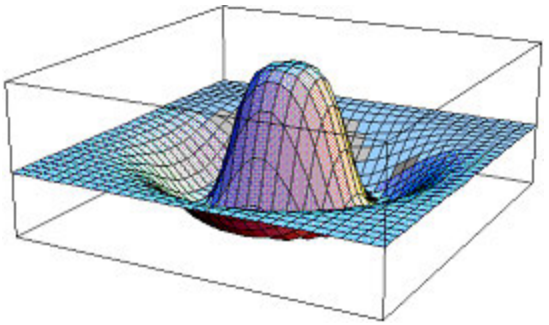
- The neighborhood function $h(n_b,t)$ determines the degree of weight vector change of the neighbors

$$w_j^T \leftarrow w_j^T + \eta(t) \cdot h(n_b,t) \cdot \left(x - w_j^T\right)$$

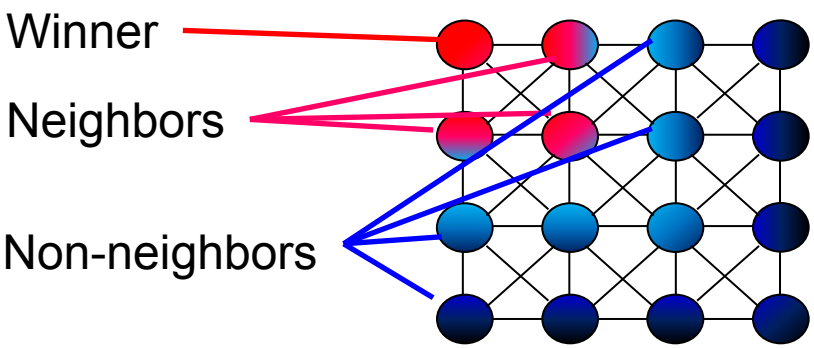
  - Mostly: Gaussian function
    rarely: Mexican Hat function
  - Width decreases during training
    (→ implicit decrease of learning rate)
  - *May* decrease to zero (→ k-means)



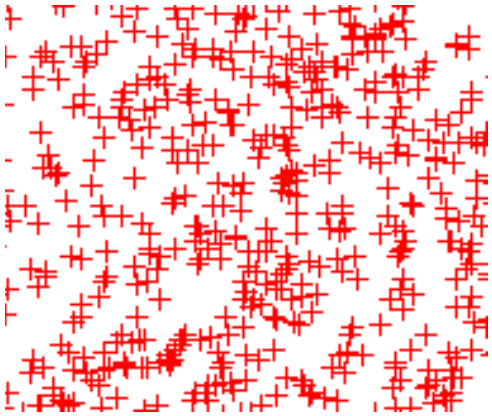
Gaussian
(not normalized)



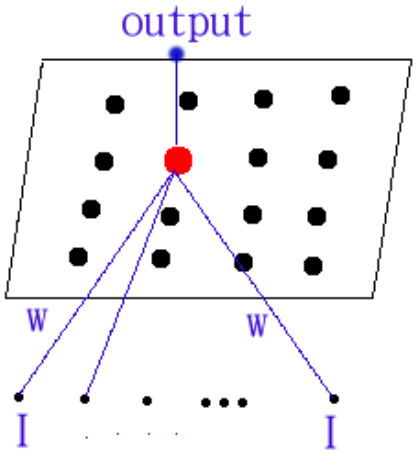
Mexican Hat
(Difference of Gaussian)

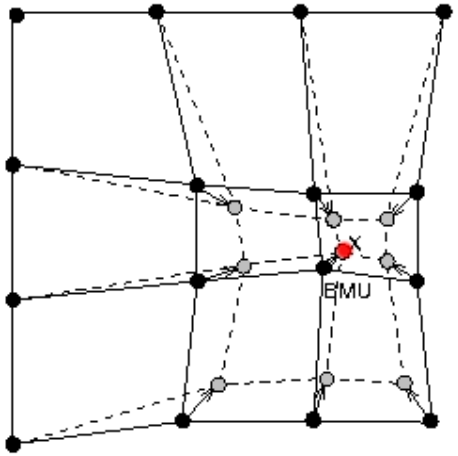


Representation in Self-organizing Networks



Raw Data





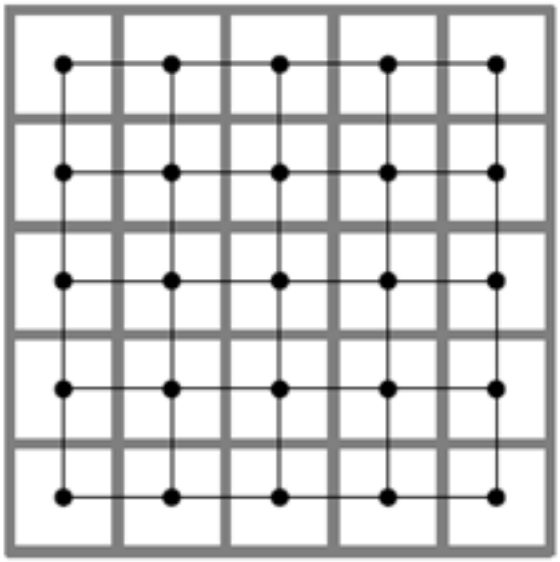
While computational bounds are not exceeded **do**

1. Select an input sample
2. Compute the Euclidean distance between input and weight vector for each output node
3. Select the output node with minimum distance
4. Update weights to all nodes within a topological distance of winning node

