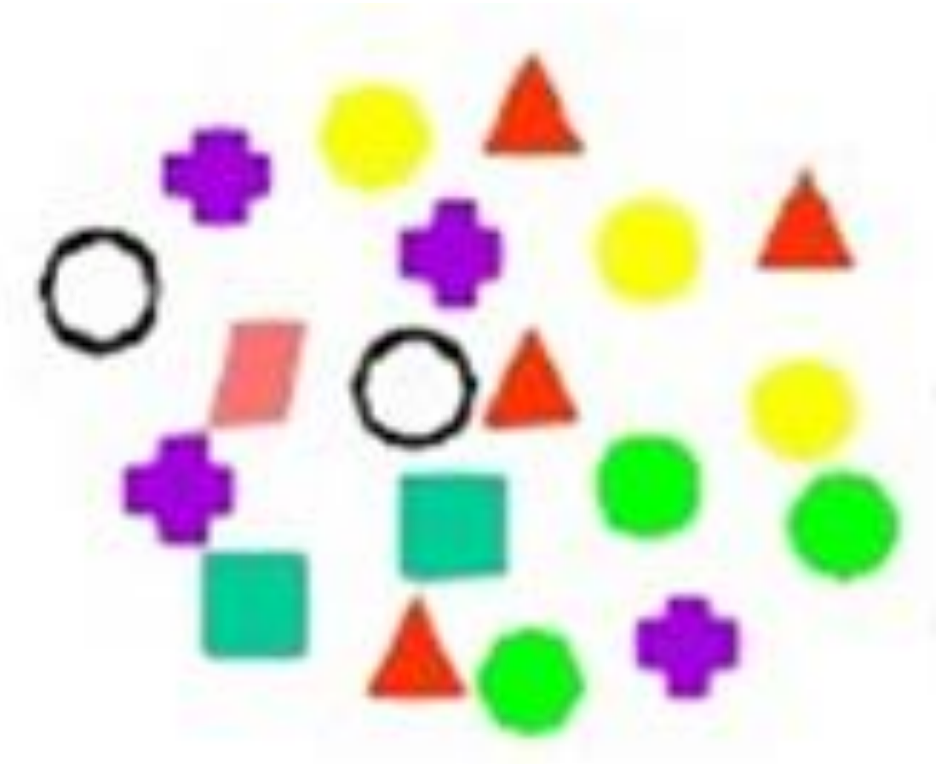
Computational Intelligence

Unit # 8

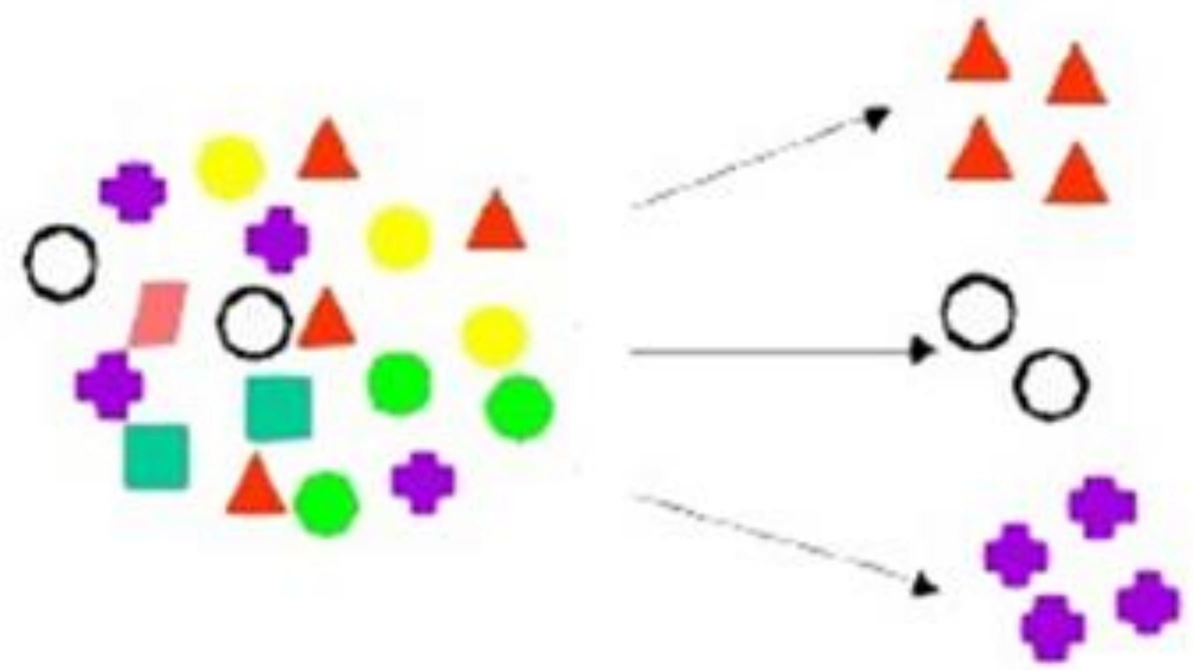
Acknowledgement

- The slides of this lecture have been taken from the following URLs:
  - http://www.ai-junkie.com/ann/som/som1.html
  - http://blog.yhat.com/posts/self-organizing-maps-1.html

