Master of Technology

Unit 2/6: Computational Intelligence I

Neural Network Architectures (II)

Dr. Zhu Fangming
Institute of Systems Science
National University of Singapore

Objective

- To introduce important NN architectures with unsupervised learning

Outline

- Clustering and unsupervised learning
- Kohonen’s Self Organizing Maps (SOM)
- Vector Quantization (VQ)
- Learning Vector Quantization (LVQ)
- Adaptive Resonance Theory Networks (ART)

Knowledge Discovery by Clustering

- **Clustering is one of the main tasks/aims of Pattern Recognition and Knowledge Discovery in Databases (KDD or Data Mining).**

- **Learning by discovery begins with completely unorganized data where instances are not labelled or classified by any teacher. This form of learning is a typical example of unsupervised learning. It usually uses a set of heuristics which implicitly organize the information in the data.**

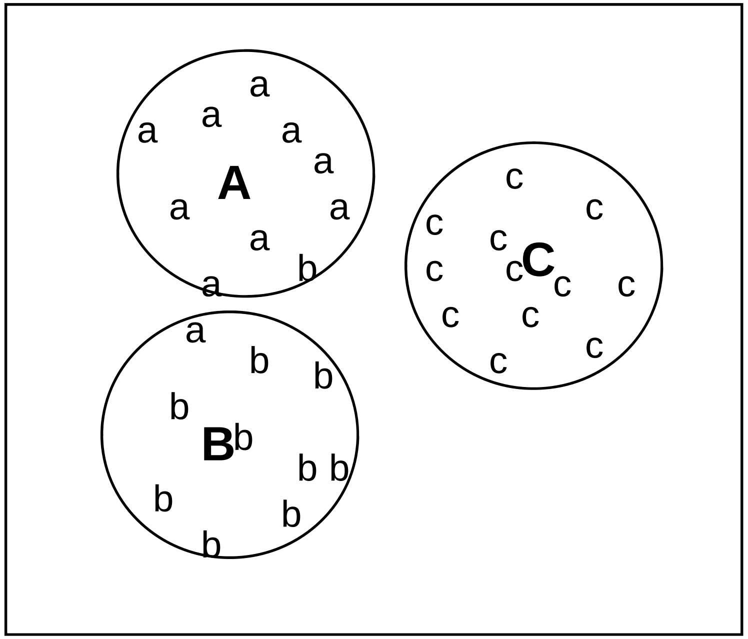
- **A discovery program often looks for regularities. *Regularity* is often defined by satisfaction of certain constraints which can be predicated in symbolic languages or mathematical relations in numerical languages.**

Clustering Types

- Clustering is an important approach for knowledge discovery.
- Clustering techniques:
    - » Conceptual clustering
    - » Numerical clustering
        - ♦ Hierarchical clustering
        - ♦ Non-hierarchical clustering
    - » Statistical approaches
        - ♦ Hard C-Means/K-Means
    - » Neural Network approaches
        - ♦ Kohonen’s Self Organizing Feature Map (SOM)
        - ♦ ART methods
    - » Fuzzy variants
        - ♦ Fuzzy C-Means
        - ♦ Fuzzy Kohonen Clustering Network
        - ♦ Fuzzy ART

Example of Data Clustering

- Clusters A, B, and C
  - » C has been clustered correctly
  - » But cluster A and B have some objects clustered incorrectly



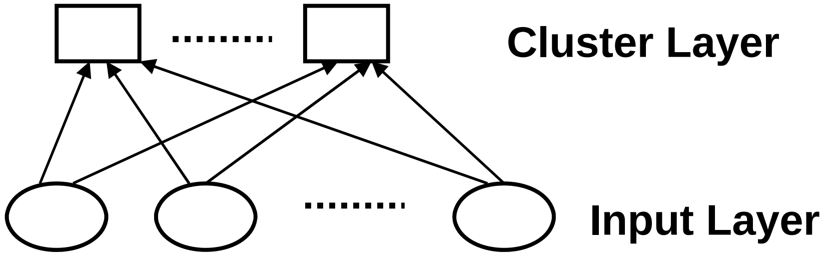
Clustering is basically a practice under unsupervised basis although some background knowledge is needed to determine similarity / distance measure

Adaptive Clustering

- **Adaptive clustering:**

  the number of clusters to be formed is not given

- **NN for clustering**
  - » **common strategy: winner-takes-all (a competitive learning)**
  - » **Basic architecture of a clustering network:**
    - ♦ **two layers: input layer and output layer**



- **NN for adaptive clustering**
  - » **Kohonen's SOM**
  - » **ART network**

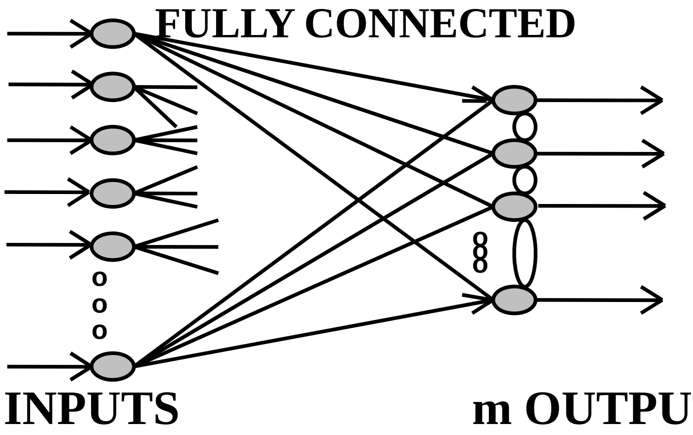
Unsupervised Learning

- Unsupervised learning — training patterns are input values only, they do not include target values to compare to the net's outputs. The network must discover some ‘structure’ in the training data.
- Types of structure which an unsupervised network may discover in the data
  - » Similarity
    - ♦ how similar new input patterns are to typical patterns seen in the past
  - » Principal Components
    - ♦ Principal Component Analysis (PCA) to be briefly introduced later
  - » Clusters
    - ♦ identify how correlated patterns group together by class
  - » Prototypes
    - ♦ form categories based on the correlation in the input patterns and give prototype patterns as output
  - » Feature Map
    - ♦ output units with a fixed geometrical arrangement might form topographic maps of the input patterns