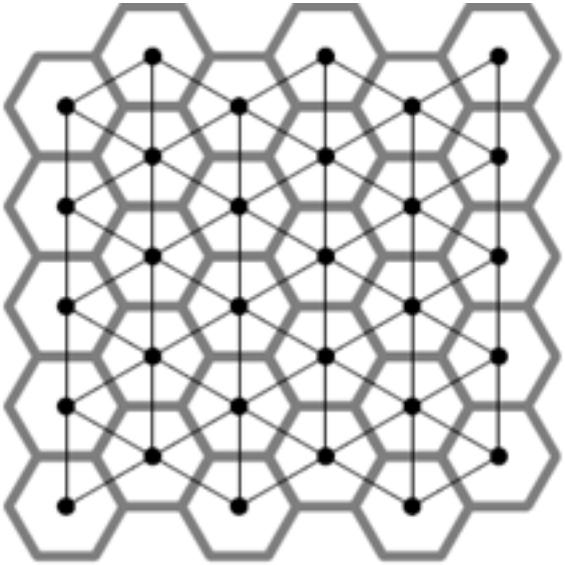
End-while

Neighborhood of Output Neurons

Quadratic



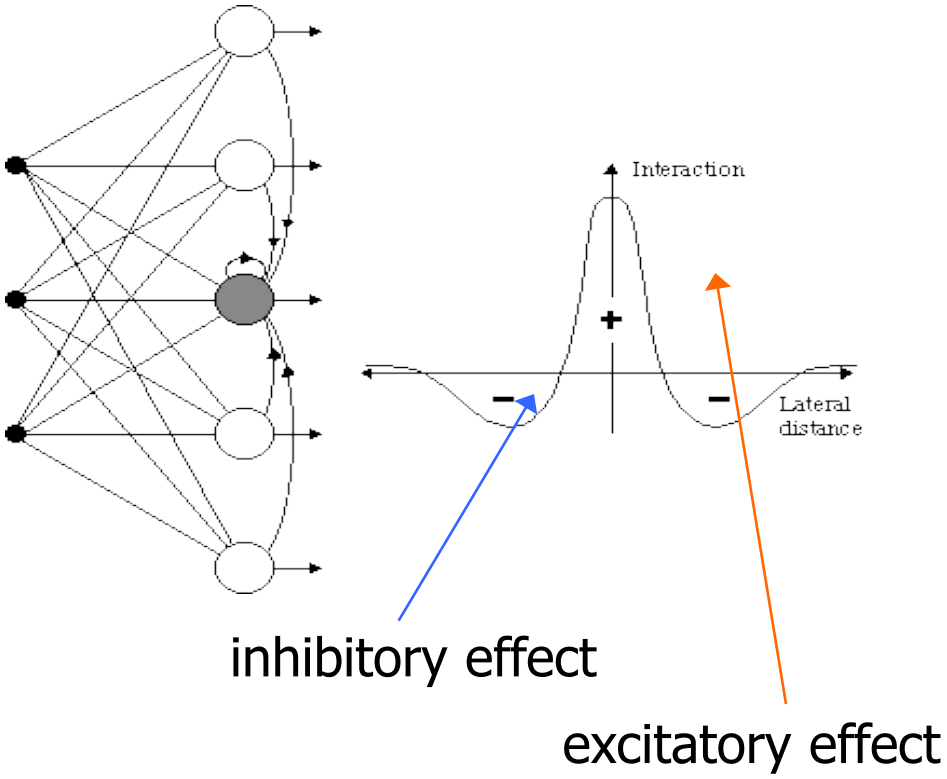
Hexagonal



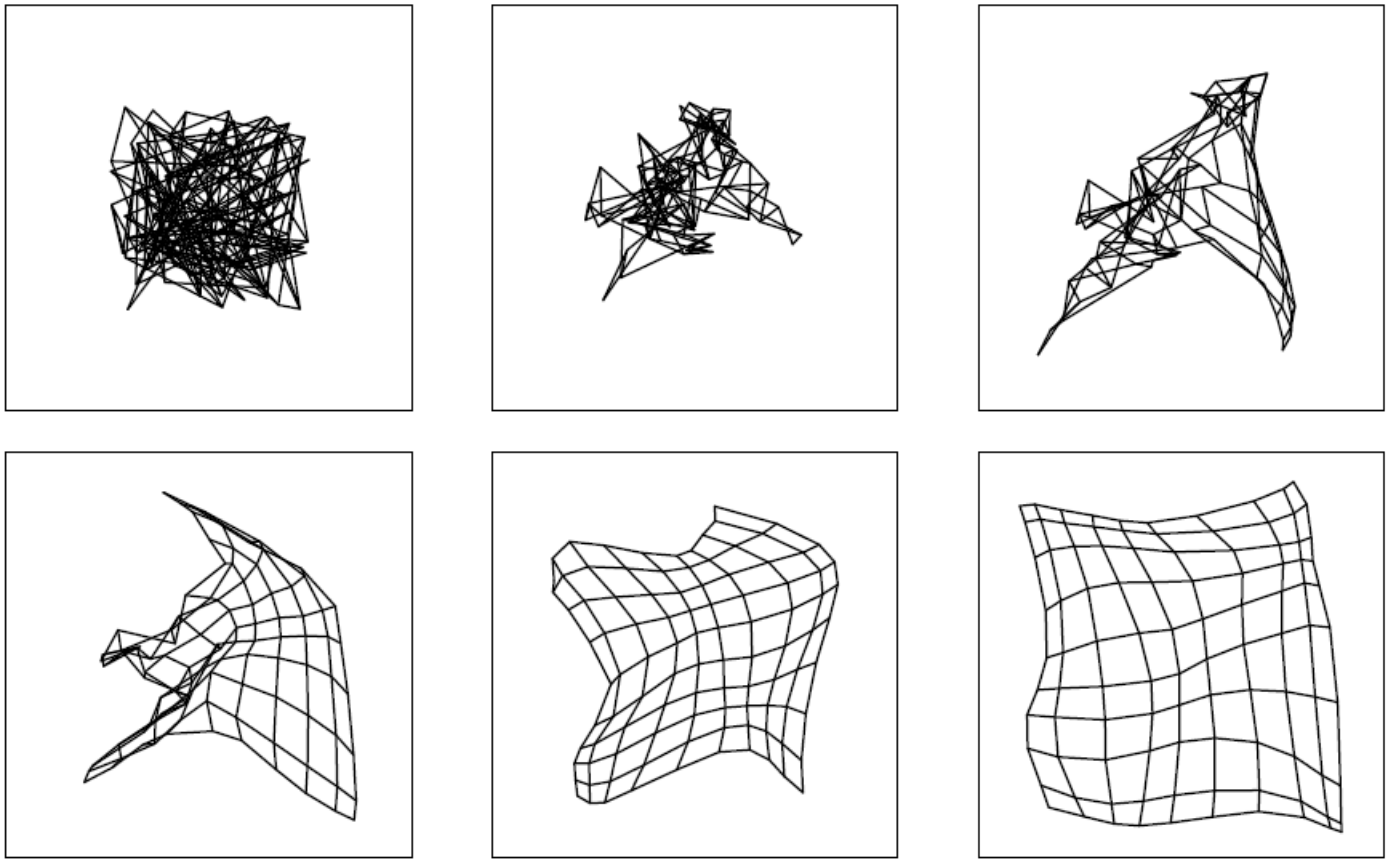
- Thin lines show next neighbors of a neuron
- Thick gray lines show regions of a neuron

Feature Map and Mexican Hat Function

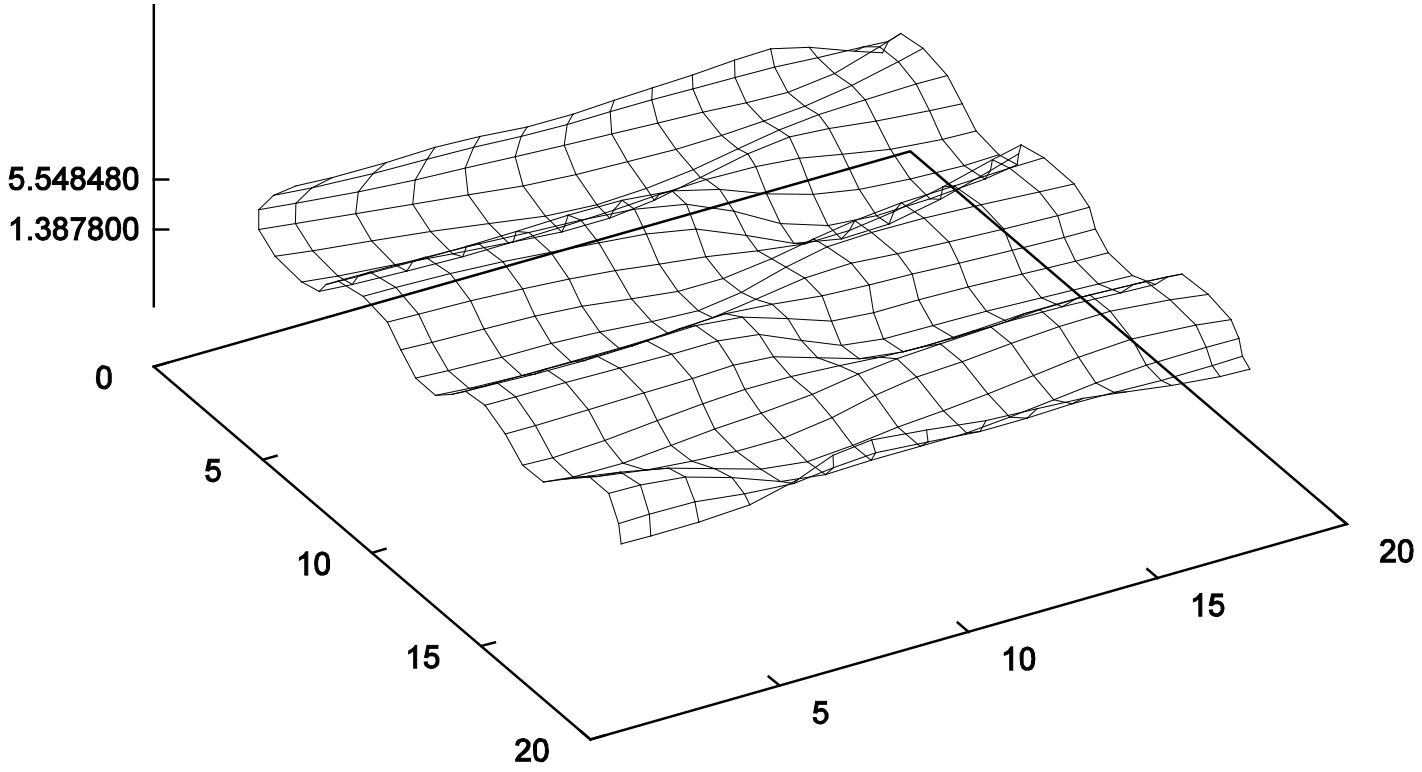
- The Mexican hat function represents the relationship between the distance from the winning neuron and the strength of "connections" within the Kohonen layer
- Near neighbourhood – short range lateral excitation area has strong positive effect
- Remote neighbourhood – has a weak negative inhibitory effect



Unfolding of a 2-D Map in a 2-D Data Space

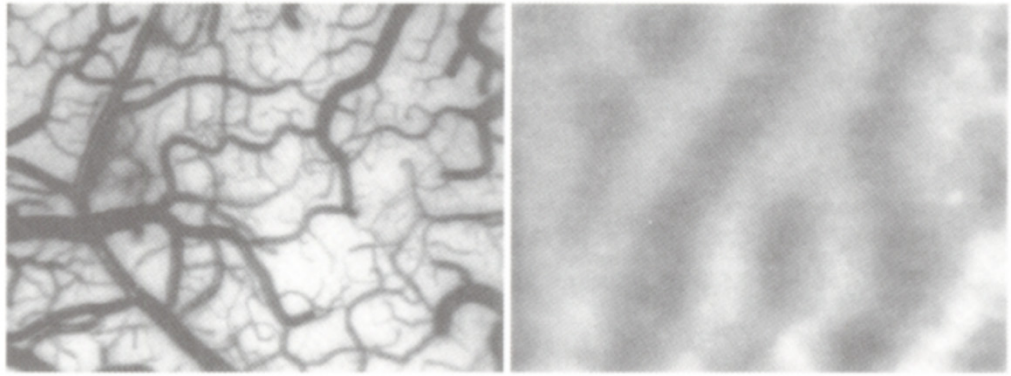


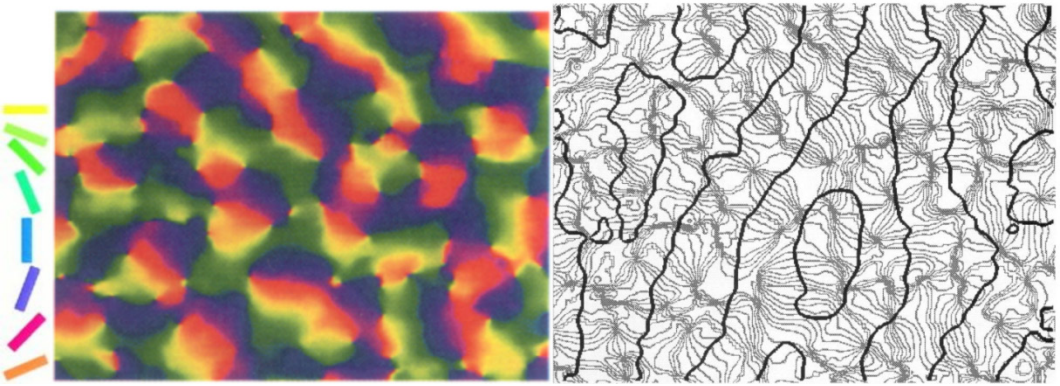
Unfolding of a 2-D Map in a 3-D Data Space



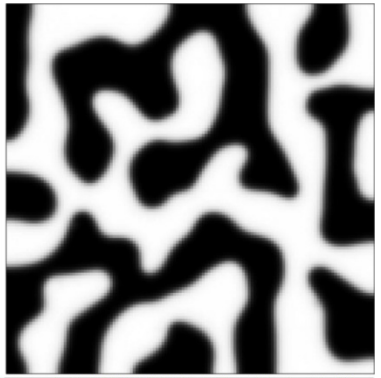
Unfolding of a 2-D Map in a 4-D Data Space