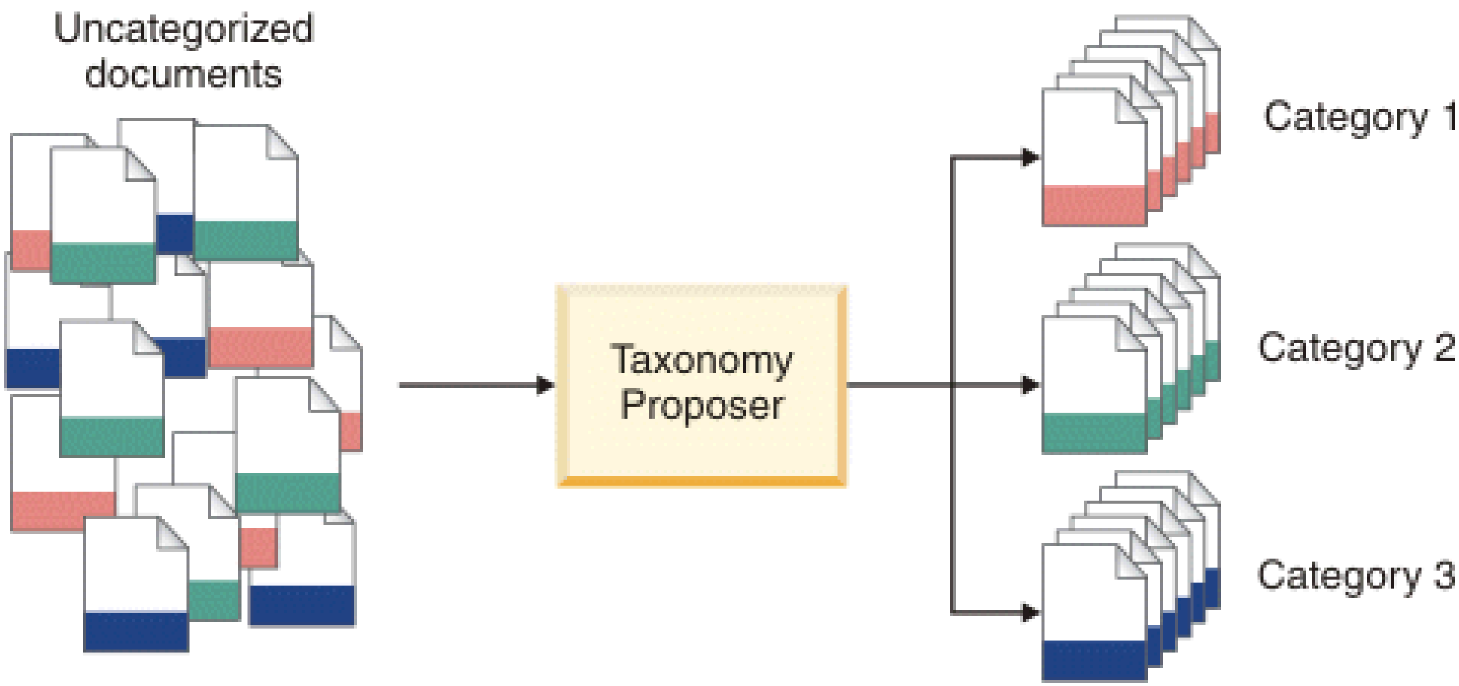
Know your data!