Competitive Network Operation

- **Signals feed forward from input nodes and feed lateral among output neurons, a form of recurrent feedback**
- **A single-layer network architecture with *n* inputs and *m* output units, one output for each class or category**
- ***m prototype* weight vectors w_i, *i* = 1, 2, ..., *m* correspond to the *m* classes. The network classifies each input pattern x as belonging to class *i* iff**

$|w_i - x| \le |w_k - x|$ k=1, 2, ..., m, $k \neq i$ (find the most similar class)



COMPETITIVE "WINNER-TAKES-ALL“ NETWROK

Competitive Network Operation (cont.)

Competitive learning rule

1) Initialize the weights to small random values

2) Find the *k* (*k*≥1) centres $\mathbf{w}_c$ from the *m* prototypes,

$$|\mathbf{w}_c - \mathbf{x}| = \min_i \{|\mathbf{w}_i - \mathbf{x}|\}$$

where

$$|\mathbf{w}_i - \mathbf{x}| = \left[\sum_{j=1}^{n} \left(w_{ij} - x_j \right)^2 \right]^{1/2}$$

3) Update the weights using (update the winner only)

$$\Delta w_{ij} = \alpha \left(x_j - w_{ij} \right)$$

This moves the winning weights toward the input pattern centres ($\alpha > 0$)

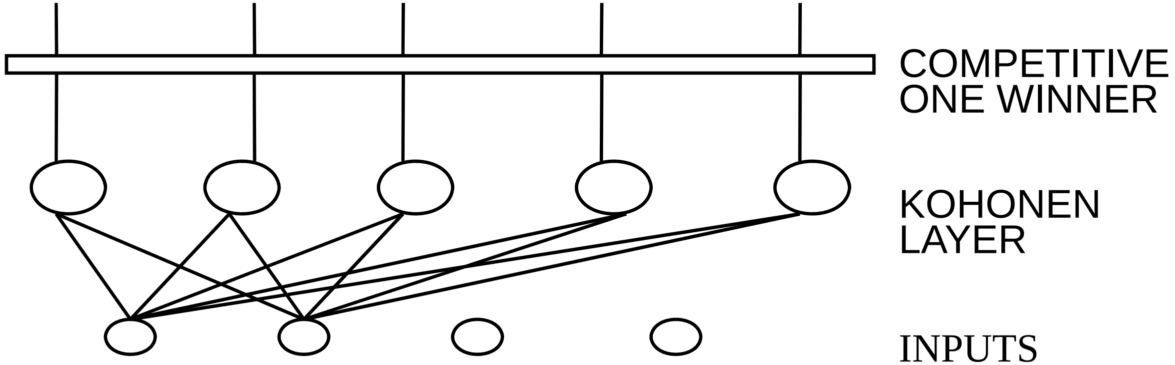
Self-Organizing Feature Map (SOM, Kohonen) Network

- **Feature mapping converts patterns of arbitrary dimension (the pattern space) into the response of one- or two-dimensional arrays of neurons (the feature space).**
- **Kohonen learning rule is one approach to design a self-organizing feature map network**
- **SOM network**
  - » **Consists of a group of geometrically organized neurons in one- or two-dimensions or even higher dimensions.**
    - ♦ **A one-dimensional network is a single layer of units arranged in a row**
    - ♦ **In two-dimensional network, the units are arranged as a lattice array.**
  - » **A form of unsupervised competitive learning where winner shares the gains with neighbours to produce activation zone**

When only the winning neuron participates in the SOM learning process, the array learns to perform *vector quantization* (discussed later)

Simple SOM (Kohonen) Network
— One-dimension —

- A simple 1-dimensional Kohonen (SOM) network
- Input pattern vector fully connected to all neurons
- Lateral interactions between neurons constrain activations to spatially bounded ‘excitation zone’