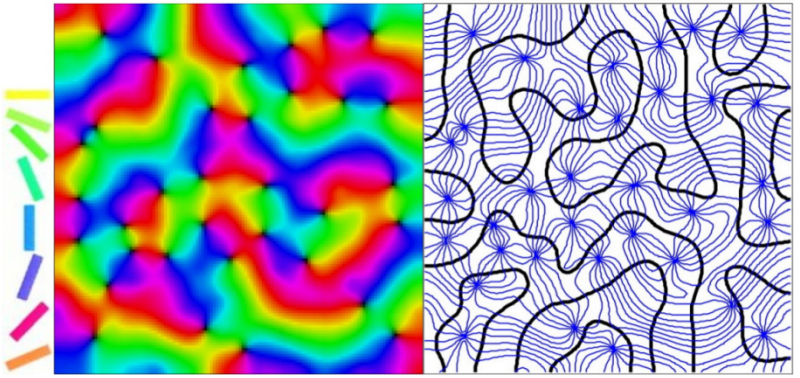
ocular dominance





orientation preference

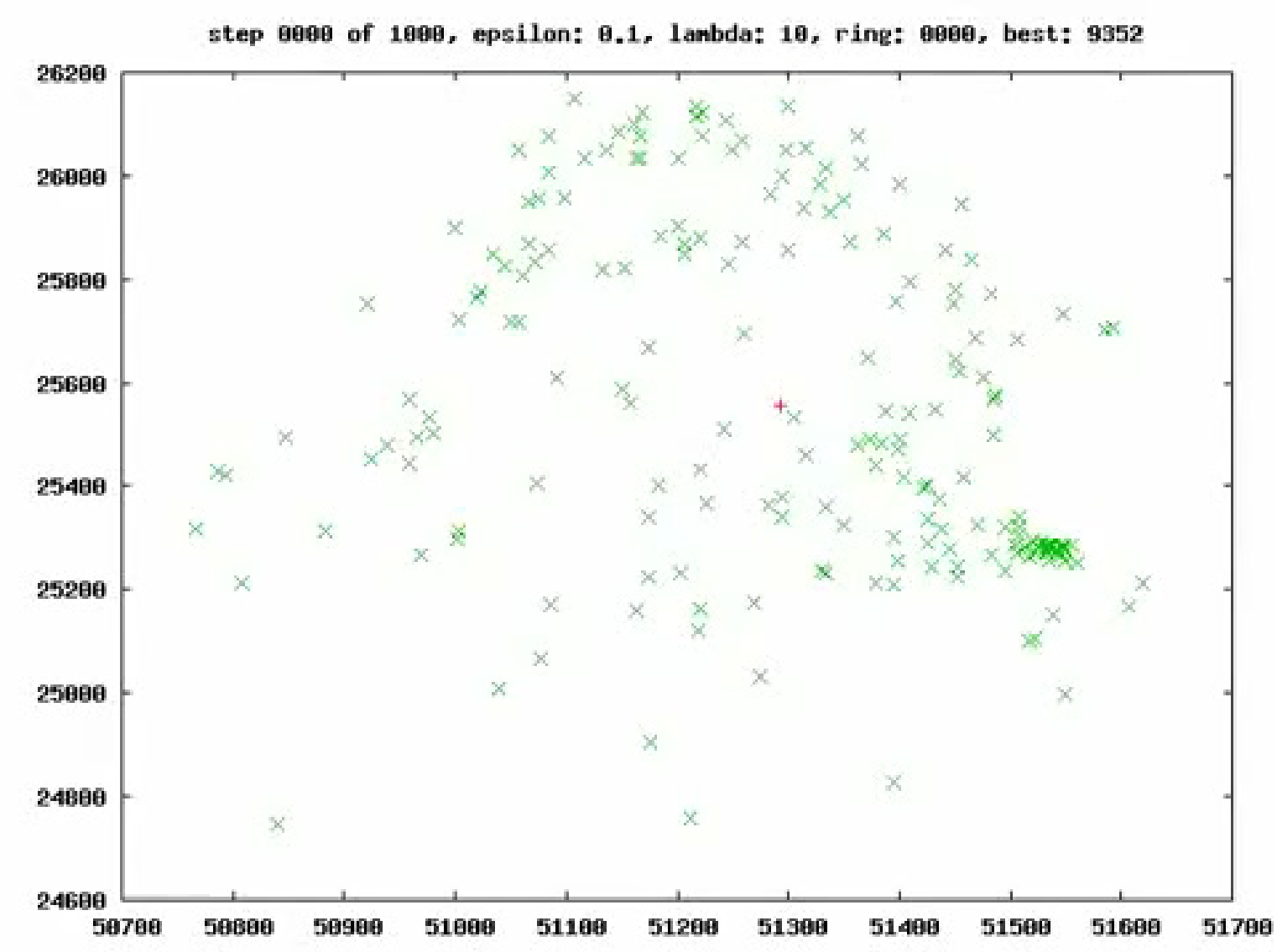




Obermayer, Blasdel. .. Orientation and Ocular Dominance Columns in Monkey .. Jneurosci, 1993

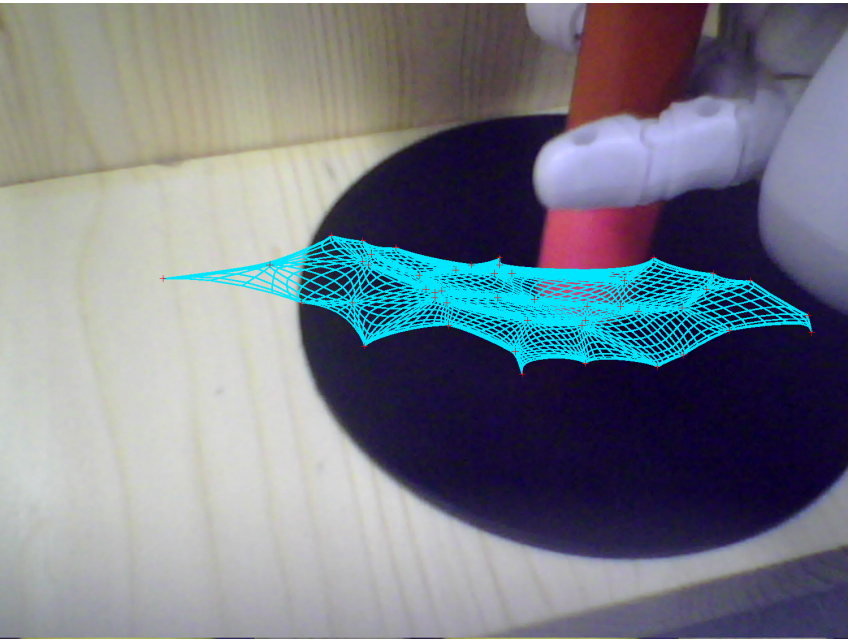
Goodhill . Theoretical Modelling to .. Neural Map Development. Neuron, 2007

Unfolding of a 1-D Map in a 2-D Data Space



http://www.youtube.com/watch?v=8tnxgfE6gII

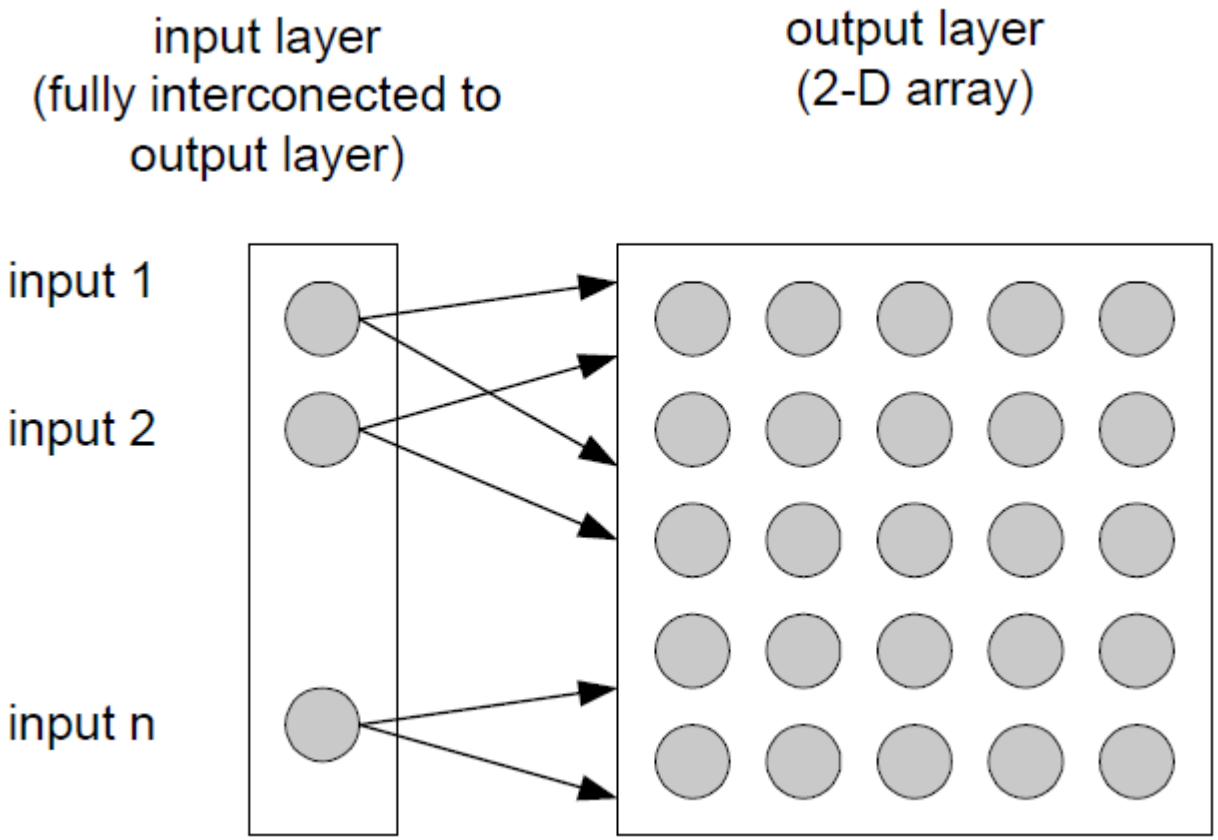
2-D Map in 2-D Visual Space with 5 DOF Output



Rule extraction from SOMs

- SOM has several important properties that can be used within the data mining/knowledge discovery and exploratory data analysis process
- Effort is still required to interpret the cluster boundaries
- Malone/McGarry/Wermter/Bowerman 2006