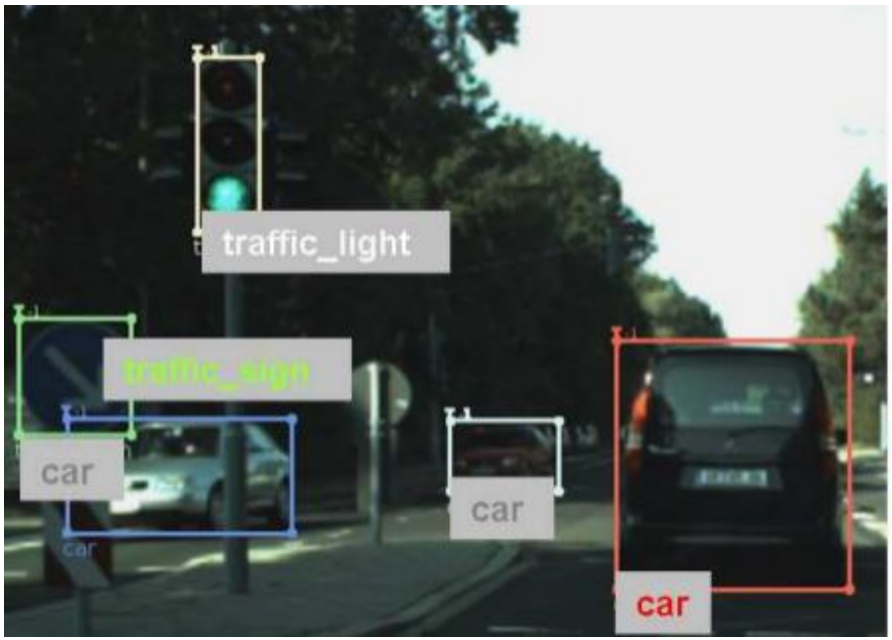
Unsupervised Learning

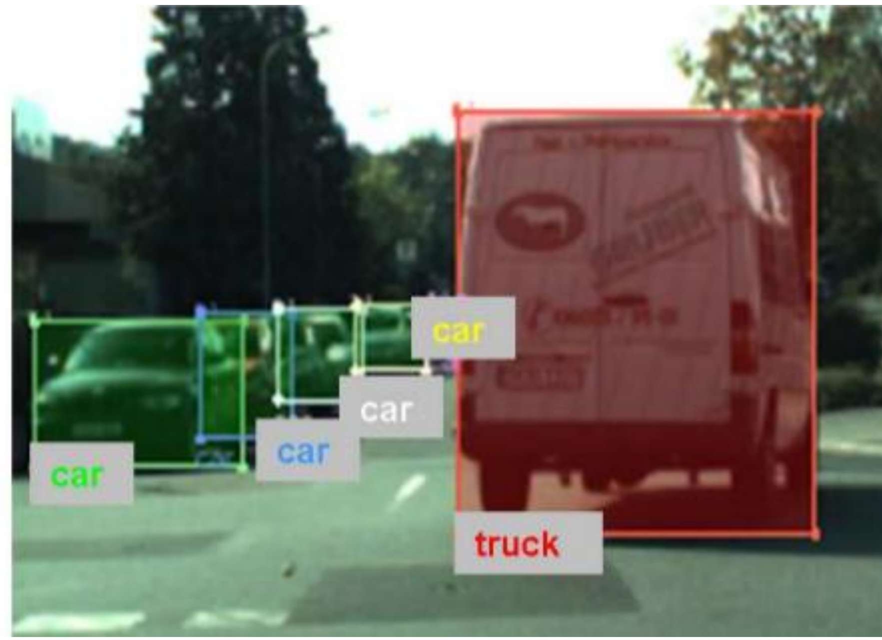


Documents Sorting



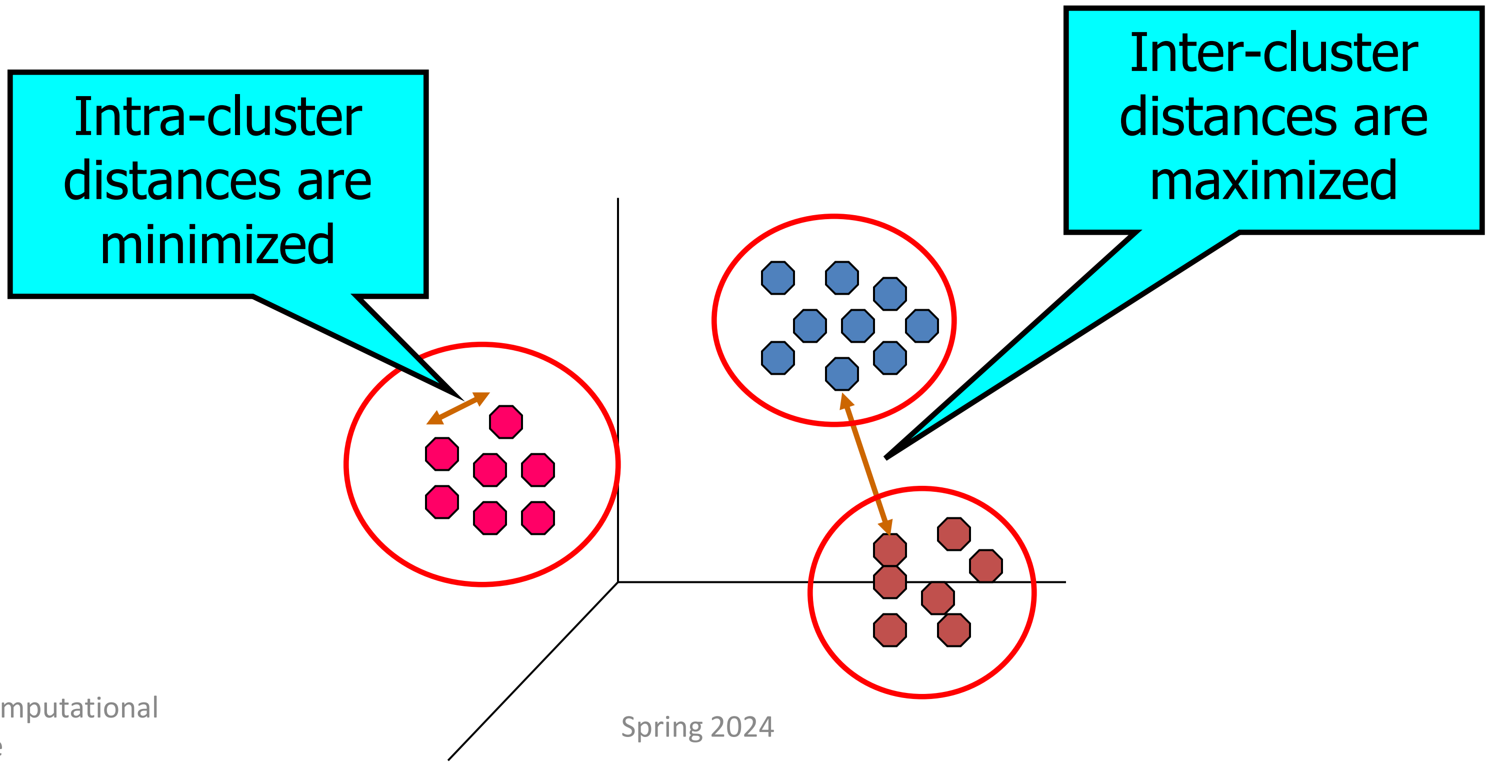
Object Detection





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



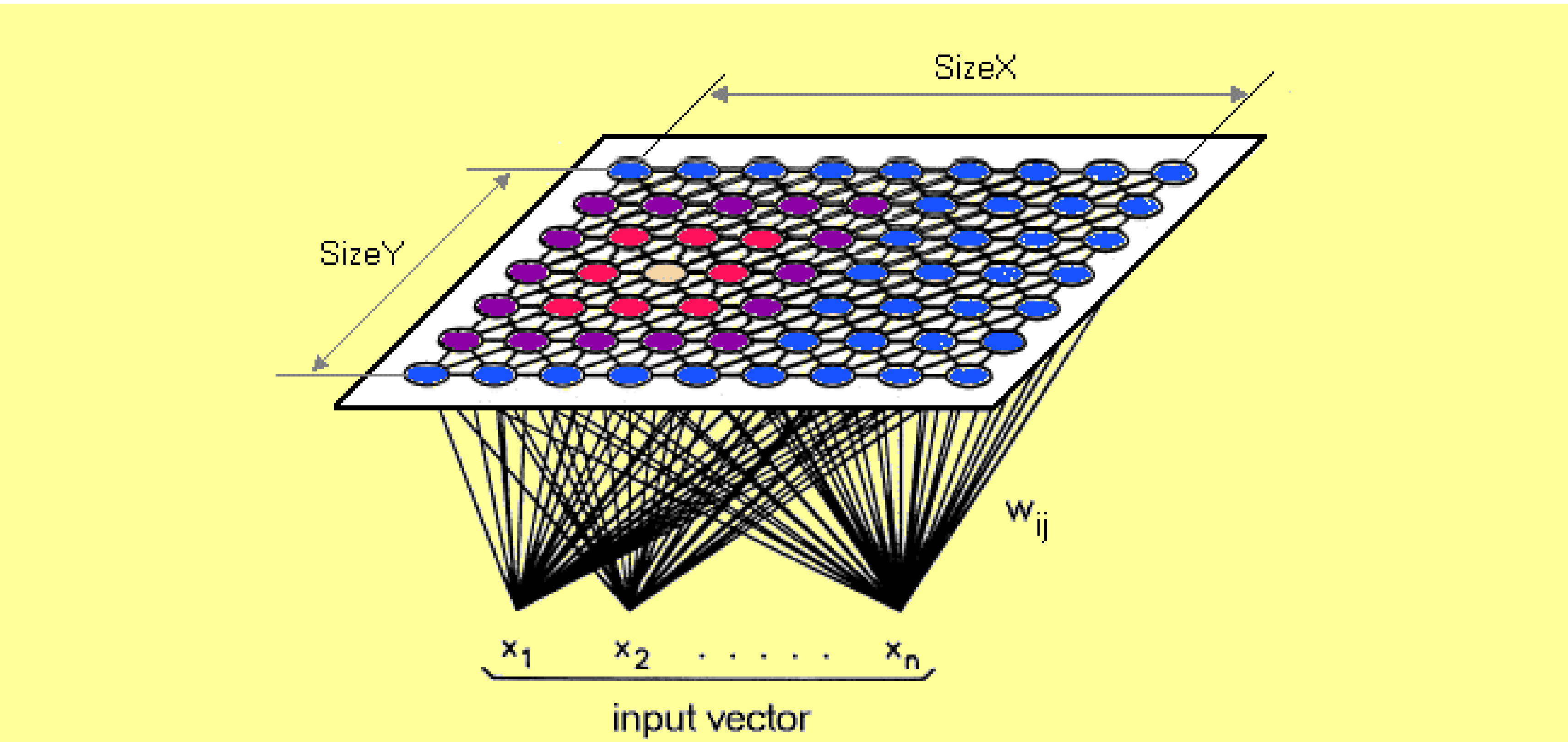
Self Organizing Map

- A **Self-Organizing Map**, or SOM, falls under the domain of unsupervised learning in Neural Networks.
- Its a grid of *neurons*, each denoting one *cluster* learned during training.
- A **self-organizing map** (**SOM**) or **self-organizing feature map** (**SOFM**) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a **map**, and is therefore a method to do dimensionality reduction.

Self Organizing Map

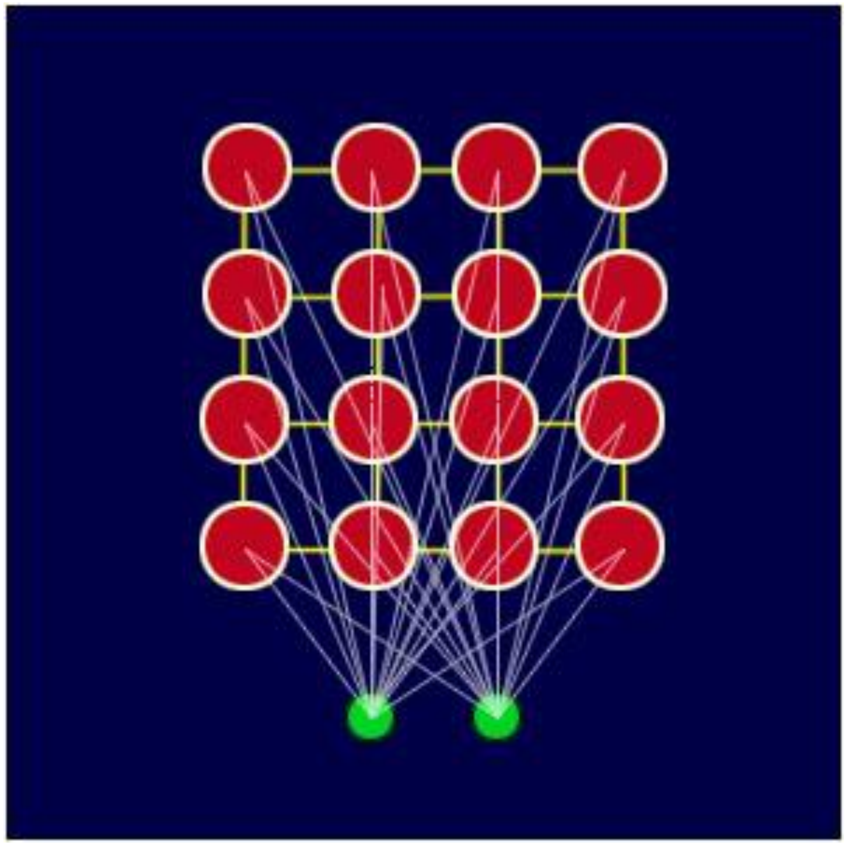
- Traditionally speaking, there is no concept of neuron ‘locations’ in ANNs. However, in an SOM, each neuron has a location, and *neurons that lie close* to each other represent clusters with *similar properties*. Each neuron has a weightage vector, which is equal to the centroid of its particular cluster.

SOM



SOM

- There are three main ways in which a Self-Organising Map is different from a “standard” ANN:
  – A SOM is not a series of layers, but typically a 2D grid of neurons
  – They don’t learn by error-correcting, they implement something called competitive learning
  – They deal with unsupervised machine learning problems
  – Competitive learning in the case of a SOM refers to the fact that when an input is “presented” to the network, only one of the neurons in the grid will be activated. In a way the neurons on the grid “compete” for each input.