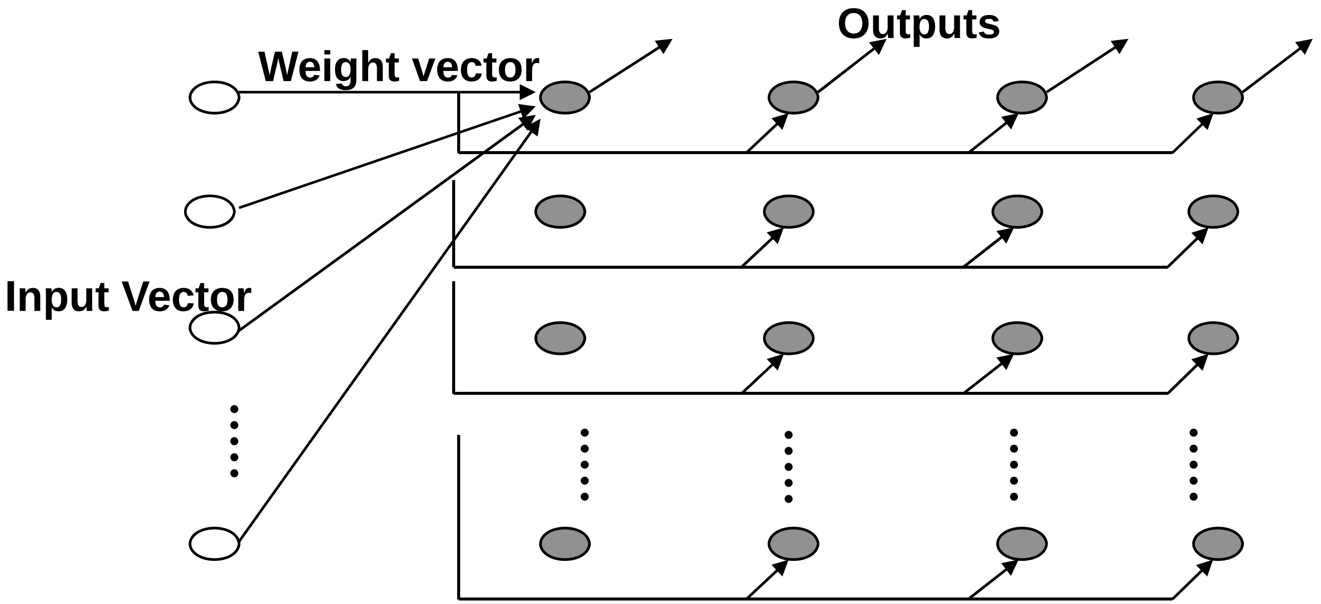


NEURONS COMPETE AND ONLY ONE "WINS"

SOM (Kohonen) Network (cont.)

— Two-dimension —

- A 2-dimensional Kohonen discrete lattice network
- Input vector patterns x are fully distributed to each node in the discrete lattice through adjustable weight vectors w_r
- The node with weight vector closest to the input vector becomes the "excitation centre" winner for the lattice



Kohonen Learning Algorithm
— Procedure —

Procedure:

1) **Initialize all weights w_r to random numbers following uniform distribution (0,1)**
2) **Apply an input signal vector x to the network**
3) **Select the winning output unit as the one with the smallest dissimilarity measure (or largest similarity measure) among all weight vectors w_i and the input vector x**

$$\|\mathbf{x} - \mathbf{w}_r\| \leq \min_{r'} \|\mathbf{x} - \mathbf{w}_{r'}\|$$

4) **Update**

the weights of the winner unit:

$$\mathbf{w}_{\mathbf{r}}^{\mathbf{new}} = \mathbf{w}_{\mathbf{r}}^{\mathbf{old}} + \alpha \cdot \mathrm{h}_{\mathbf{rr'}} \cdot \left(\mathbf{x} - \mathbf{w}_{\mathbf{r}}^{\mathbf{old}}\right)$$

the neighbouring

$$\mathbf{w}_{\mathbf{r'}}^{\mathbf{new}} = \mathbf{w}_{\mathbf{r'}}^{\mathbf{old}} + \alpha \cdot \mathrm{h}_{\mathbf{rr'}} \cdot \left(\mathbf{x} - \mathbf{w}_{\mathbf{r'}}^{\mathbf{old}}\right)$$

where $h_{rr'}$ is a neighborhood function with maximum value centered at the winning neuron *r* and decreases to zero as the distance between *r* and neighboring units r' increase (i.e. $h_{rr'}$ is defined in terms of the distance between r and r').

5) **Repeat steps 2 through 4 until the network weights stabilize**
6) **Reduce the neighbourhood and learning rate parameters and iterate steps 2 to 5 if necessary.**

Kohonen Learning Algorithm (cont.)
— Assumptions —

Assumptions for the above algorithm:

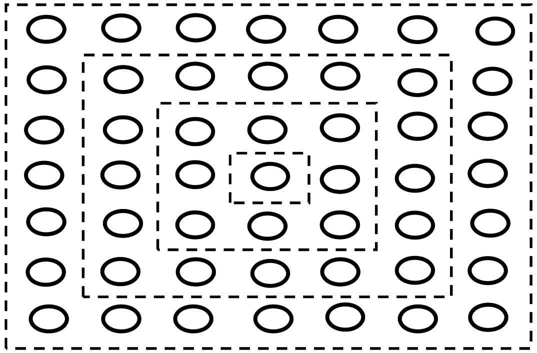
- Input data is normalized
- α is sufficiently small to prevent "overadaptation"

  (α is usually reduced to zero during training)
- The input sample set (x_1, x_2, ..., x_n) covers the unit sphere with nonzero probability at all points

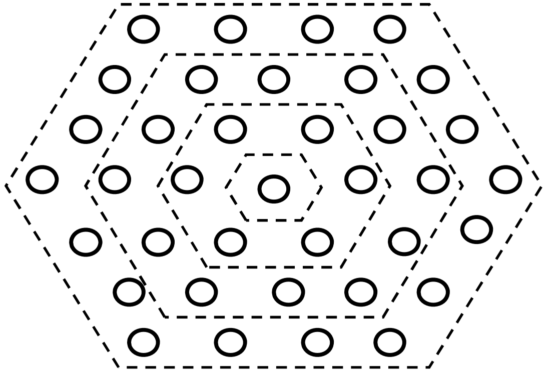
Kohonen Learning Algorithm (cont.)
— Calculations —

