VIT UNIVERSITY
VELLORE INSTITUTE OF TECHNOLOGY
VIT®
BHOPAL
www.vitbhopal.ac.in

Self-Organizing Map (SOM)

Dr. Virendra Singh Kushwah

Assistant Professor Grade-II

School of Computing Science and Engineering

Virendra.Kushwah@vitbhopal.ac.in

7415869616

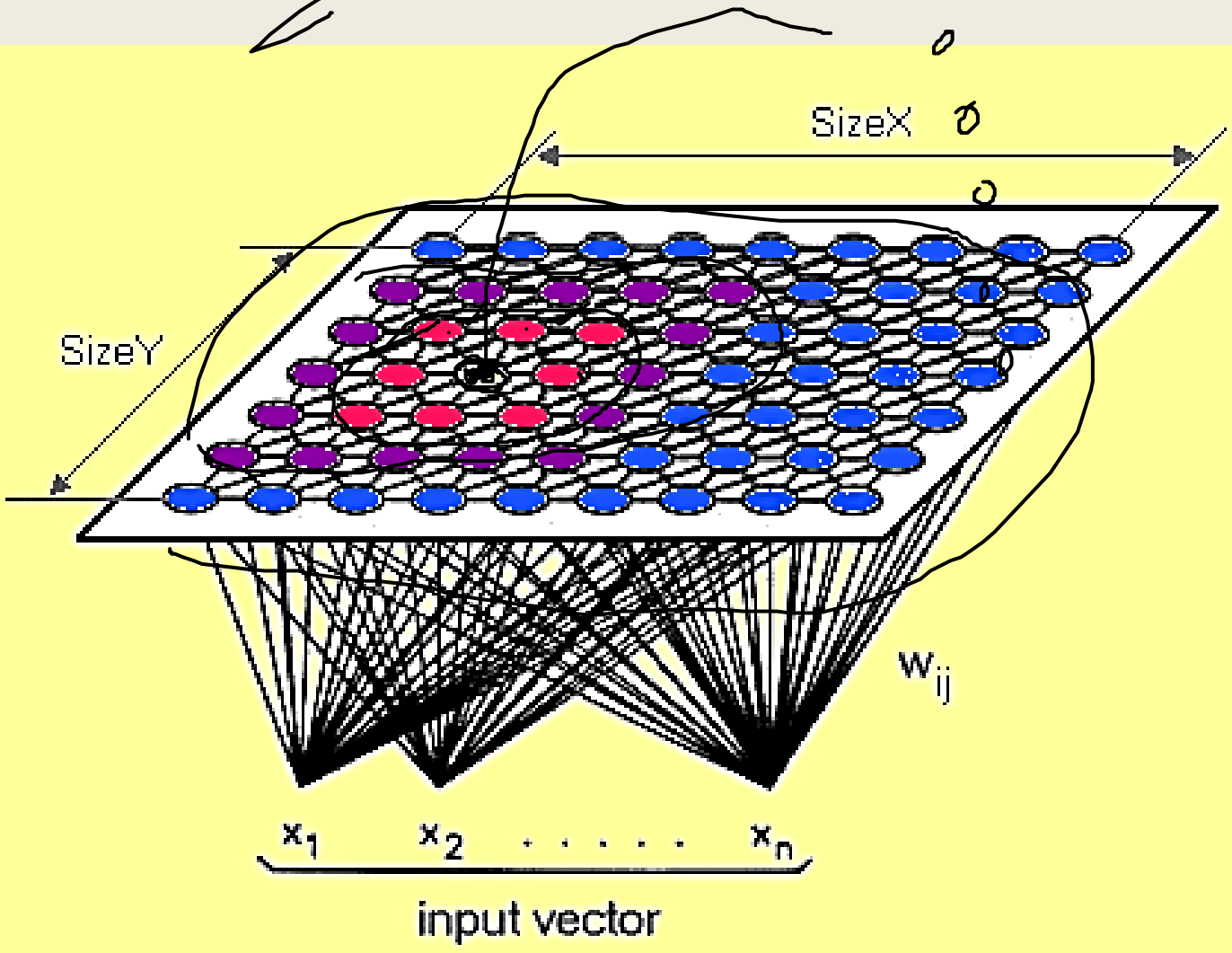
Self-organizing map

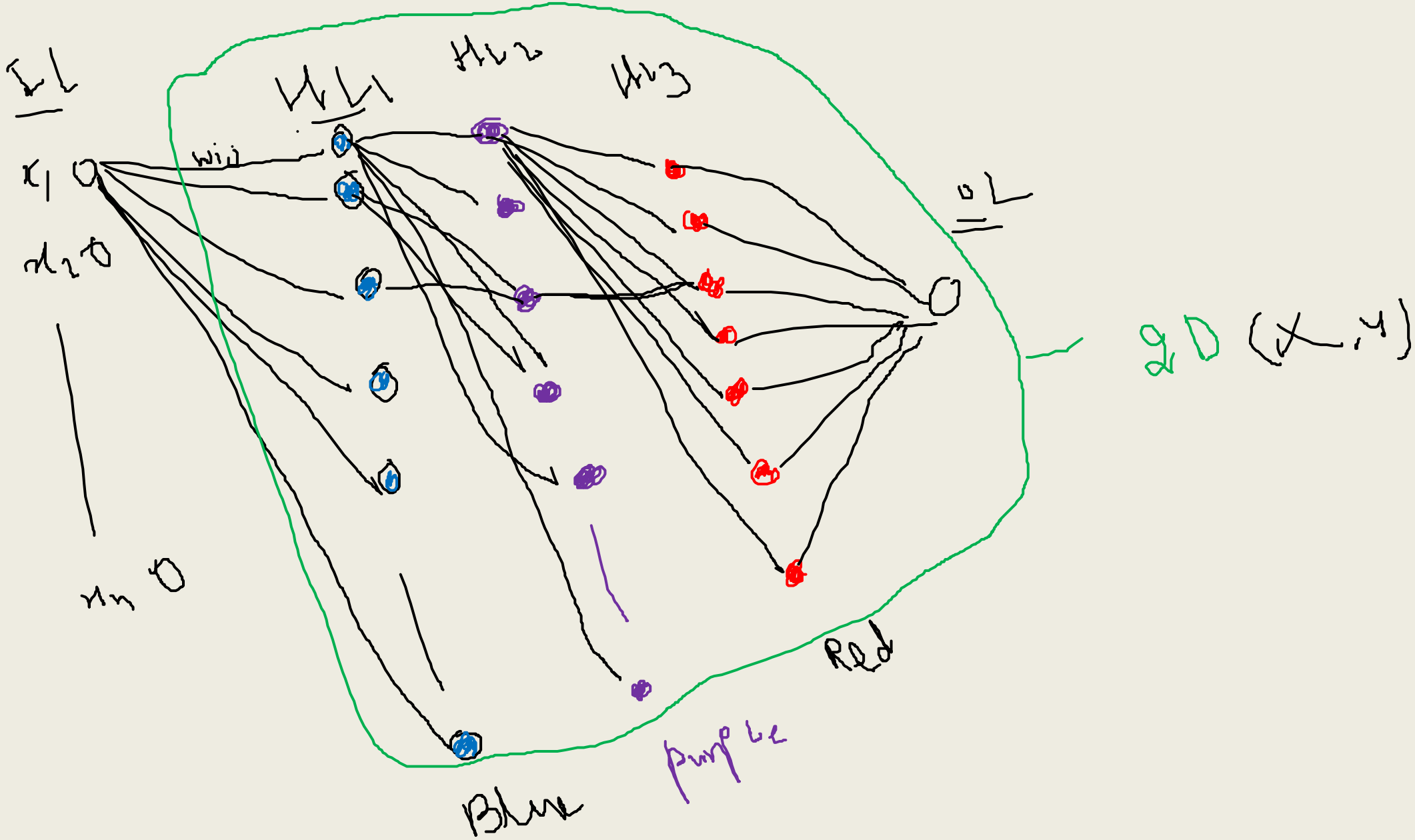
- A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction.

- *Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data.*

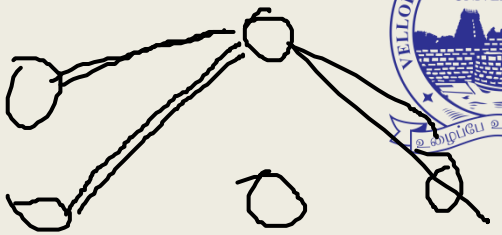
- Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space.

- This makes SOMs useful for visualization by creating low-dimensional views of high-dimensional data, akin to multidimensional scaling. The artificial neural network introduced by the Finnish professor Teuvo Kohonen in the 1980s is sometimes called a Kohonen map or network.


SizeX
SizeY
w_{ij}
x_1 x_2 . . . . . x_n
input vector


IL
HL1
HL2
HL3
OL
wij
x1
x2
xn
Blue
Red
Purple
g(x,y)