SOM Setup

- Your data contains vector X of n dimensions:

  $x_{1,}\ x_{2,}\ x_{3,}\ x_4, \ldots\ldots\ldots\ldots\ldots\ldots\ldots x_n$

- Each neuron contains a weight vector of W dimensions:

  $w_{1,}\ w_{2,}\ w_{3,}\ w_4, \ldots\ldots\ldots\ldots\ldots\ldots\ldots w_n$

- *There are no lateral connections between neurons.*

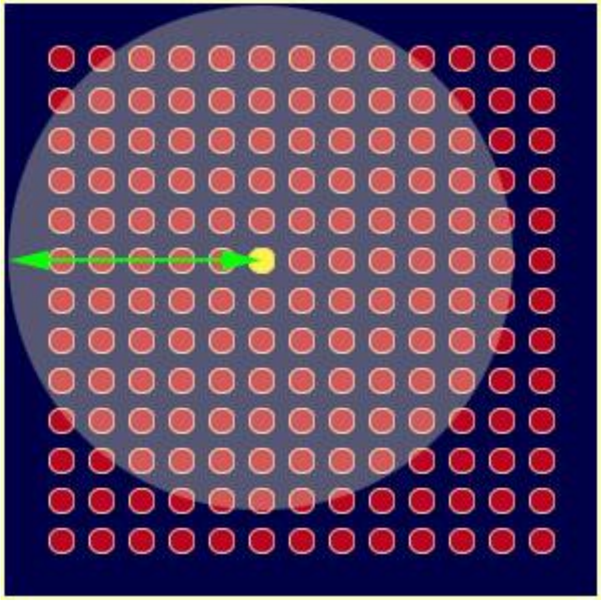
Training SOM

Training occurs in several steps and over many iterations:

1. Each node's weights are randomly initialized.
2. A vector is chosen at random from the set of training data and presented to the lattice.
3. Every node is examined to calculate which one's weights are most like the input vector. The winning node is commonly known as the Best Matching Unit (BMU).
4. The radius of the neighborhood of the BMU is now calculated. Any nodes found within this radius are deemed to be inside the BMU's neighborhood.
5. Each neighboring node's (the nodes found in step 4) weights are adjusted to make them more like the input vector. The closer a node is to the BMU, the more its weights get altered.
6. Repeat step 2 for N iterations.

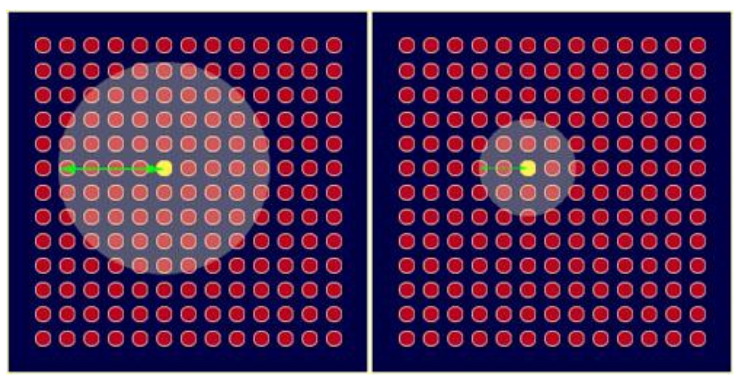
Neighborhood Function

- Each iteration, after the BMU has been determined, the next step is to calculate which of the other nodes are within the BMU's neighbourhood. All these nodes will have their weight vectors altered in the next step.



Neighborhood Function

- A unique feature of the Kohonen learning algorithm is that the area of the neighborhood shrinks over time. This is accomplished by making the radius of the neighborhood shrink over time.



[SOM tutorial part 1 (ai-junkie.com)]

Adaptation

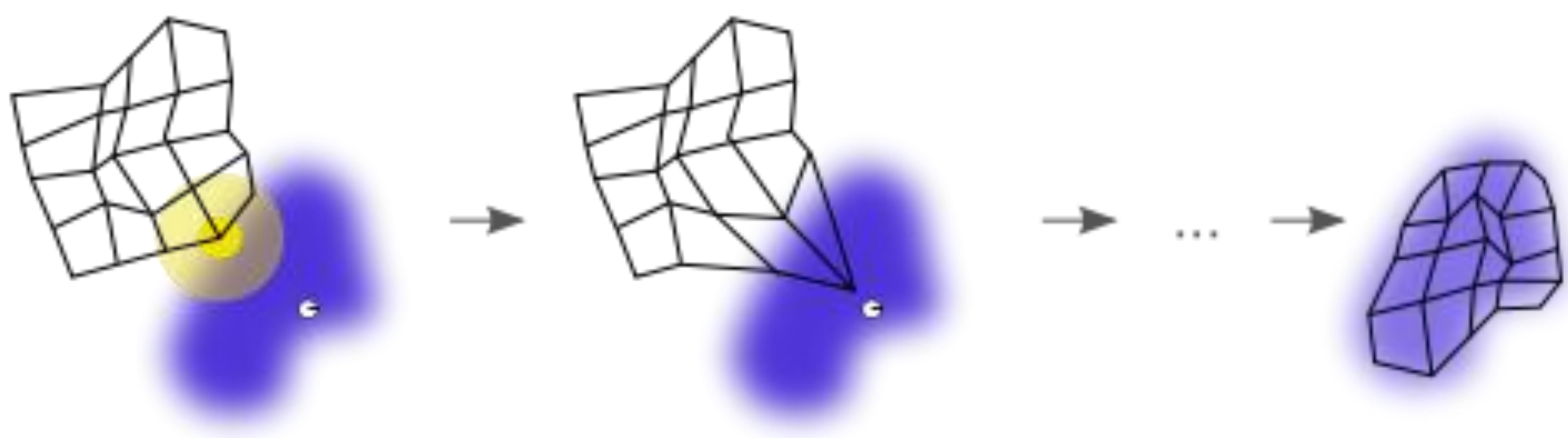
$$W(t+1) = W(t) + \Theta(t)L(t)(V(t) - W(t))$$

- Where L is learning rate and theta is neighborhood function. Both of them are time variant.