Rule extraction from SOMs



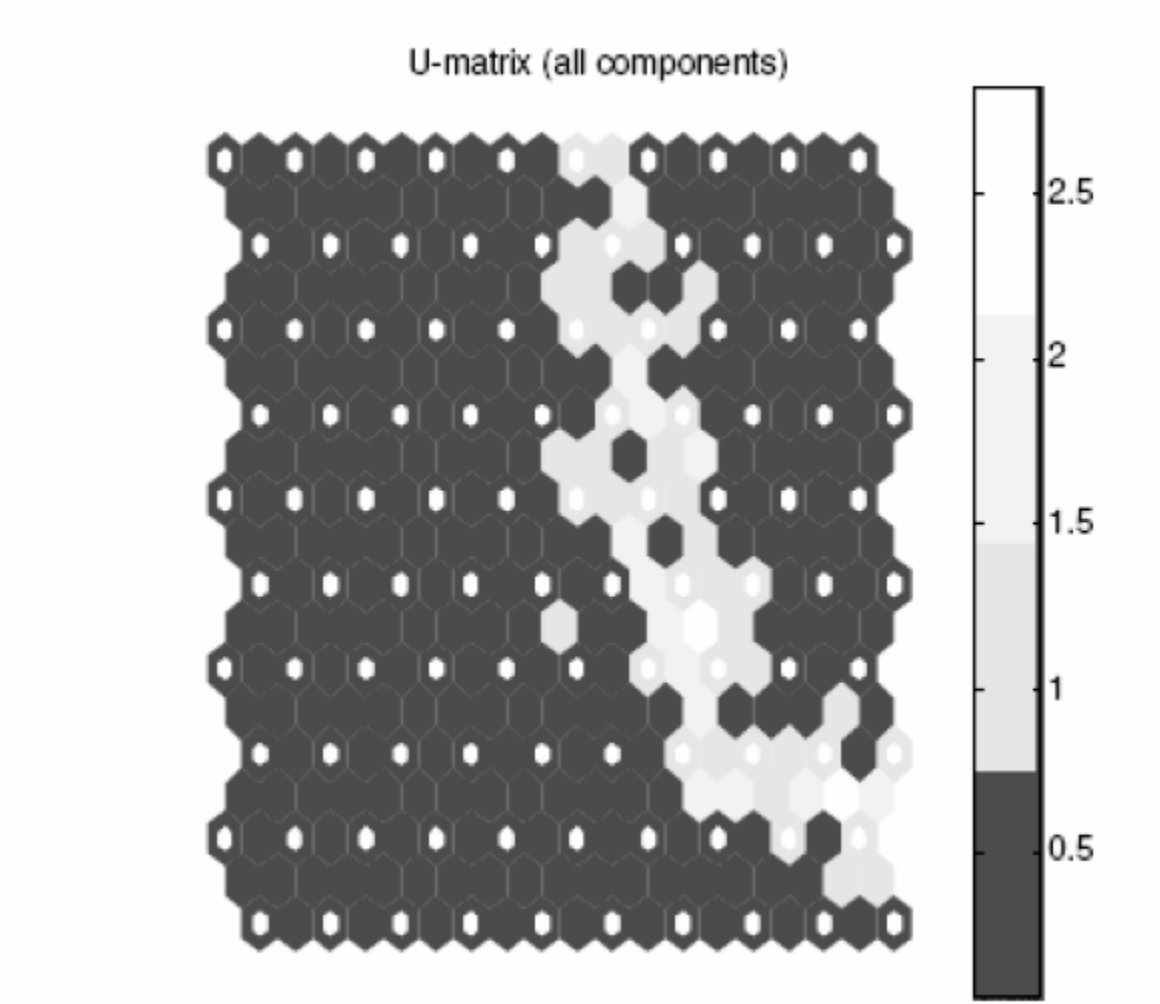
U-matrix of a SOM for identifying cluster boundaries

- Interpret the cluster boundaries by taking the average distance d of a weight vector of a unit i to the weight vectors of its neighbors N

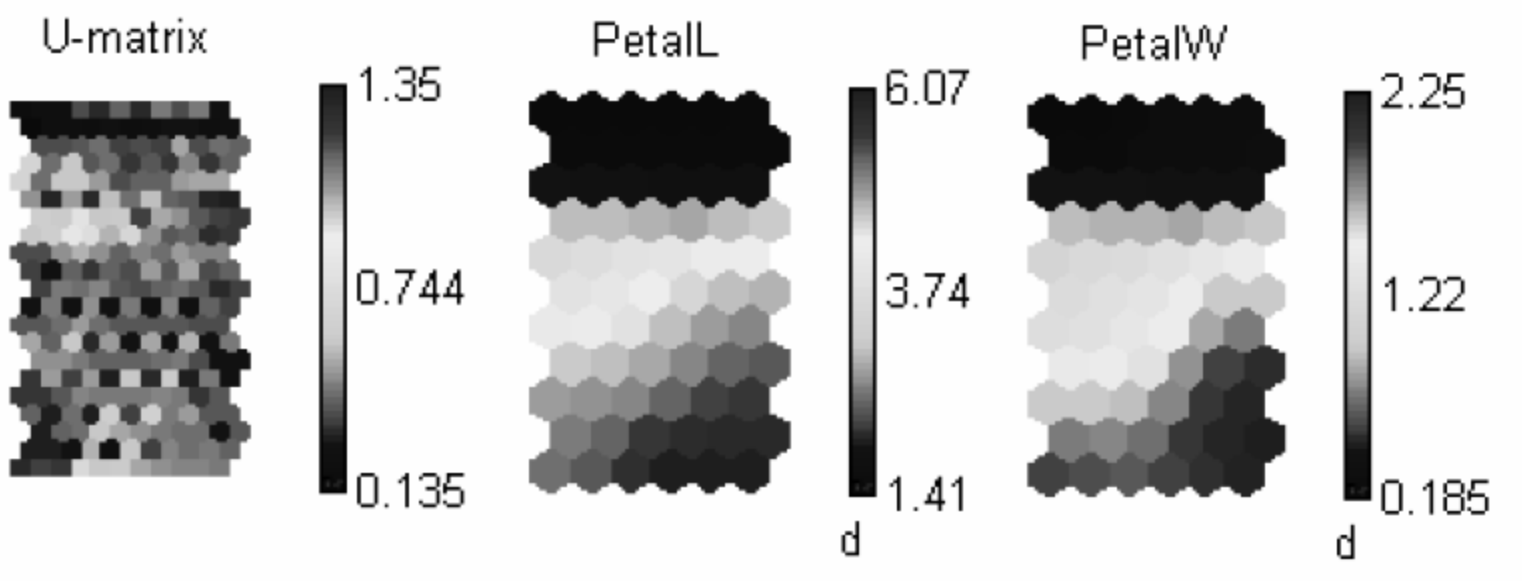
$$u(i) = \frac{1}{n} \sum_j d(w_i, w_j), n_j \in \mathrm{N}(i), \mathrm{n} = |\mathrm{N}(i)|$$

- ***Unified distance matrix*** by Ultsch & Siemon, 1990

Rule extraction from SOMs based on U-Matrix



Rule extraction from SOMs on IRIS data set



1 If PetalL < 2.8 and
If PetalW < 0.75 then Setosa

2 If PetalL > 2.8 and
If PetalL< 5 then Versicolor

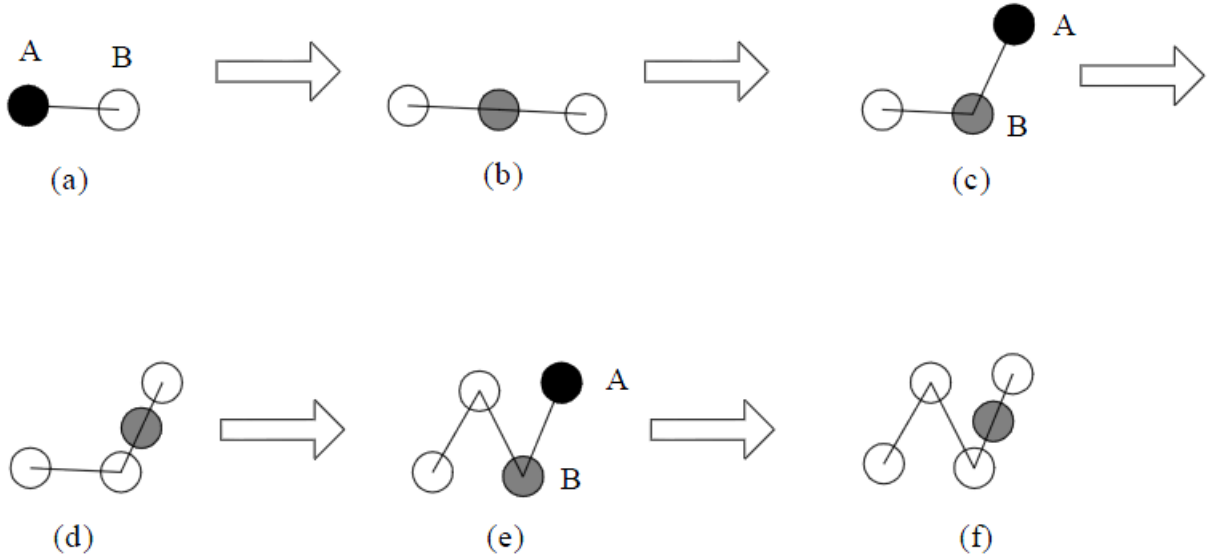
3 If PetalL > 5 then Virginica

Advanced Neural Clustering models

- ***Static Models***: Competitive Learning (CL), Self-Organizing Map (SOM), Neural Gas (NG), …
- ***Dynamic Models***: Growing Grid (GG), Growing Cell Structure (GCS), Growing Neural Gas (GNG), Grow When Required (GWR), Dynamic Adaptive Self-organizing Hybrid model (DASH), etc.
- ***Hierarchical Models***: Multilayered Self-Organising Feature Maps (M-SOM), Growing Hierarchical Self-Organizing Map (GHSOM), etc.