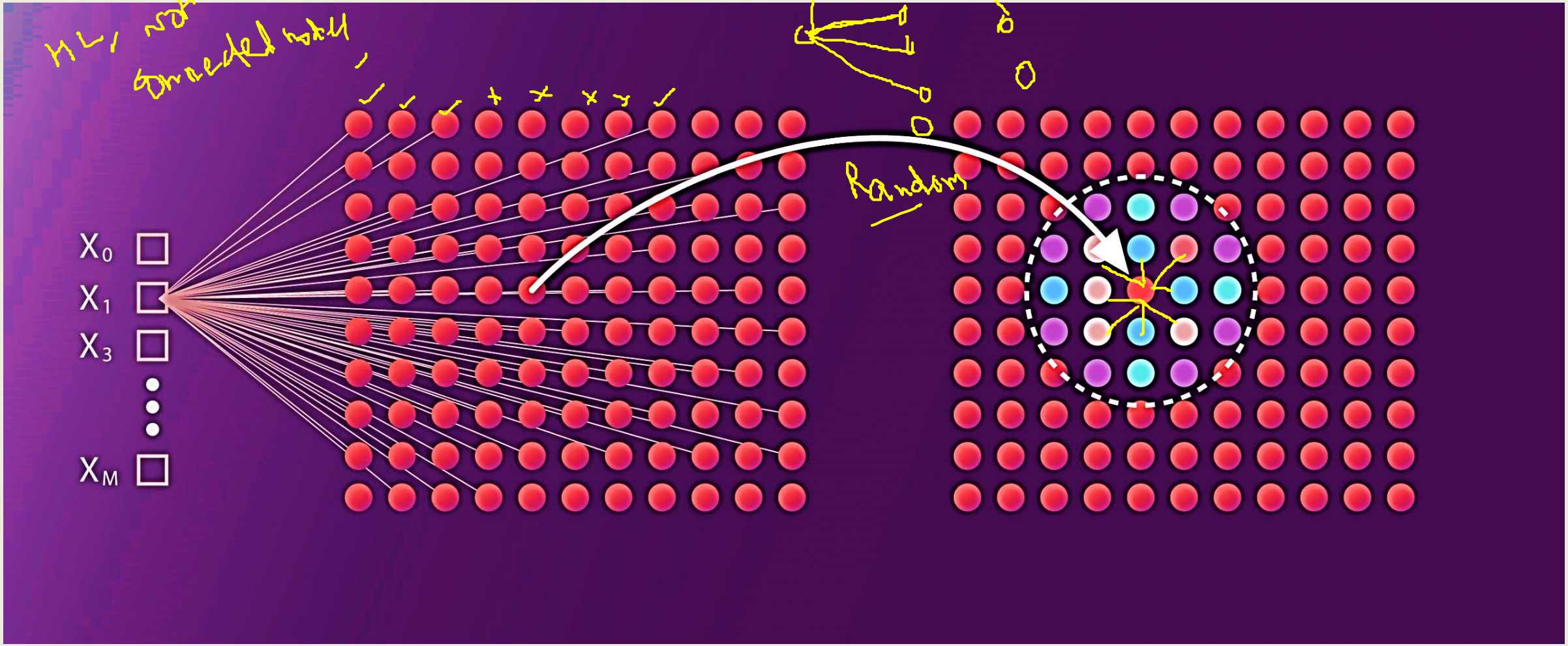
What really happens in SOM ?