SOM in Visualization

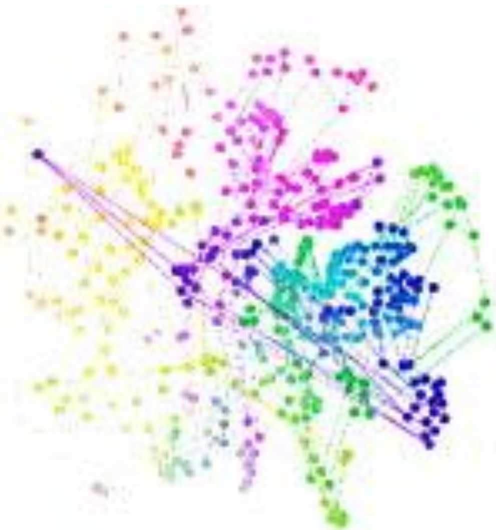
- A *self-organizing map* (SOM) is a grid of neurons which adapt to the topological shape of a dataset, allowing us to visualize large datasets and identify potential clusters.
- An SOM learns the shape of a dataset by repeatedly moving its neurons closer to the data points. Distinct groups of neurons may thus reflect underlying clusters in the data.

Illustration

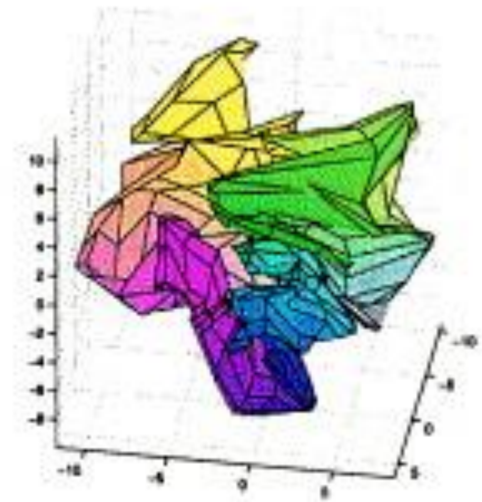


Dimensionality Reduction

- They provide a way of representing multidimensional data in much lower dimensional spaces - usually one or two dimensions