Activation Neighbourhood can be

- **Square**



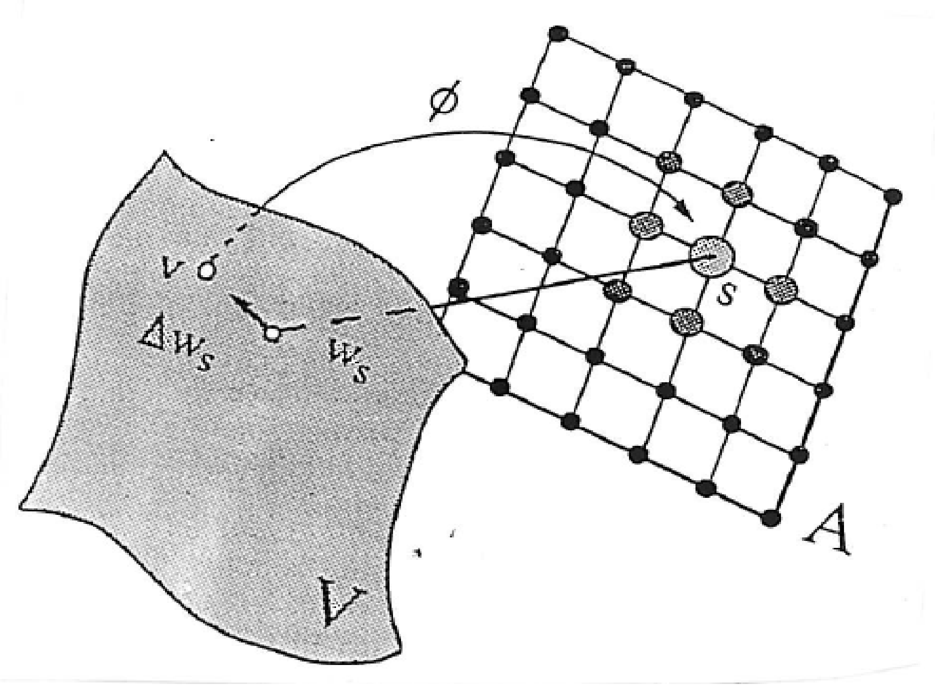
- **Hexagon**



- **Smooth functions such as Gaussian (bell shaped) or other types may also be used.**

SOM Mapping Operation

- Mapping the input signal space onto the NN lattice
- Weight vectors of neurons in the neighborhood of the winning neuron *s*, are shifted towards the input vector value. Those nearer to *s* have weights shifted more and those farther away shifted less



Simulations of SOM

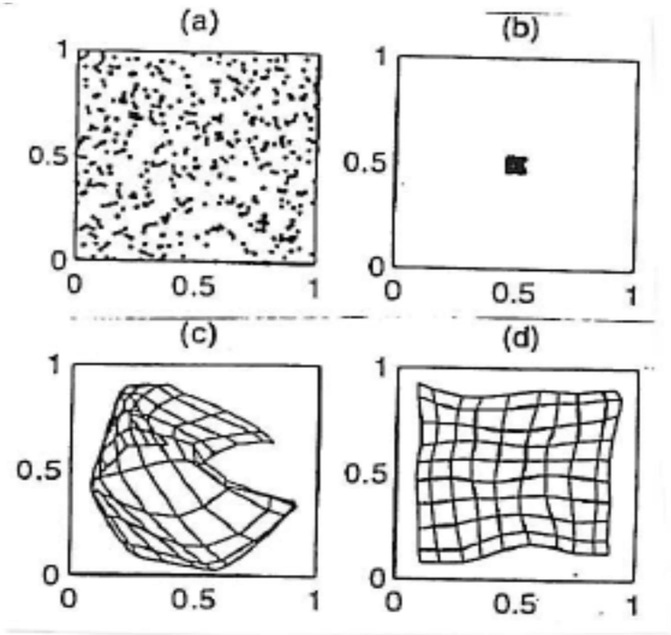
- Mapping two inputs x_1 and x_2 onto a SOM array.
- Input patterns are chosen randomly from the unit square
- Line intersections on the maps specify weight vector values for a single neuron
- Lines between nodes on the graph merely connect weight points for neurons that are topologically nearest neighbors

(a) input data uniformly distributed within [0, 1]× [0, 1]

(b) initial weights

(c) weights after 30 iterations

(d) weights after 1000 iterations



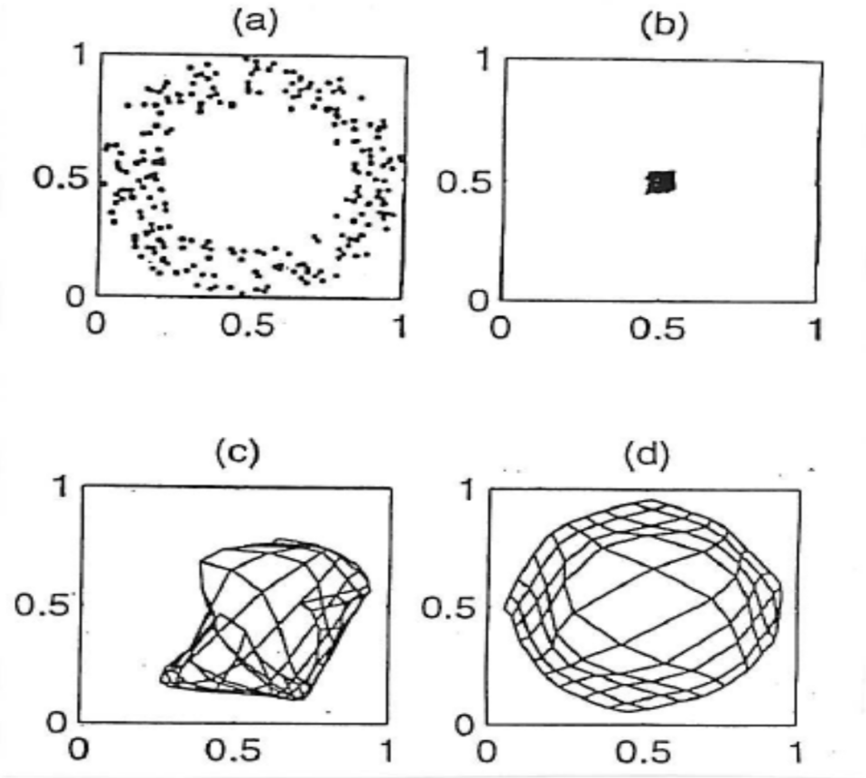
Simulations of SOM (cont.)