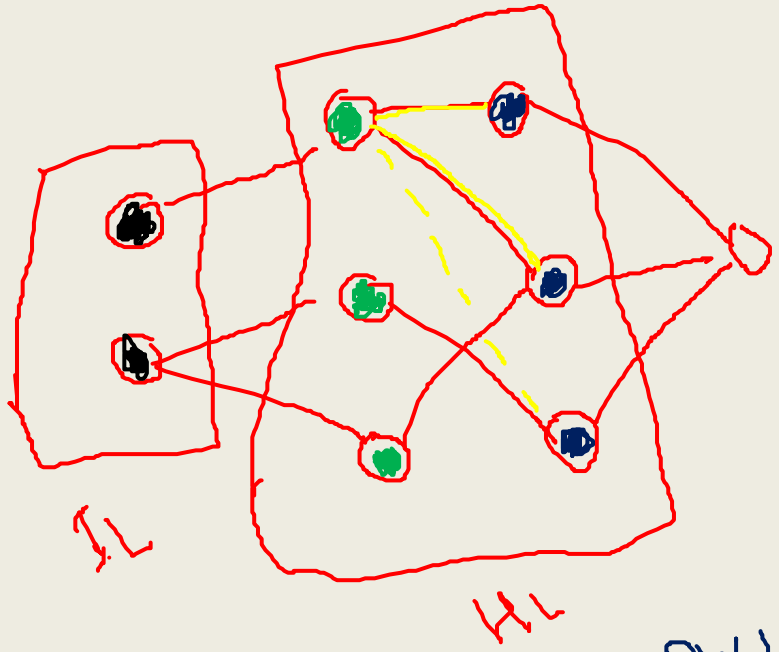


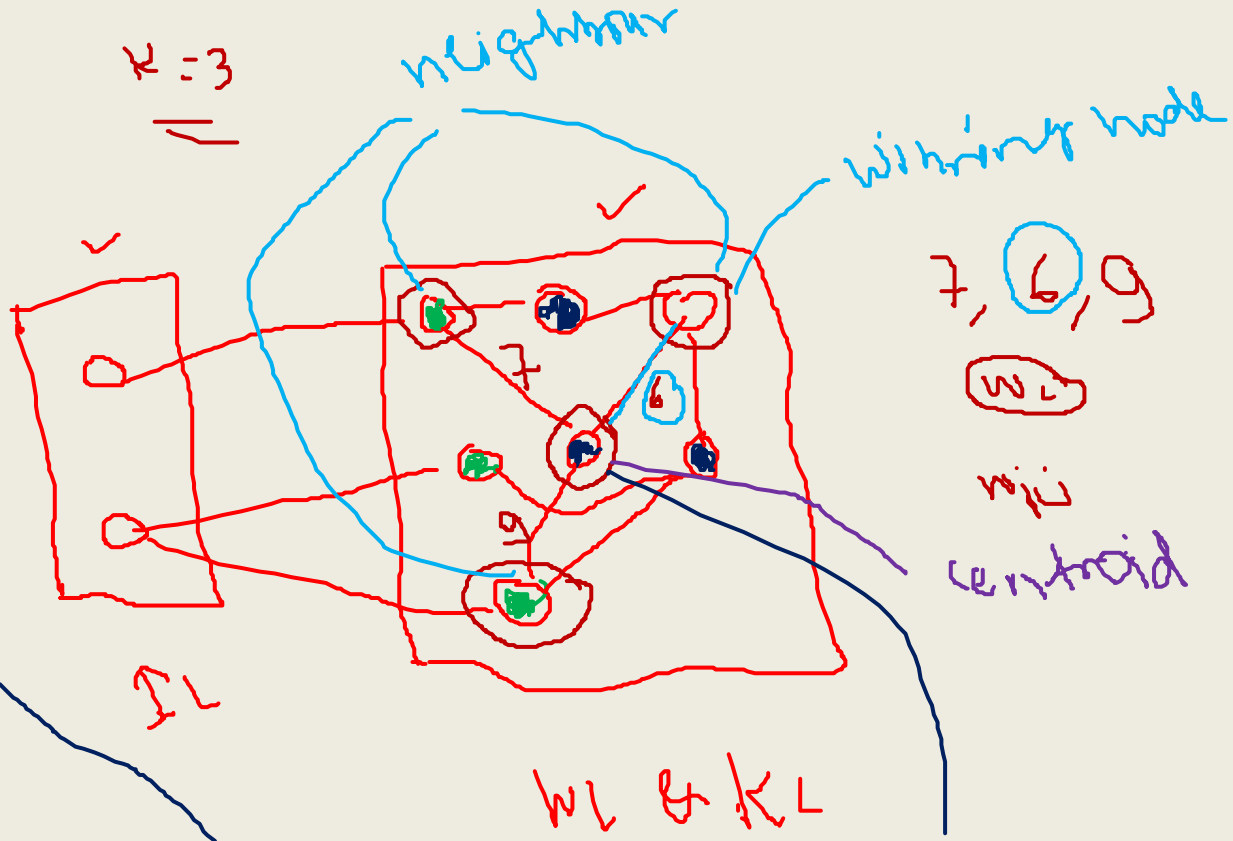
VIT BHOPAL
www.vitbhopal.ac.in

- Each data point in the data set recognizes themselves by competeting for representation.
- SOM mapping steps starts from initializing the weight vectors. From there a sample vector is selected randomly and the map of weight vectors is searched to find which weight best represents that sample.
- Each weight vector has neighboring weights that are close to it. The weight that is chosen is rewarded by being able to become more like that randomly selected sample vector.
- The neighbors of that weight are also rewarded by being able to become more like the chosen sample vector. This allows the map to grow and form different shapes. Most generally, they form square / rectangular / hexagonal / L shapes in 2D feature space.

Kohonen Self-Organizing Maps: One of the Fundamental UL Algorithms







IL = 2 feature = $[I_1, I_2]$ (size)

HL = 2 HLs = $[3 \times 3]$

OL = $[1 \times 1]$

2×1 3×3 1×1

Wi & KL

2×1 $n \times m$

weight matrix

$W_{ij} = [7, 6, 9]$

- A SOM is a set of processing units (neurons) placed in a 1 or 2-dimensional array, where all the neurons have weight vectors that correspond to points in an n-dimensional feature space.

- The instances of data are assigned to a processing unit whose weight vector it is closest to and training is when the initially randomized weight vectors are adjusted in a way to make neighboring neurons close to each other and, consequently, assign similar instances of data to the same neuron or nearby neurons.

- The result is a combination of clustering and dimensionality reduction – each cluster in the n-dimensional feature space is linked to a processing unit in the 1 or 2-dimensional array.

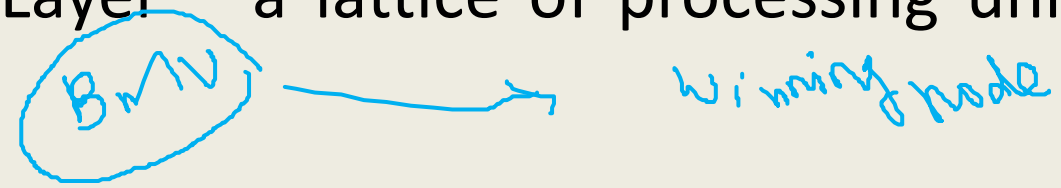
Detailed description

- A Kohonen SoM is a neural network consisting of three layers:

1. Input layer – a set of inputs in an n-dimensional feature space.

2. Weight layer – a set of adjustable weight vectors belonging to the network's processing units and defining their spatial location in a 1 or 2-dimensional space.

3. Kohonen Layer – a lattice of processing units to which the inputs are mapped.



- Unlike most neural nets, SOM utilizes competitive learning. There are always winning nodes that have the highest synaptic values and there are units around them that represent fewer instances of input data.

- These groups of neurons are called neighborhoods and they grow together, but the farther a unit is from the winning neuron, the less synaptic value it's got – these are topological neighborhoods that decay with distance.

- If compared to K-Means, these neighborhoods are clusters with the winning neuron being a sort of immovable centroid.

- During the competition phase of learning, the Euclidean distances between the weight vectors and the feature vectors are calculated and each instance of data is assigned to a neuron whose weight vector's values most resemble its own. This is how continuous input spaces are mapped to discrete spaces.