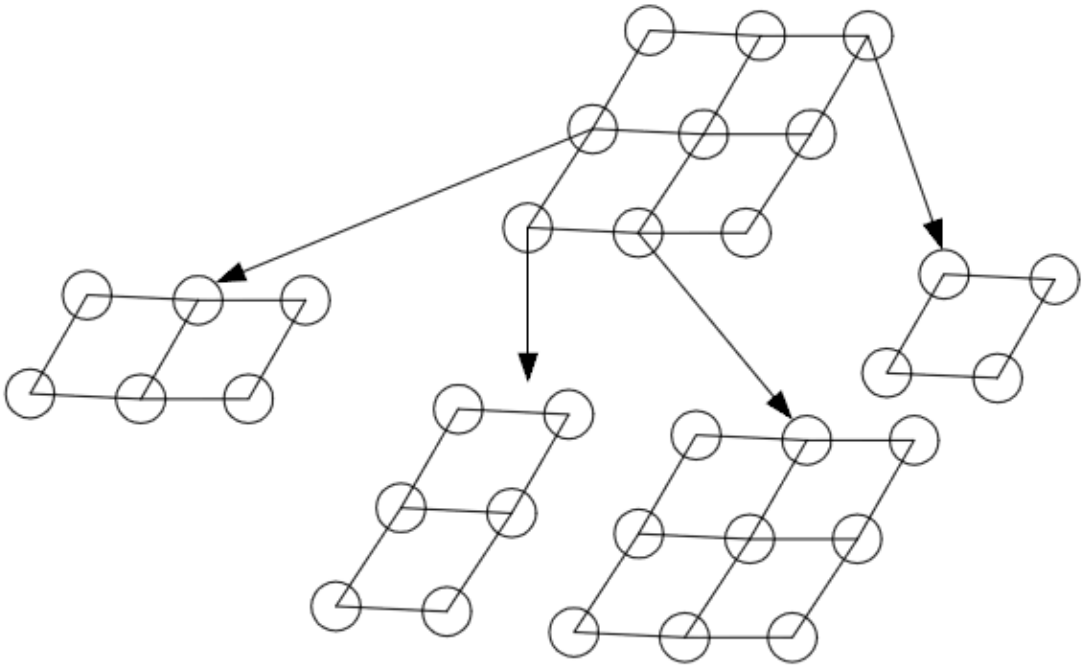
Non-stationary environments are hard to master!

Growing Processes for DASH Model
(Growing Neural Gas – GNG Fritzke)



- GNG adds unit after every pre-defined period. Units are represented as circles
- Circle A indicates ***unit with biggest error***
- Circle B indicates ***neighbour with biggest error*** for Circle *A*
- The grey circle is the ***new unit*** at each stage

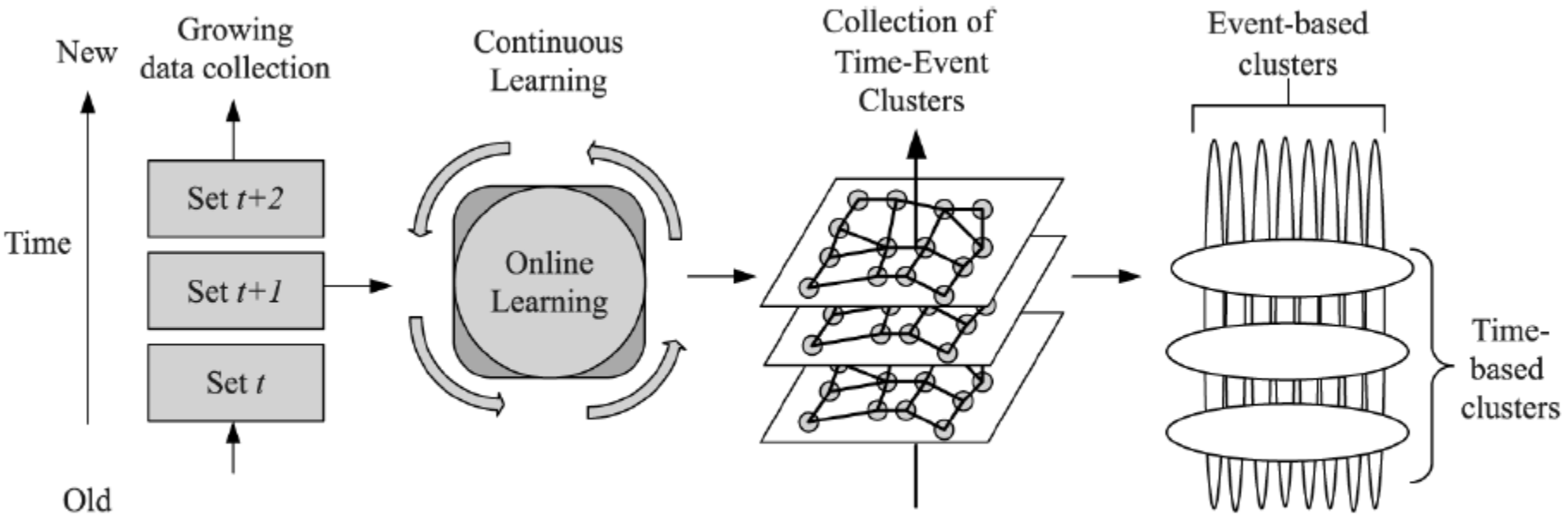
Growing Hierarchical SOM
(Rauber et al)



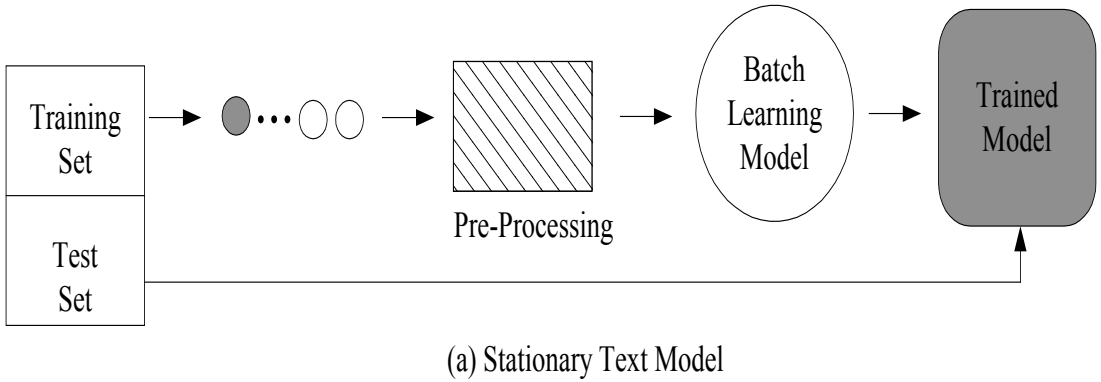
A sub-map ***grows*** from a unit whose error is greater than a pre-defined proportion of the expected unit error

Dynamic Adaptive Self-organising Hybrid DASH Model (Hung, Wermter 2007)

- Cluster documents changing over time in a collection of growing SOM

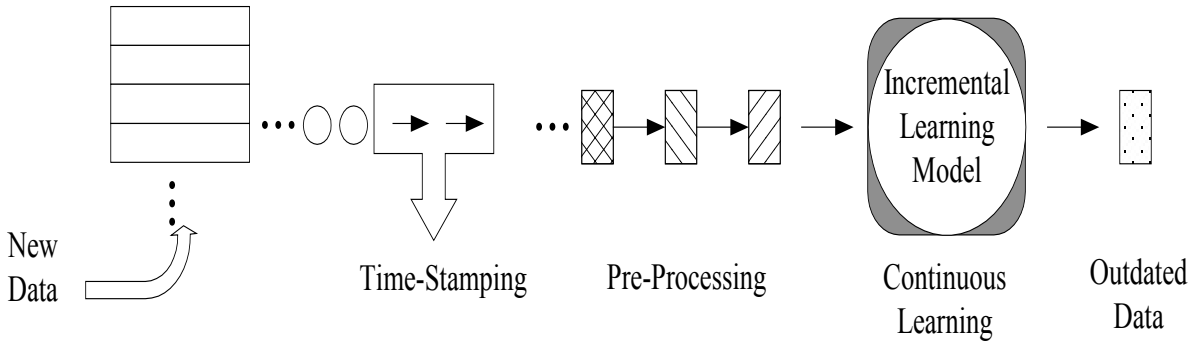


Stationary and Non Stationary Models



(a) Stationary Text Model

- Fixed number of documents
- Model is out-of-date soon
- Results require a complete training procedure



(b) Non-Stationary Text Model

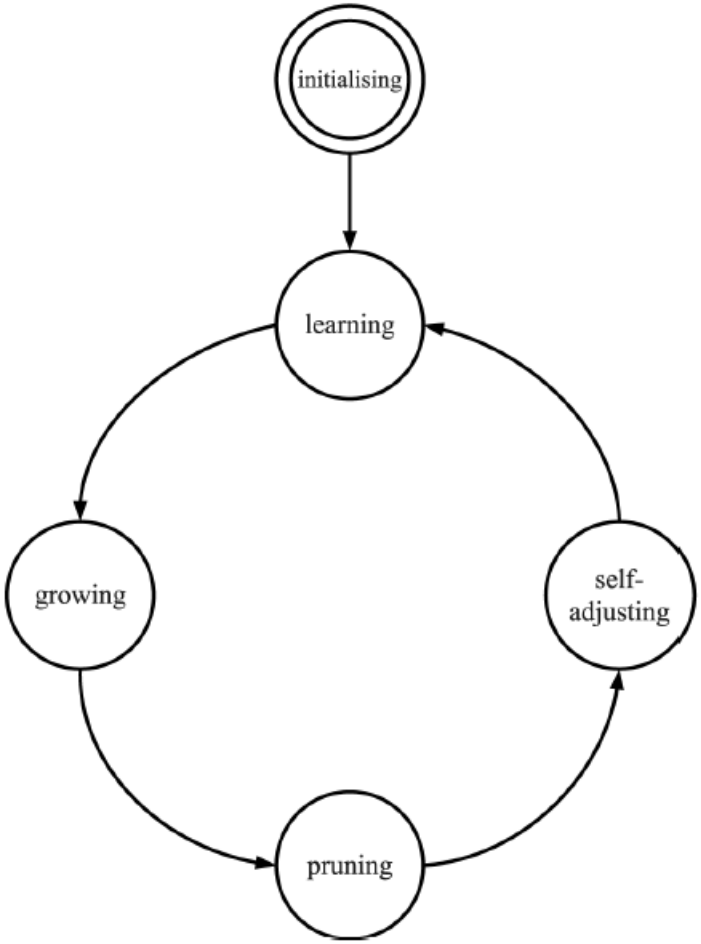
- Number of documents ever-increasing
- Model kept up-to-date
- Existing data set can be replaced by new data set at any time