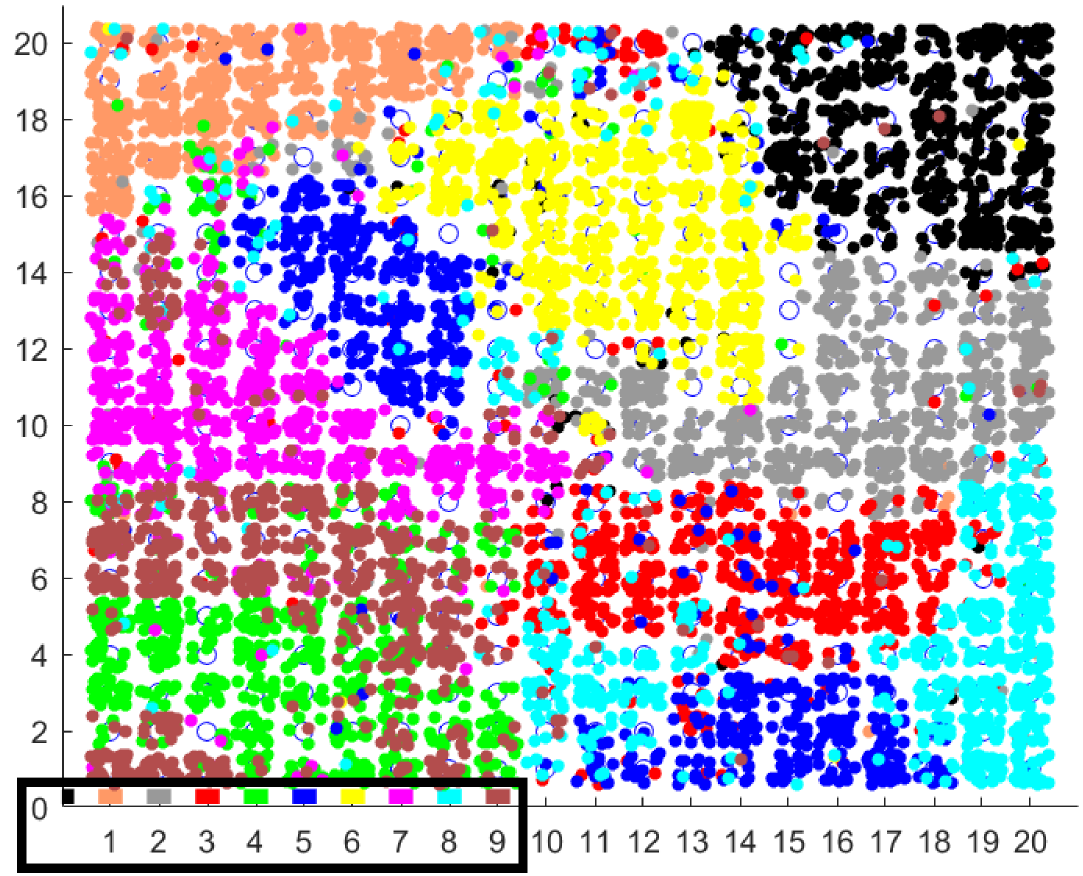
(a) 2D projection



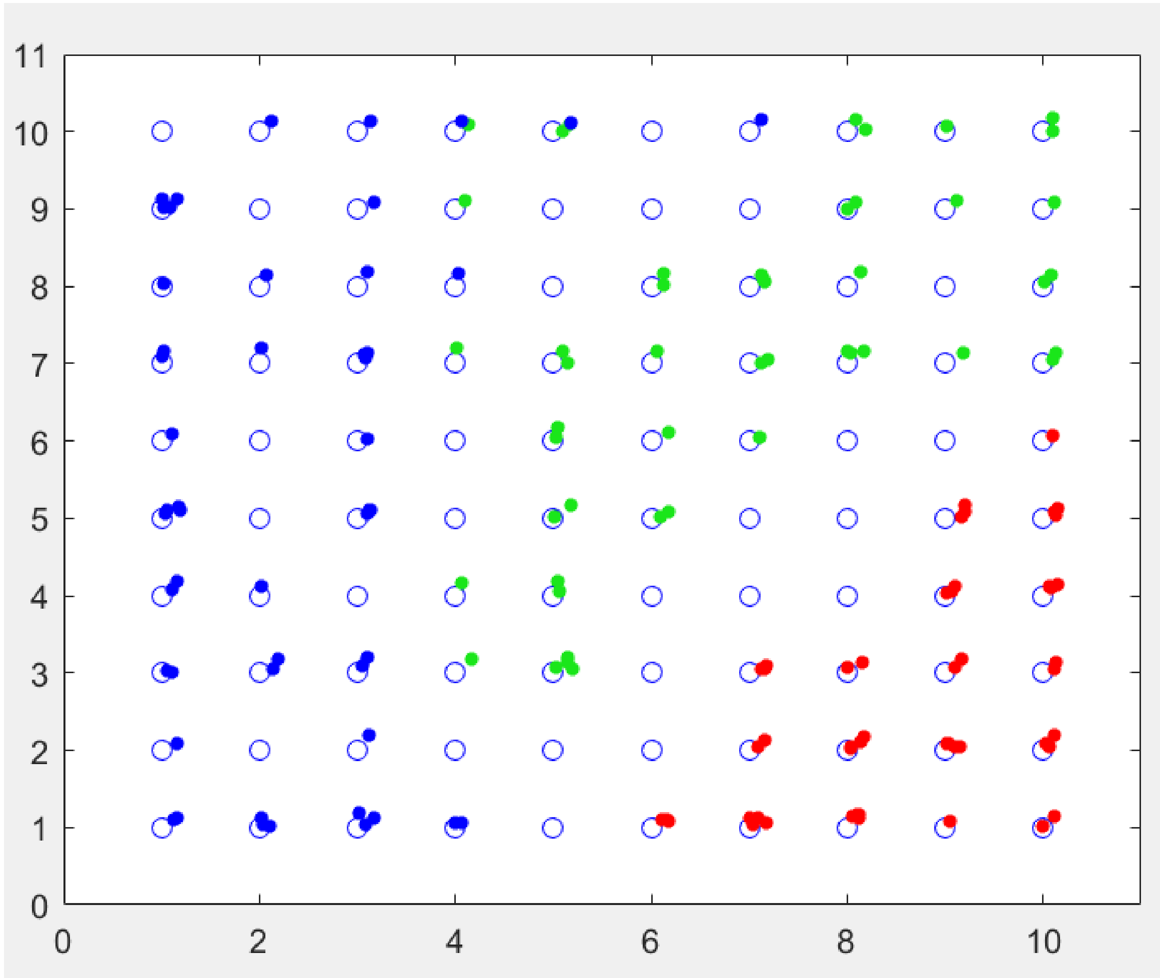
(b) 3D projection

(c) Color coding

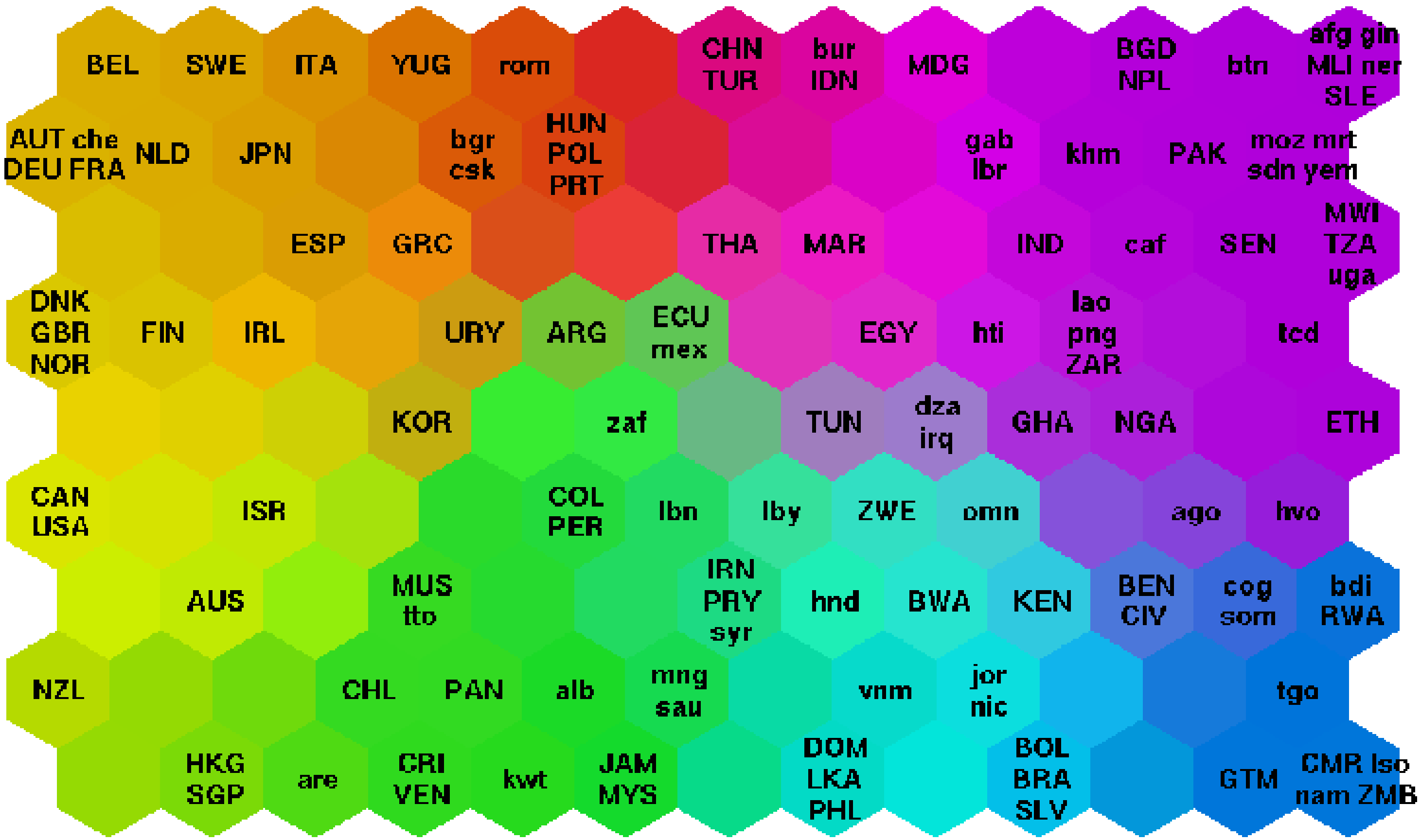