- The following formula is used to calculate the distances and establish the winner node (best matching unit):

$$Dist = \sqrt{\sum_{i=0}^{i=n} (V_i - W_i)^2}$$

V = the current input vector

W= the node's weight vector

The Algorithm

1. Each node's weights are initialized.
2. A vector is chosen at random from the set of training data.
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. Then the neighborhood of the BMU is calculated. The amount of neighbors decreases over time.
5. The winning weight is rewarded with becoming more like the sample vector. The neighbors also become more like the sample vector. The closer a node is to the BMU, the more its weights get altered and the farther away the neighbor is from the BMU, the less it learns.
6. Repeat step 2 for N iterations.

- Best Matching Unit is a technique which calculates the distance from each weight to the sample vector, by running through all weight vectors. The weight with the shortest distance is the winner. There are numerous ways to determine the distance, however, the most commonly used method is the Euclidean Distance

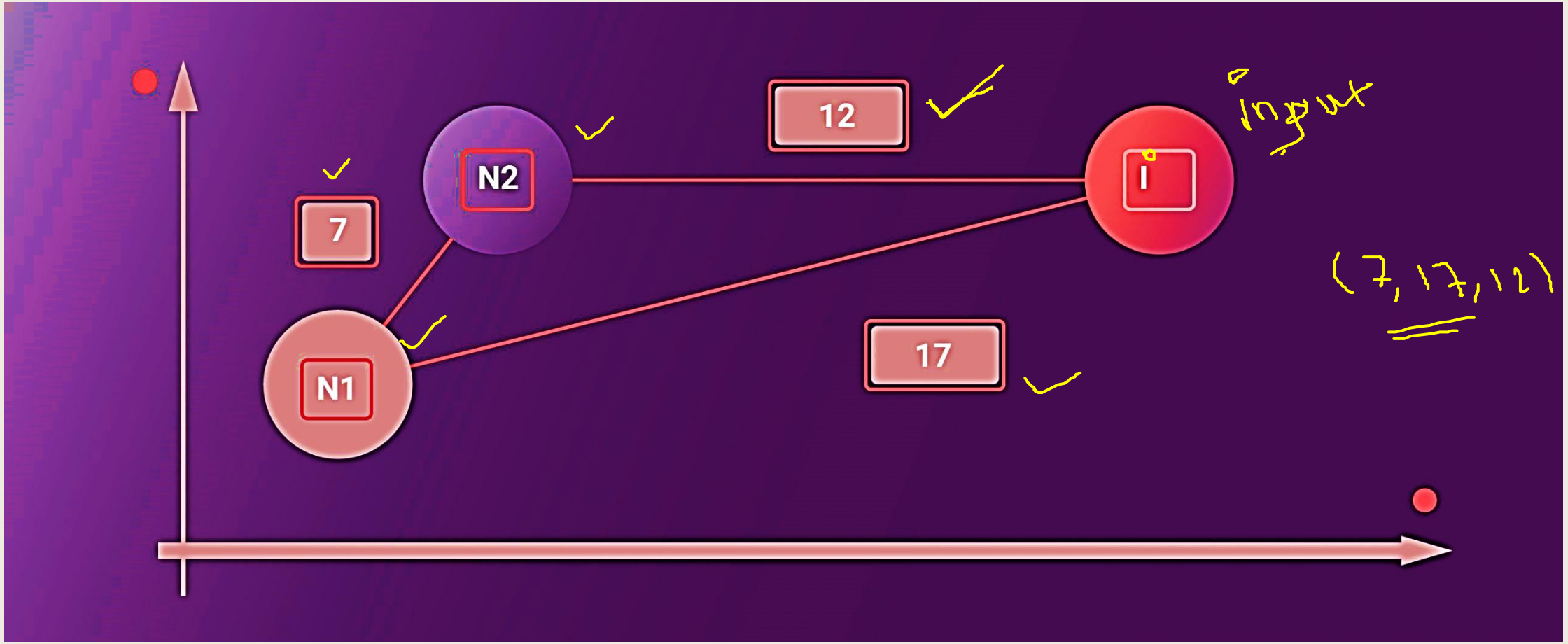
So, how does this result in dimensionality reduction?

- As we've mentioned, in SOMs neurons that represent similar/related pieces of input are kept together and the lateral distance is key to this because it is determined by the Euclidean distance between neurons' weights.

- Suppose we have one piece of input i and two neurons – n1 and n2. We've calculated that the distance (S) between n1 and n2 is 7, the S between n1 and i is 17, and between n2 and i, the S is 12.

- Originally these input feature vectors and neuron weigh vectors might have had hundreds of dimensions but it doesn't matter since their dimensionality is now reduced to singular scalar values (7,17,12) which we can put on a graph that's easy to comprehend.

- As you can see, this forces the placement of the neurons together; we've visualized them as points on the 2-dimensional plot with some distances between each other.