Five Stages of dynamic adaptive Self-Organization Hybrid Model

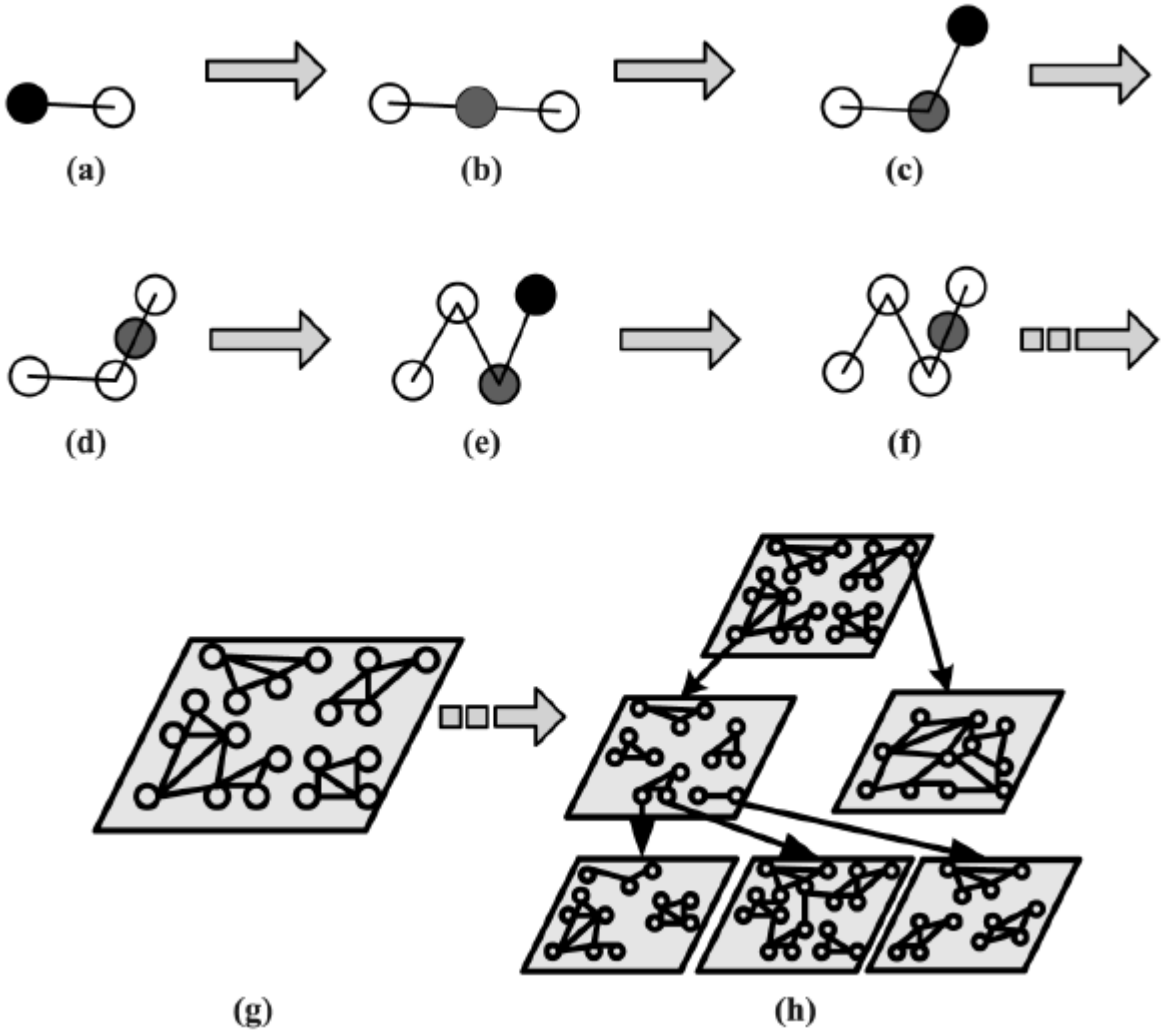
- *Concept*: Define an age threshold B for a connection between output units *i* and *j* – if a connection is too old it is pruned

$$\frac{age_{ij}}{age_{\max}} > \beta$$

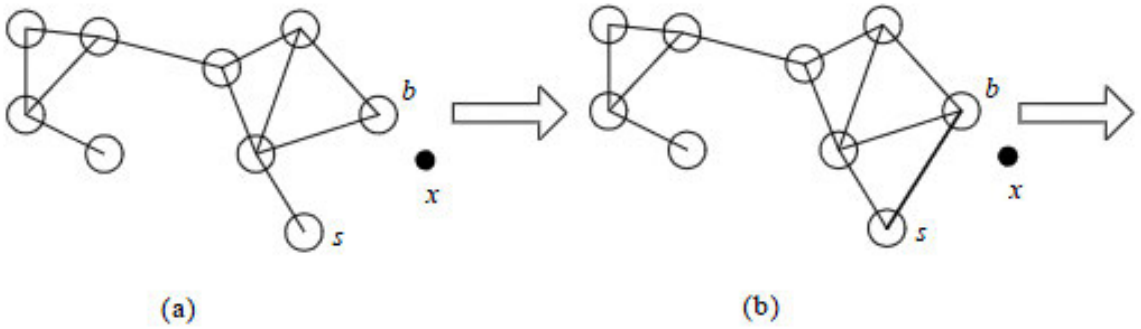
- DASH increases the age for all connections except best matching unit (***BMU***)

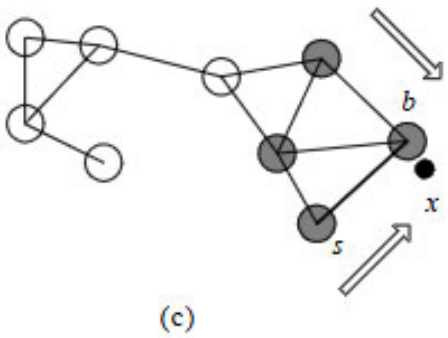


Growing Processes for DASH Model



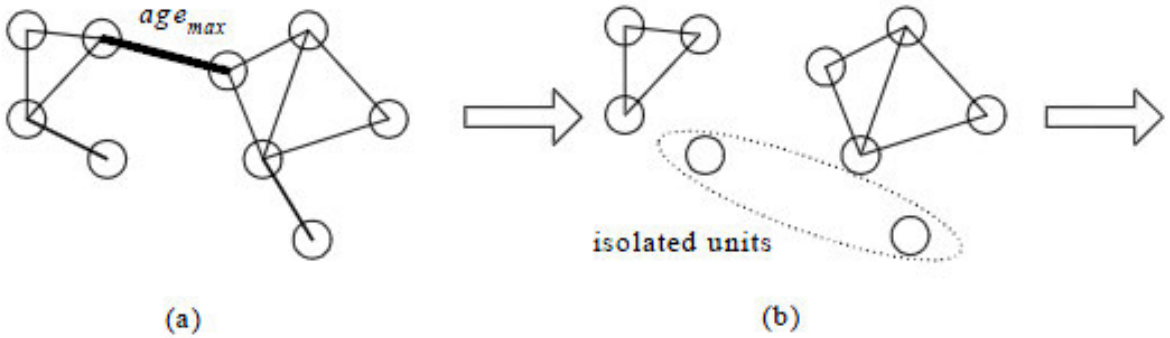
Growing for DASH model

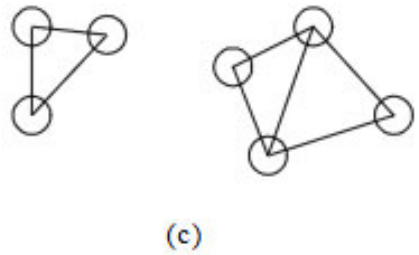




- (a) Current input vector x is represented as a black circle.
- (b) Unit b is the Best Matching Unit (BMU) for input vector x and Unit s is the Second best Matching Unit (SMU) for input vector x. A connection is built between Units b and s.
- (c) Unit b and its direct neighbors which are represented as grey circles move towards the input vector x.

