**Project Summary**

**Self-Organizing Map**

Group Project: [Academic, 2018], Subject: **Data Mining**

**(missing data imputation and Algorithm Acuteness Processing)**

**Objective:** Find the missing data imputation of Datasets (Unsupervised Learning) and tune the parameter of algorithm to achieve NRMS 0.1 and AE < 1.

**Supportive Environment:**

**Programming language**: MATLAB 2018a (Math work)

**Software Toolkit**: Orange (data mining, machine learning)

**Explanation:**

* There are 24 real modified datasets given by the professor. And in the group, our goal is to
* identify the missing data pattern of 19 types of different dataset with [1%,5%,10% ,20%] using data mining and find the possible answer with the help of the Unsupervised algorithm.
* We choose SOM (self organize map) to find the answer with clustering, gradient descent and identify the no of iteration for best accuracy of the model by comparing with true data set.
* Use gaussian function and train the model using parameters of neighborhood function with measurement of BMU (neuron u) and neuron v.

.

**Parameters:**

**1 Initial Width (w)**

**2 learning rate (),** for better time profile and fast optimization

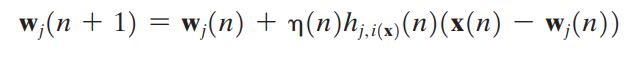
**Used Notations:**

These are the variables needed, with vectors in **‘bold’**

* {\displaystyle s} ***S***  is the current iteration
* {\displaystyle \lambda }***ƛ*** is the iteration limit
* ***t*** is the index of the target input data vector in the input data set {\displaystyle \mathbf {D} }
* {\displaystyle {D}(t)}***D(t)***  is a target input data vector
* {\displaystyle v}***v*** is the index of the node in the map
* {\displaystyle \mathbf {W} \_{v}} ***Wu*** is the current weight vector of node {\displaystyle v}
* {\displaystyle u} ***u*** is the index of the best matching unit (BMU) in the map
* {\displaystyle \theta (u,v,s)} is a restraint due to distance from BMU, usually called the neighbourhood function, and
* {\displaystyle \alpha (s)} is a learning restraint due to iteration progress.

{\displaystyle t}

**Algorithm:**

1. Randomize the node weight vectors in a map
2. Randomly pick an input vector {\displaystyle {D}(t)}***D(t)***
3. Traverse each node in the map
   1. Use the Euclidean Distance formula to find the similarity between the input vector and the map's node's weight vector
   2. Track the node that produces the smallest distance (this node is the best matching unit, BMU)
4. Update the weight vectors of the nodes in the neighborhood of the BMU (including the BMU itself) by pulling them closer to the input vector
   1. {\displaystyle W\_{v}(s+1)=W\_{v}(s)+\theta (u,v,s)\cdot \alpha (s)\cdot (D(t)-W\_{v}(s))}
5. Increase {\displaystyle s} ***s*** and repeat from step 2 while {\displaystyle s<\lambda }***s< ƛ***

**Work challenges:**

**1]** **Space complexity:** For dealing with the big datasets as well as iterative process produce shortage of the local environment space.

**2]** **Data cleaning**, Pattern recognition is complex task

**3]** **Computational complexity:** T = O(N\*C) = O(S2), with increase of the vector and matrix size the complexity increase due to depend on no. of rows and columns. In term of row and columns its 0(m\*n)

**Achievements:**

* Use NRMS and AE method for identify the error and accuracy percentage. Our goal is to keep NRMS near to 1 and AE must be more than 1. We achieve accuracy between 75 to 90 percent. However, we can not manage some dataset due to inefficient time and space availability problem.
* But, successfully Identify the measurement weakness of SOM, that with increase of missing rate the accuracy decreases significantly.