(a) input data uniformly distributed within [0, 1]× [0, 1]

(b) initial weights

(c) weights after 30 iterations

(d) weights after 1000 iterations



Advantages of SOM

- It is viable alternative to C-Means clustering
- It does so more elegantly, by removing the main drawback of C-Means - specifying the ***number of clusters upfront***
- It provides a method of data compression
- When used for labeled data, this acts as a classifier too.
- Most importantly, it provides the user with much wanted ***'data visualization'*** feature, which is absent in many of the competing algorithms
- This feature makes it eminently suitable for ***DATA MINING*** applications.
- Nowadays, SOM is used together with MLFF-BP or in the unsupervised part of the RBF to come out with better results
- Above all, it is surprisingly simple to implement while being extremely powerful

Some Applications of SOM

Optimization

- One-dimensional SOM networks have been used to successfully solve optimization problems such as the traveling salesman problem (TSP)

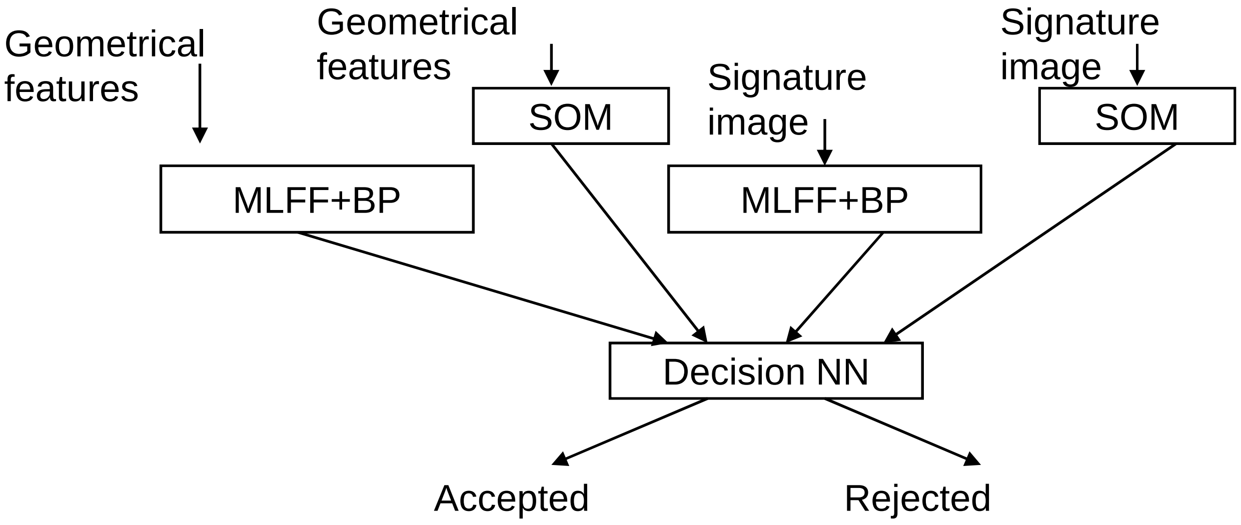
Control

- Robot arm control
  - » as a part of a robot control system, SOM network performs “data fusion” to provide the input data to the robot

Some Applications of SOM (cont.)

Pattern Recognition

- **Handwritten signature authentication**
  - » **uses a hybrid neural network system with SOM networks and MLFF network**
    - ♦ **2 SOM networks were used for initial signature classification into similar sets**
    - ♦ **2 MLFF+BP for authentication**
    - ♦ **1 supervised learning NN for final decision**



Some Applications of SOM (cont.)

Handwritten signature authentication (cont.)

- **training data**
  - ♦ **a database of 6,000 digitised checks obtained from the French Post Office**
  - ♦ **the forgeries used for training were signatures randomly taken from the same database belonging to the same signature class as the genuine signature used in the training**
- **authentication is done in stages**
  - ♦ **Pre-classification by two SOM networks**
    - **using geometrical parameters and outline as inputs to pre-classify the signatures into similarity groups**
  - ♦ **authentication by two MLFF networks**
    - **have the same inputs as SOM networks and single output**
  - ♦ **final decision of reject / accept**
    - **has inputs from the four previous stage networks to output a binary value: accept / reject**

Some Applications of SOM (cont.)

- **Knowledge discovery from databases -— WEBSOM**

  (Teuvo Kohonen, “*New Lines in the Study of Self-Organizing Maps*”, 6th International Conference on Soft Computing, Iizuka, Japan, 2000)

  - » **The newest version of WEBSOM**
    - ♦ **selected a database of 6,840,568 patent abstracts that were available in electric form and written in English.**
    - ♦ **These patents were granted by the US, European, and Japan patent offices and stored in two databases: the “First Page” database (1970-1997), and “Patent Abstracts of Japan” (1976-1997).**
    - ♦ **The average length of each text was 132 words. The size of the SOM was 1,002,240 neurons. (it had 20 times as many words as all the 34 volumes of Encyclopaedia Britannica taken together)**

Some Applications of SOM (cont.)

