

## **D7041E “Applied artificial intelligence”**

---

**IT IS STRICTLY FORBIDDEN TO USE AI GENERATED CODE AND  
COPY EXISTING CODE FROM THE INTERNET. ALL CASES OF  
VIOLATION WILL BE REPORTED!**

### **LAB 4: Unsupervised learning with Self-Organized Maps**

#### **1. Introduction**

In this lab we will work with Self-Organized Maps. A Self-Organizing Map, or SOM, falls under the rare domain of unsupervised learning in Neural Networks. It is essentially a grid of neurons, which is used for projecting multidimensional data onto a meaningful coordinates revealing the class structure.

#### **2. Fundamentals of data pre-processing and basic skills in Jupyter**

Following the example of SOM implementation in the Jupyter notebook from the class, extend it to learn the SOM topology of the MNIST dataset. Study the effect of SOM hyperparameters.

#### **Task 1.1 Unsupervised learning of hand-written digits with SOM:**

1. Load the MNIST dataset
2. Use the flattened (that is 1D) array of pixels of each image as a feature vector.
3. Initializing the weights in the SOM network randomly, train SOM with grid 20x20, 40x40, 80x80.
4. Display the initial, intermediate (at 50% of iterations) and the final learned weights of SOM neurons as a grid of 28x28 images.

5. Now by passing the TRAINING examples through the trained SOM and recording the statistics of matching in each node assign labels to the neurons.
  - Display the confusion matrix of classification of the TRAINING SET
  - Display the confusion matrix of classification of the TEST SET
6. For a fixed number of iterations increase and decrease the learning rate of SOM:
  - What is the resulting effect?
7. For a fixed number of iterations and the best learning rate increase and decrease the exponential decay of the neighbourhood parameter.
8. What is the effect?
9. What is a biological neuron? How does it relate to the concept of neurons in SOM?

**Congrats, you have just become familiar with a very powerful tool for unsupervised learning - Self-Organized Networks. Well done!**