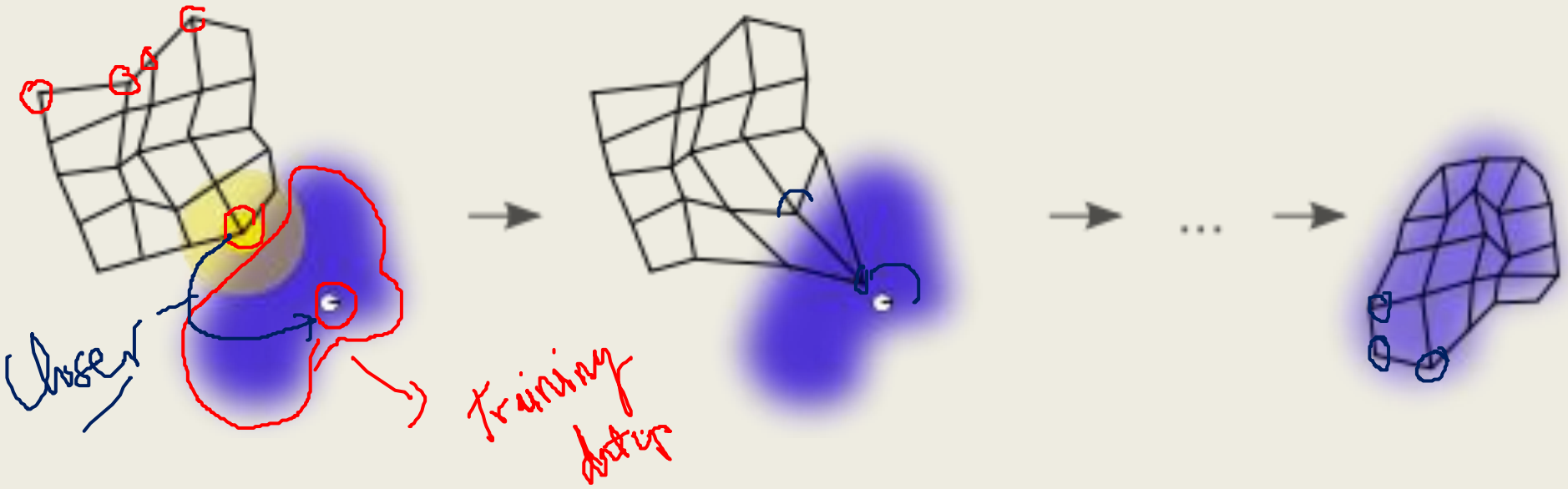


- Unsupervised learning is on the rise and Kohonen Self-organizing Map (and its extensions/modifications) has been one of the most widely used algorithms for clustering and dimensionality reduction. Its architecture allows transforming n-dimensional inputs to 1- or 2-dimensional outputs which are easy to comprehend and extract insights from.

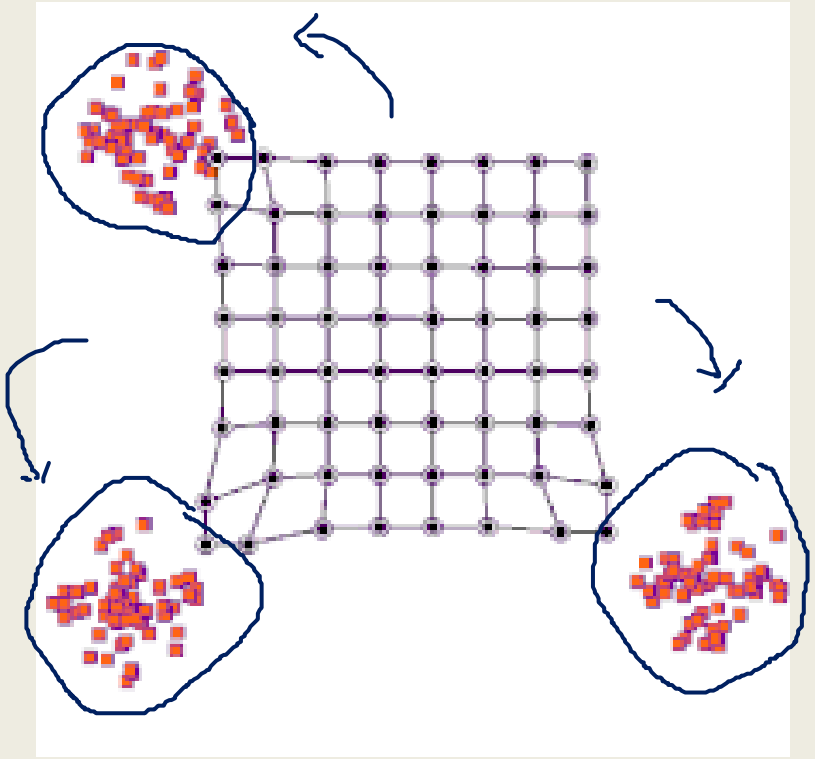
Learning algorithm

- The goal of learning in the self-organizing map is to cause different parts of the network to respond similarly to certain input patterns. This is partly motivated by how visual, auditory or other sensory information is handled in separate parts of the cerebral cortex in the human brain.

- The weights of the neurons are initialized either to small random values or sampled evenly from the subspace spanned by the two largest principal component eigenvectors. With the latter alternative, learning is much faster because the initial weights already give a good approximation of SOM weights.