- **WEBSOM (cont.)**
  - » **The basic idea is to present each document (such as a patent abstract) by its word histogram.**
    - ♦ **From the text, first discard numbers, symbols and stopwords that do not convey the meaning of the text.**
    - ♦ **Very rare and very frequent words are discarded**
    - ♦ **The remaining vocabulary consists of 43,222 words (base forms)**
    - ♦ **The histograms of these words in each document are formed**
    - ♦ **Since the histogram has a too large dimensionality to be considered as input vectors to the SOM, the dimensionality is reduced by forming 500-dimensional projection vector of the histogram**

Some Applications of SOM (cont.)

- WEBSOM (cont.)
  - » The document map is presented to the user as a series of HTML pages that enable the exploration of the map: when clicking on a point on the map display, links to the document database enable reading the content of the document
  - » If the map is large, subset of it can first be viewed by zooming. With the largest maps, three zooming levels have been used before reaching the documents.
  - » User can type a query, or a description of interest, in the form of a short "document". This query is pre-processed and a document vector is formed in the exactly same manner as for the stored documents prior to construction of the map. The "user" vector is then compared with the model vectors of all map units, and the best-matching points are marked with circles on the map display: the better the match, the larger the circle.
  - » With the WEBSOM map of all patent abstract, the search takes only a few seconds.

Vector Quantization (VQ)

Vector Quantization (VQ)

- When only the winning neuron participates in the SOM learning process, the array learns to perform ***vector quantization***
- An important application of competitive learning
- It is concerned with how to divide the input space into disjoint subspaces so that each input vector can be represented by the label of the subspace it belongs to.

Basic VQ Algorithm:

- Determine the neuron with weight closest to the input,

$$\left\| \mathbf{x} - \mathbf{w}_{\mathbf{r}} \right\| \leq \min_{\mathbf{r}'} \left\| \mathbf{x} - \mathbf{w}_{\mathbf{r}'} \right\|$$

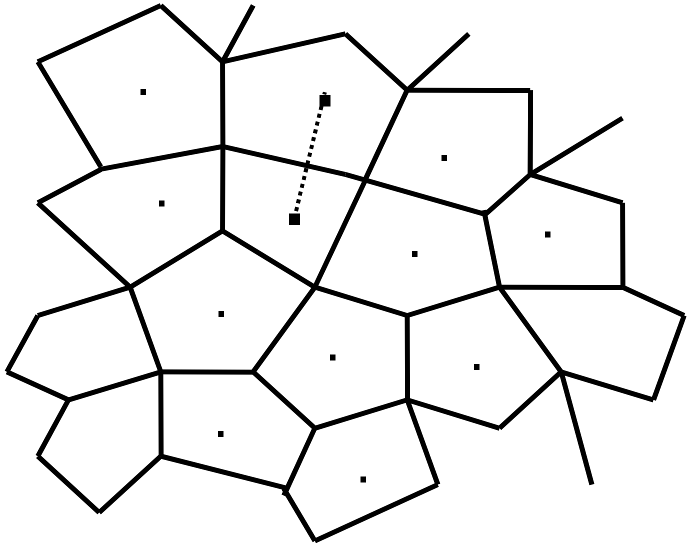
- Adjust the neuron weights as follows:

$$\mathbf{w}_{\mathbf{r}}^{\mathbf{new}} = \mathbf{w}_{\mathbf{r}}^{\mathbf{old}} + \alpha \left(\mathbf{x} - \mathbf{w}_{\mathbf{r}} \right)$$ for the winning neuron ***r***

$$\mathbf{w}_{\mathbf{i}}^{\mathbf{new}} = \mathbf{w}_{\mathbf{i}}^{\mathbf{old}}$$ for $i \neq r$

VQ Mapping

- A two-dimensional portrayal of the partitions and corresponding reference prototype vectors
- Each polyhedral region represents one class and the dots represent the class prototype weight vectors. The boundaries are perpendicular bisector planes of lines joining neighbouring prototype vectors.



Voronoi Sets

Learning Vector Quantization (LVQ)

Vector Quantization with or without supervision

- VQ — Unsupervised learning algorithm
  - » the class partitions are found without supervision
- LVQ — Supervised learning algorithm. The correct class of the input vector x is provided
  - » LVQ learning involves steps
    - ♦ an unsupervised learning data clustering method is used to locate initial cluster centres without using the class information
    - ♦ label each cluster by the voting method (a cluster is labelled class *k* if majority of its data points belong to class *k*)
    - ♦ the class information is used to fine-tune the clusters to minimize the number of misclassed cases
      - for training pattern *x*, find a W_c which is closest to *x* (winning), and update W_c

Learning Vector Quantization (cont.)