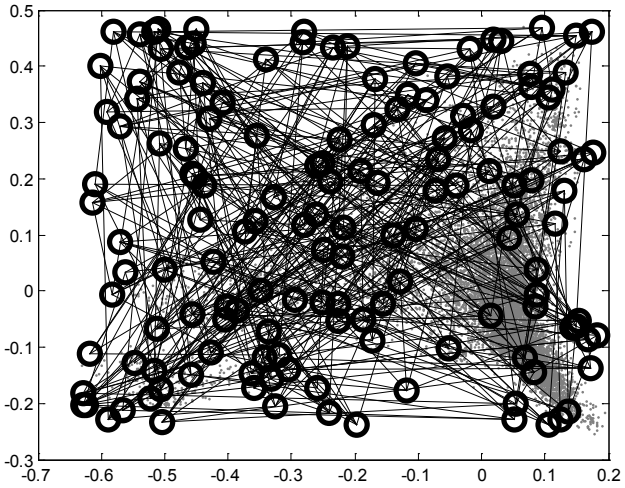
Pruning for DASH Model

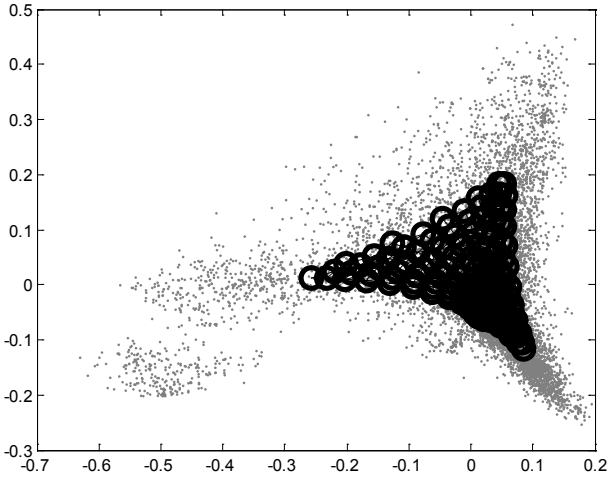


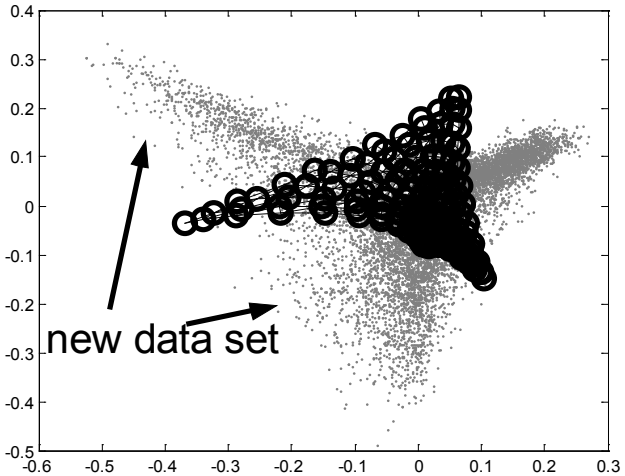


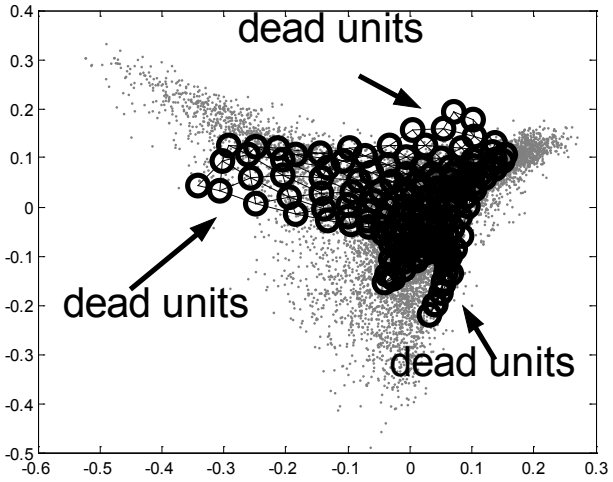
- (a) The thickest line indicates the maximum age of connections.
- (b) Remove any connection whose age is larger than the β portion of the maximum age of connections.
- (c) Remove isolated units.