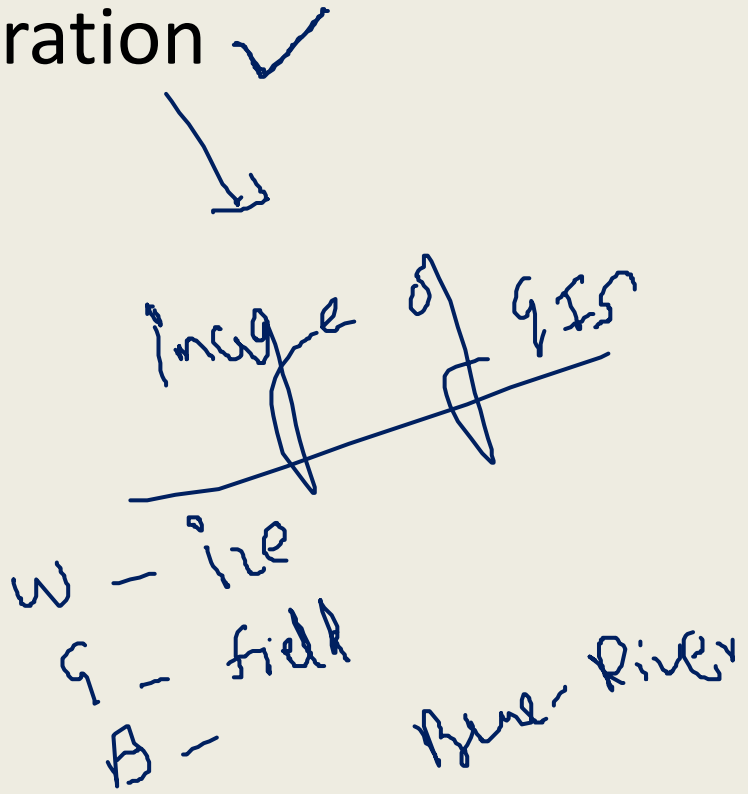


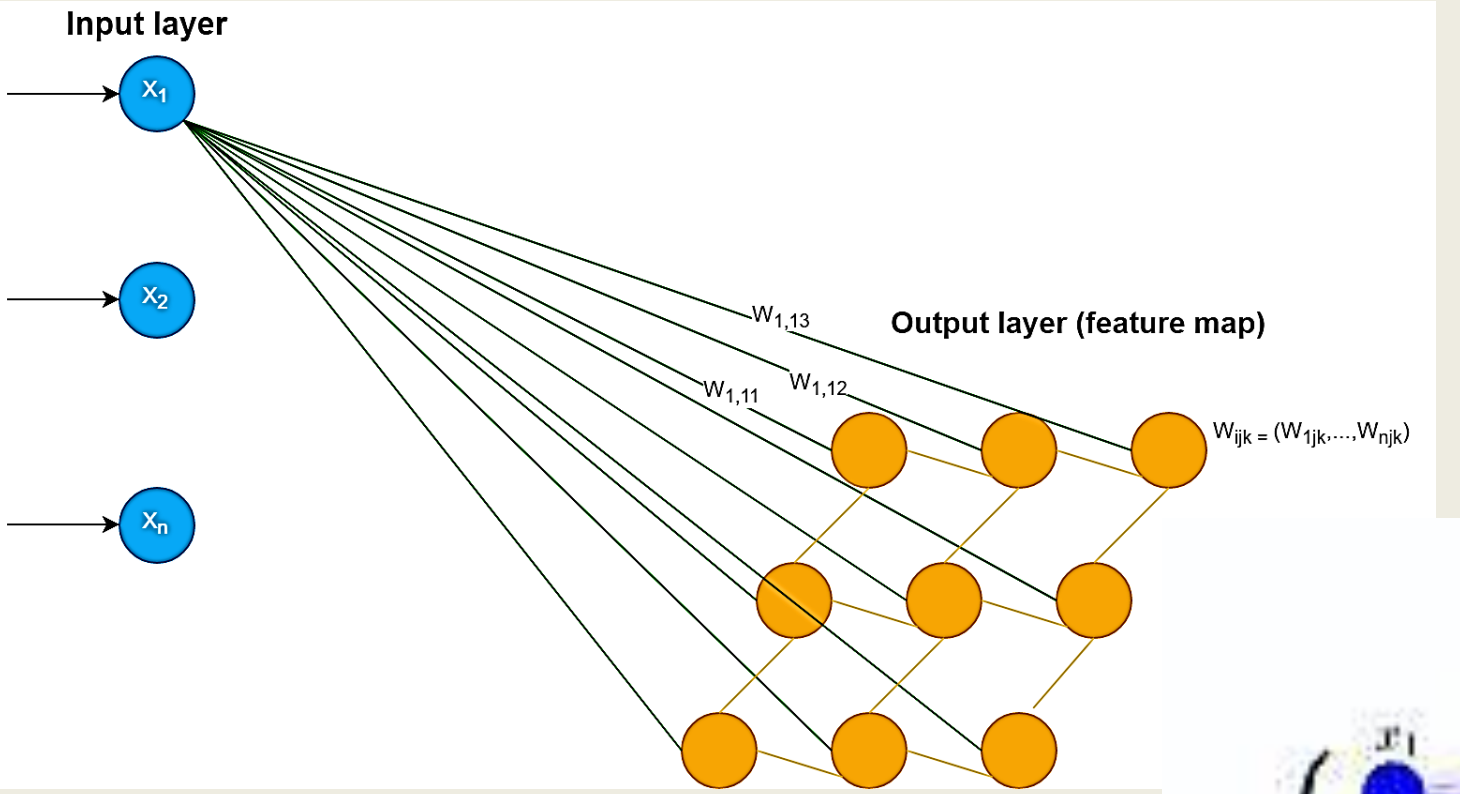
An illustration of the training of a self-organizing map. The blue blob is the distribution of the training data, and the small white disc is the current training datum drawn from that distribution. At first (left) the SOM nodes are arbitrarily positioned in the data space. The node (highlighted in yellow) which is nearest to the training datum is selected. It is moved towards the training datum, as (to a lesser extent) are its neighbors on the grid. After many iterations the grid tends to approximate the data distribution (right).