Variations on LVQ algorithms

- **LVQ1 (only one winning node selected)**

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) + \alpha\left[\mathbf{x}(t) - \mathbf{w}_c(t)\right]$$

$$\text{if } \mathbf{x} \text{ and } \mathbf{w}_c \text{ belong to the same class}$$

$$\mathbf{w}_c(t+1) = \mathbf{w}_c(t) - \alpha\left[\mathbf{x}(t) - \mathbf{w}_c(t)\right]$$

$$\text{if } \mathbf{x} \text{ and } \mathbf{w}_c \text{ belong to different classes}$$

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) \qquad \text{for } i \neq c$$

$0 < \alpha < 1$ is a constant and may decrease during training

- **LVQ2:**
  - » **attempts to use training data more efficiently by updating *winner* under certain conditions**
  - » **the *two nearest* vectors W_i and W_j are updated together, where x and W_i belong to the same class and x and W_j belong to different classes**
  - » **the updating is the same as LVQ1** (details omitted)

Adaptive Resonance Theory (Carpenter & Grossberg, 1988)

- Theoretical background of ART
  - » ART suggests a solution to the stability-plasticity dilemma (*How can a network retain its stability while still being plastic enough to gainfully adapt in a changing environment*)
  - » it would be easy either to learn new patterns (learning plasticity) or to retain the knowledge of previously learned patterns (learning stability), but difficult to do both
  - » ART uses pattern resonance to solve the difficulty

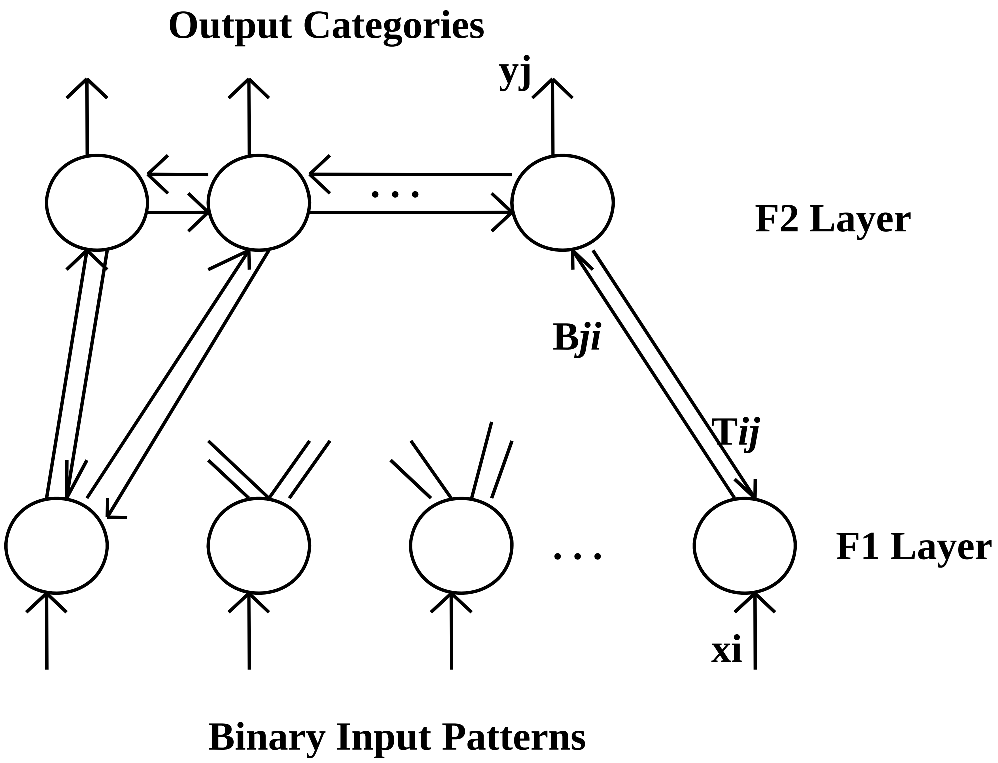
Adaptive Resonance Theory (ART) Networks

ART networks

- **Architecture: two-layer recurrent dynamic network**
- **Applications: pattern clustering, classification**
- **Learning:**
  - » **unsupervised**
  - » **when acquiring new concepts, the old patterns are still kept in the memory (new facts do not "wash away" old ones)**
  - » **continually adapt in a changing environment (learning incrementally without reviewing old instances)**
  - » **map *n*-dimensional input patterns to output categories based on nearest-neighbour grouping. The degree of similarity is adjustable**
  - » **new categories are formed from novel inputs until network exhausts its pool of uncommitted output neurons**

ART-1 Network

- **Fully connected feedforward and feedbackward connections**
- **Lateral connections in the output layer for competitive response and "winner-takes-all" operation**
- **Recurrent computations operation with winner learning**



ART-1 Algorithm

Conceptual description (without calculation details):

1. **Initialization**
   - **Initialise the vigilance parameter ρ so $0 \le \rho \le 1$**
   - **Initialise the set P of prototype vectors**
2. **Apply new input vector**
   - **Let X :=[next input vector]**
   - **Let P' :=P be the set of candidate prototype vectors**
3. **Find the closest prototype vector Pi from P' which maximizes** $\dfrac{P_i \bullet X}{\|P_i\|}$
4. **Test for vigilance acceptability** $\dfrac{P_i \bullet X}{\|X\|} \ge \rho$

4.1) If acceptable, then X belongs to P_i's cluster.