SOM in a Non-Stationary Environment

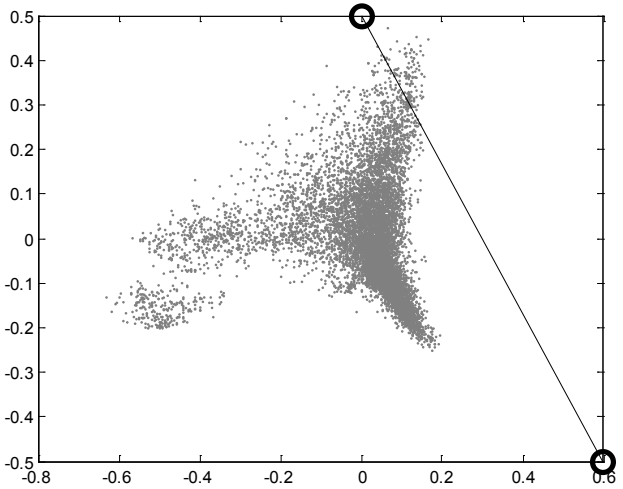


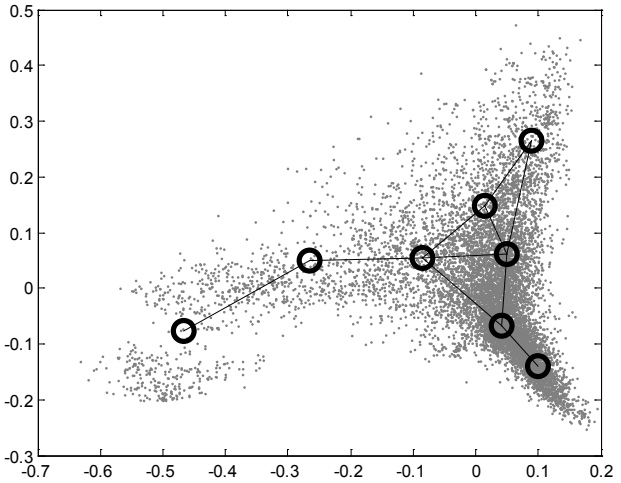


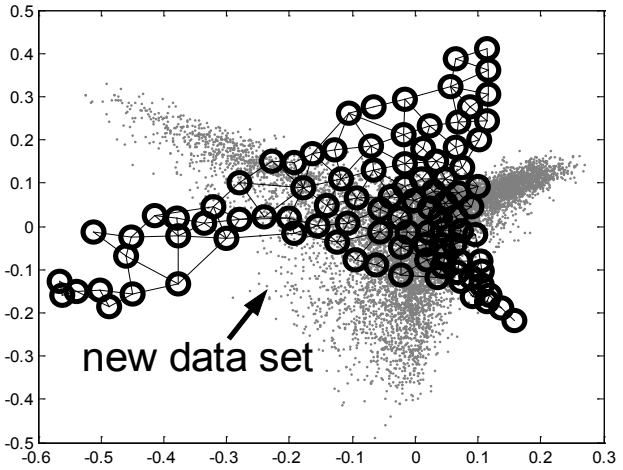


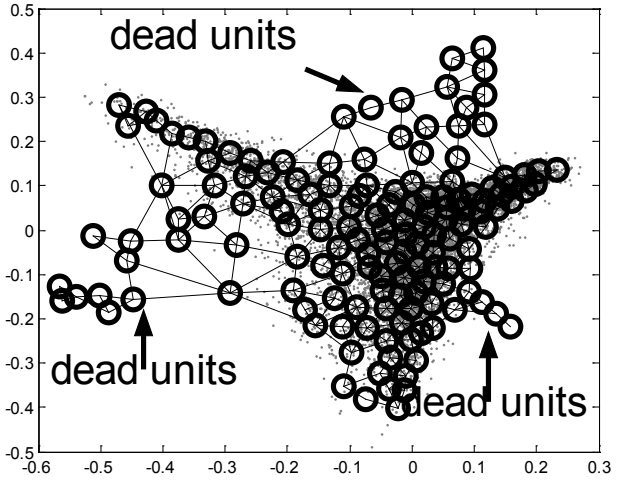


GNG in a Non-Stationary Environment

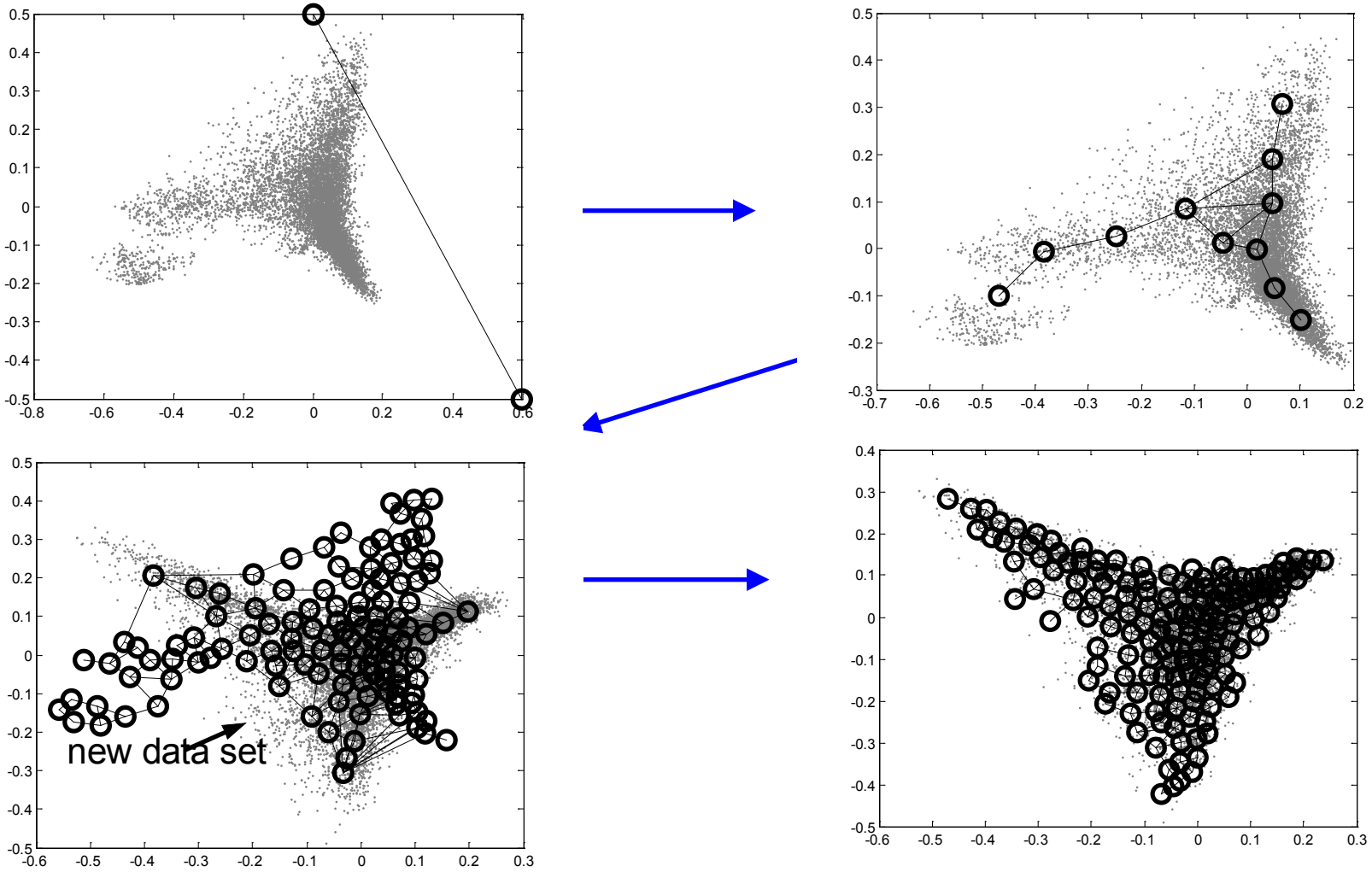




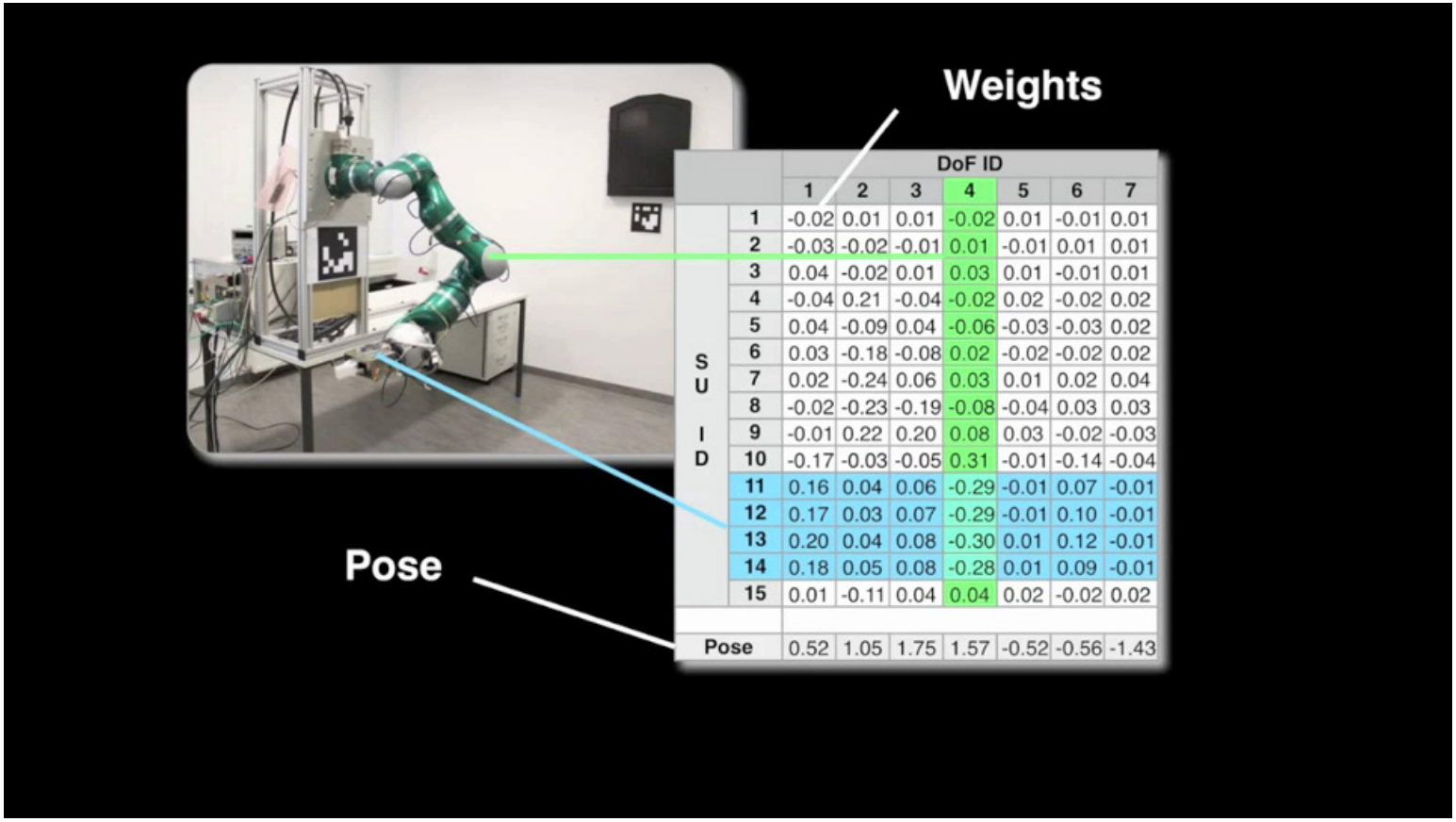




DASH in a Non-Stationary Environment



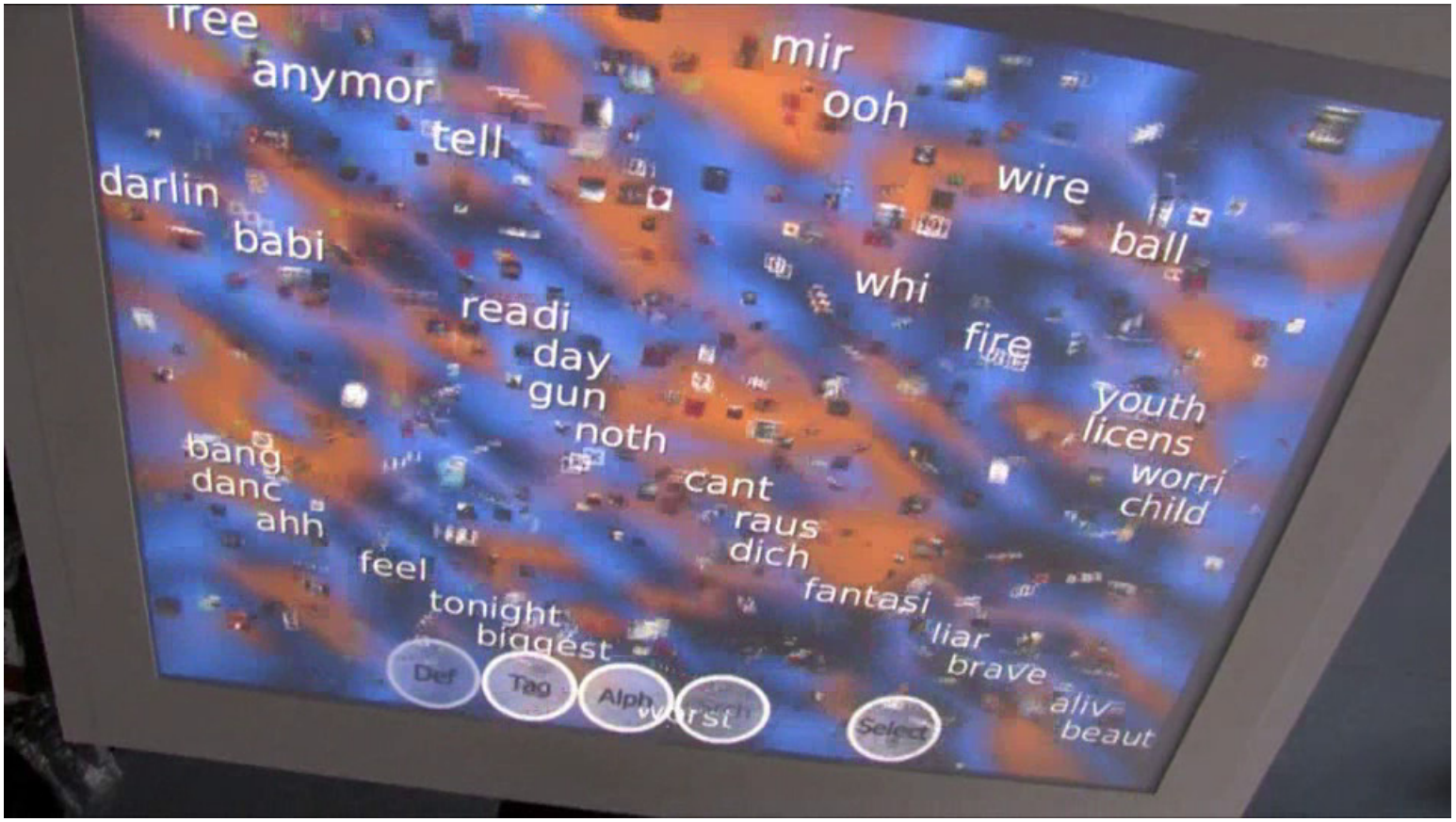
Example: Self-Organizing Sensory-Motor Map for Low-Level Touch Reactions



SUID	DoF ID 1	2	3	4	5	6	7
1	-0.02	0.01	0.01	-0.02	0.01	-0.01	0.01
2	-0.03	-0.02	-0.01	0.01	-0.01	0.01	0.01
3	0.04	-0.02	0.01	0.03	0.01	-0.01	0.01
4	-0.04	0.21	-0.04	-0.02	0.02	-0.02	0.02
5	0.04	-0.09	0.04	-0.06	-0.03	-0.03	0.02
6	0.03	-0.18	-0.08	0.02	-0.02	-0.02	0.02
7	0.02	-0.24	0.06	0.03	0.01	0.02	0.04
8	-0.02	-0.23	-0.19	-0.08	-0.04	0.03	0.03
9	-0.01	0.22	0.20	0.08	0.03	-0.02	-0.03
10	-0.17	-0.03	-0.05	0.31	-0.01	-0.14	-0.04
11	0.16	0.04	0.06	-0.29	-0.01	0.07	-0.01
12	0.17	0.03	0.07	-0.29	-0.01	0.10	-0.01
13	0.20	0.04	0.08	-0.30	0.01	0.12	-0.01
14	0.18	0.05	0.08	-0.28	0.01	0.09	-0.01
15	0.01	-0.11	0.04	0.04	0.02	-0.02	0.02
Pose	0.52	1.05	1.75	1.57	-0.52	-0.56	-1.43

[http://www.ics.ei.tum.de/]

Organising Song Words with a SOM



[http://www.medien.ifi.lmu.de/]

Summary

- Neural clustering as unsupervised techniques, data grouped by similarities
- Elements in different clusters have different characteristics
- Outliers can be classified as separate clusters or force a cluster to accept it