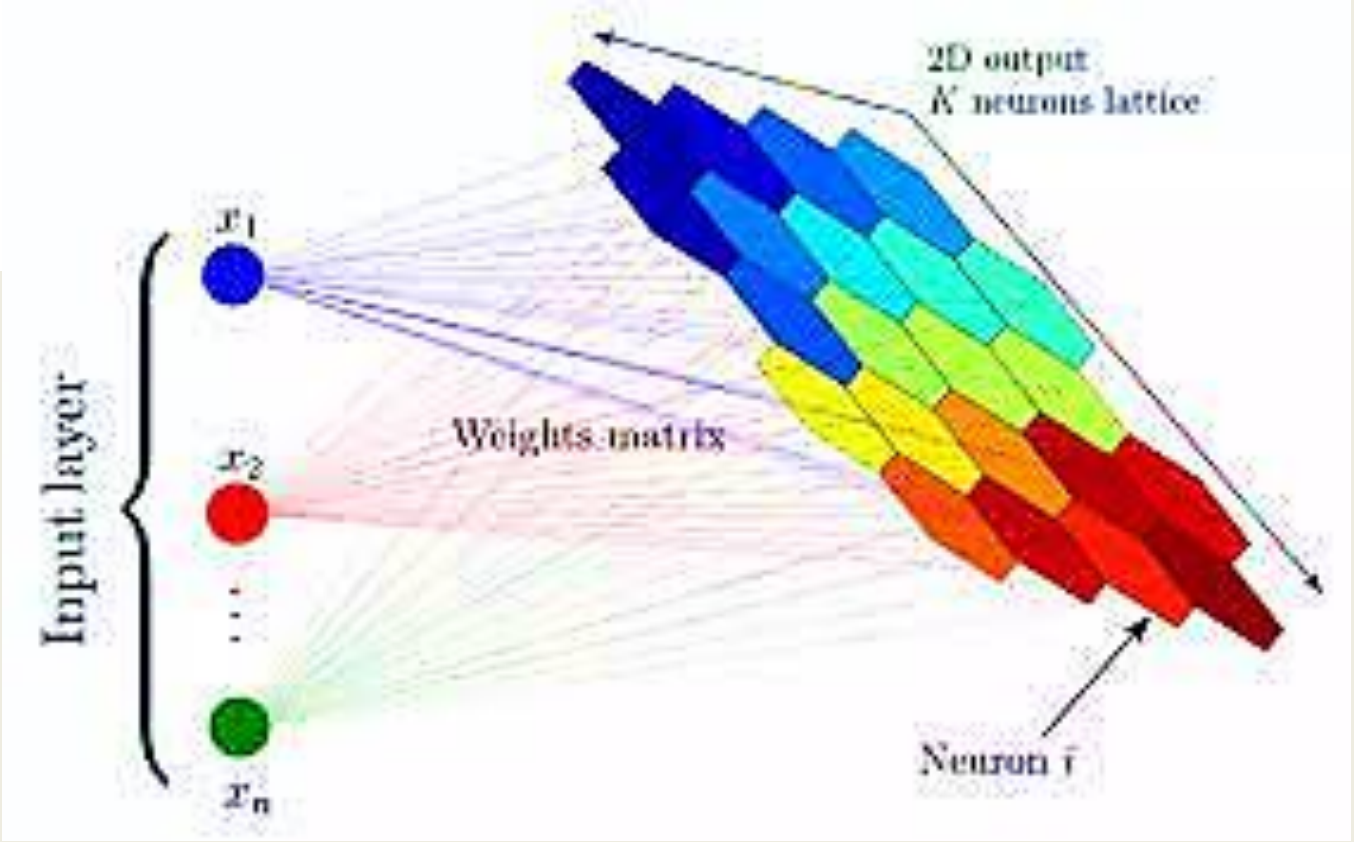


Applications

- Project prioritization and selection
- Seismic facies analysis for oil and gas exploration
- Failure mode and effects analysis
- Creation of artwork




Input layer
x_1
x_2
x_n
$W_{1,13}$
$W_{1,12}$
$W_{1,11}$
Output layer (feature map)
$W_{ijk} = (W_{1jk},...,W_{njk})$


2D output
K neurons lattice
Input layer
x_1
x_2
x_n
Weights matrix
Neuron i

VIT®

BHOPAL

www.vitbhopal.ac.in