Modify P_i to be more like X by $P_i = P_i \bullet X$

output *i* ; go to step 2

4.2) If not acceptable, P'=P'- P_i ,

4.2.1) if P' is empty, then create a new cluster P_j equal to X; $P = P \cup \{P_j\}$;

output j; go to step 2

4.2.2) otherwise, go to step 3

An Application of ART-1
— Neural Information Retrieval System —

- **Application background**
  - » **Boeing Aeroplane Company has thousands of part designs. Engineers often inadvertently redesign the same or similar part. If part designs can be stored in a database and retrieved from descriptions, considerable savings can be realized.**
  - » **An ART-1 based retrieval system was designed to classify and store part designs for later retrieval to eliminate wasteful redesigns**

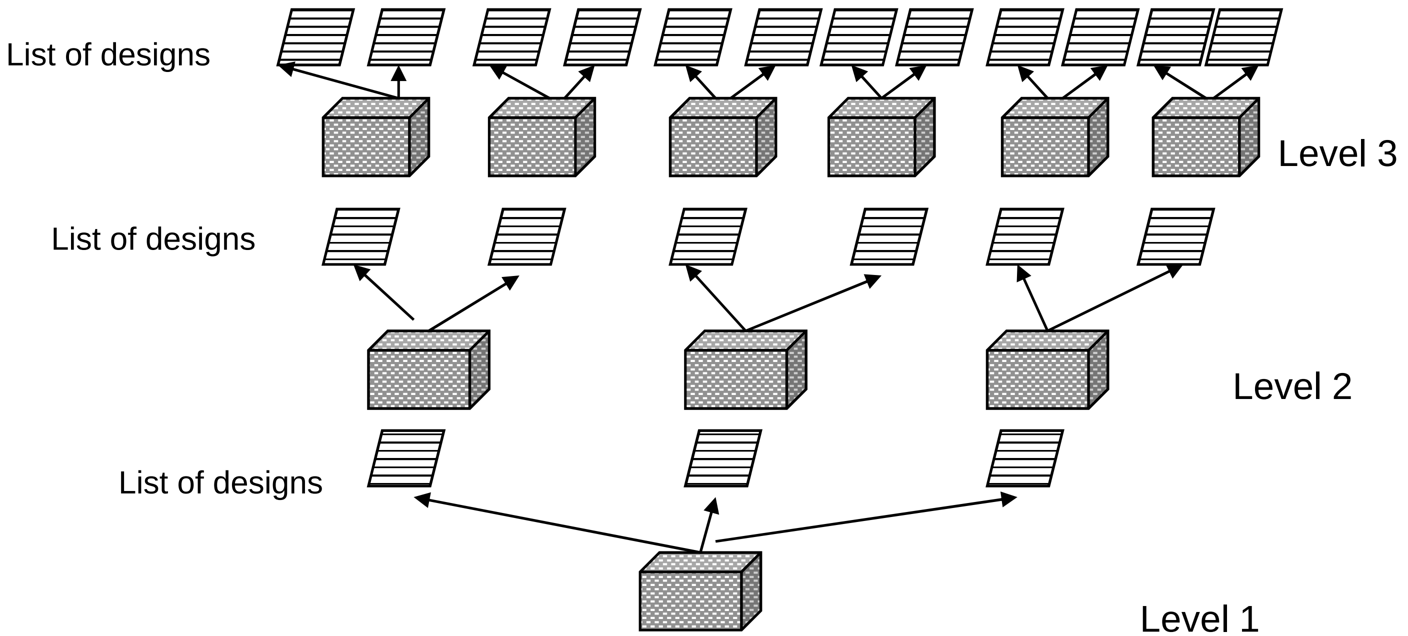
- **Over 20,000 different designs are stored in the completed system. Engineers can specify the level of granularity of retrieval in the design.**

An Application of ART-1 (cont.)
— Neural Information Retrieval System —

- **Interconnected ART-1 networks**
  - » **The system is organized as a hierarchy of three levels of ART network modules where each level corresponds to a group of features.**
    - ♦ **The lowest level of networks — select stored designs that had been clustered into groups on the basis of *shapes*.**
    - ♦ **The next level of networks select on the basis of *bend lines* in the parts**
    - ♦ **The final level selects on the basis of *holes* in the parts**
  - » **This architecture gives user the possibility to discriminate on shapes alone, on shape and holes, on shape and bends, or shape, holes and bends.**
  - » **Under the clusters formed, individual designs are stored for retrieval.**

An Application of ART-1 (cont.)
— Neural Information Retrieval System —

- **Interconnected ART-1 Networks**



An Application of ART-1 (cont.)
— Neural Information Retrieval System —

- Function of the system
  - » When a query is made
    - ♦ The lowest level puts the new design into one of its clusters.
    - ♦ Clusters in this level represent the most general abstraction of designs stored. When a winning cluster is selected at the first level, the modules in the next level associated with this group are activated.
    - ♦ The process repeats.
  - » The vigilance parameters permit the user to vary the degree of match chosen on each of the features selected.
    - ♦ a range of designs can be retrieved from a large number of loosely similar ones to a small set of highly similar designs.
  - » Retrieval time on a PC is between 30~45 seconds

Summary of ART

- **Ability of ART networks**
  - **limited ability in generalization because ART networks lack the hidden layer of neurons which perform feature recognition and extraction**
  - **good to create a new pattern class in its knowledge base on arrival of a novel pattern**
- **The underlying theoretical basis is more complex than that of the feedforward networks.**
- **Two interesting properties:**
  - **real-time learning (incremental)**
  - **self-organization**
- **More sensitive to data noise than SOM and other clustering methods**

Summary of Unsupervised Learning

- Unsupervised network computing methods permit the grouping or clustering of patterns into classes without any prior knowledge of the pattern class memberships
- Training is greatly simplified: it is not necessary to provide target values for the input patterns in advance
- Grouping or clustering unknown patterns into distinct classes having similar characteristics is a common problem with applicability to diagnosis, taxonomic classification, storage and retrieval, data compression and many others
- The main types of unsupervised networks studied in this course are ART, SOM and VQ (LVQ) networks. Although they differ in structure and operation, they can often be used for similar applications.