Handwritten Digits



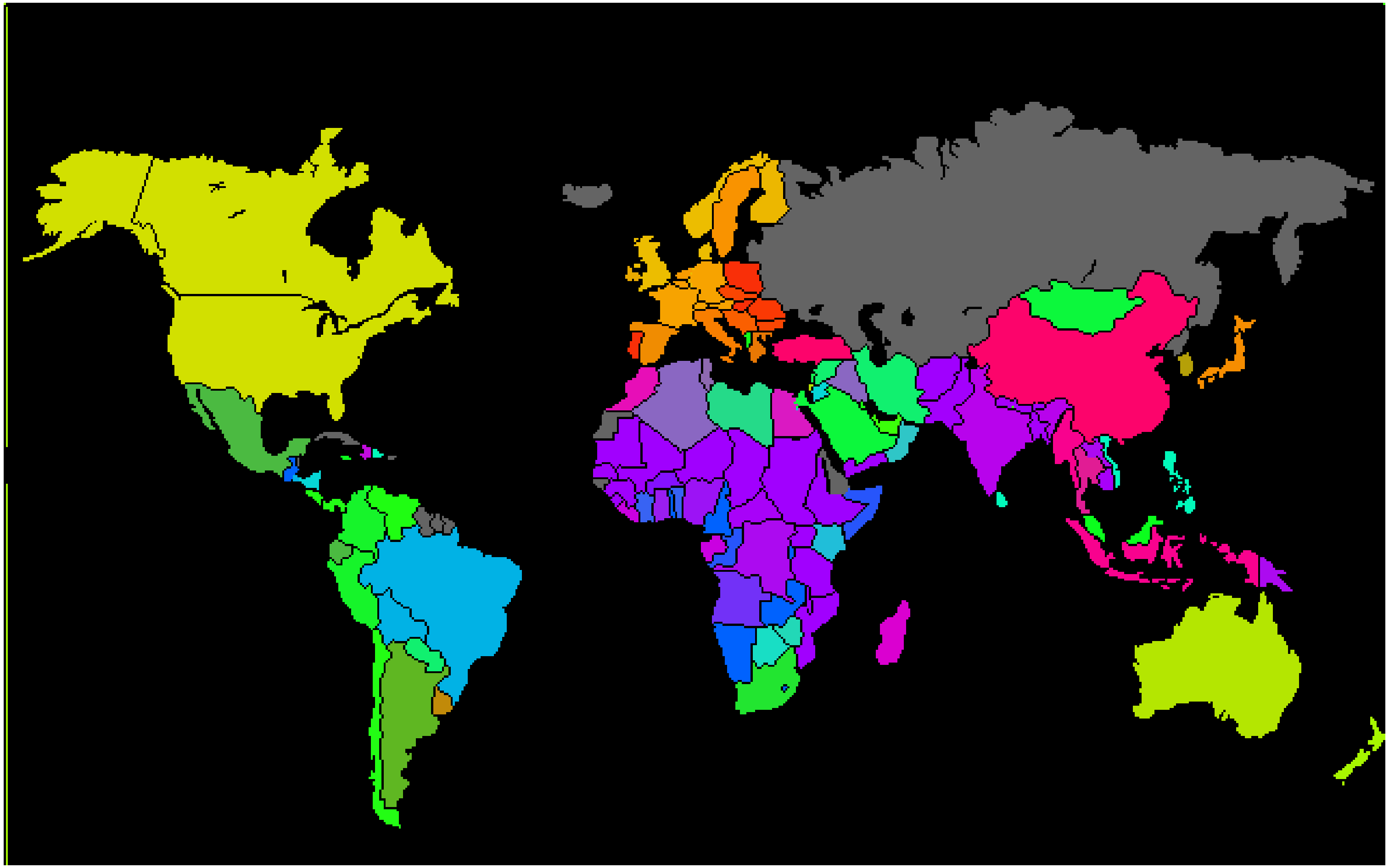
Iris Data



World Poverty Map



World Poverty Map



References

- [SOM tutorial part 1 (ai-junkie.com)]