Indian Institute of Technology, Indore

Computer Science & Engineering

CS 354N: Assignment VII - SOM

Date- 18-03-2025

Some general instructions:

● Name your file in "Assignment7_yourRollno.zip" format.
● You are not allowed to use any built-in libraries related to the topic. Code from scratch is preferably advised.
● Submission of the assignment should be made using the Google Classroom platform only.
● Plagiarism in any form will not be tolerated.
● You are allowed to submit only once before the deadline. Avoid multiple submissions. In such a case, only the last submitted file will be considered for evaluation.
● Last date for submission of the assignment: 25-03-2025
● Submit a single file (report in zip format) containing the procedure (screenshot of main procedures/code/Results).

Lab Assignment

1. Implement a Basic 1D Kohonen SOM in C/C++ to perform the following:

a) Initialize a 1D weight vector with small random values.
b) Select a random input vector from the dataset.
c) Compute the Best Matching Unit (BMU).
d) Update the weights using the learning rate and neighborhood function.
e) Repeat for multiple iterations and visualize the weight changes over time.

2. Implement in C/C++ Kohonen Self organizing Feature Map for n input and m output network with learning rate 0.5 initially and perform the following:

a) Initialize a **2D grid of weight vectors**.
b) Train the SOM using **a small dataset** (e.g., Iris dataset, handwritten digits, or customer segmentation).
c) After training, visualize how input vectors map to the 2D SOM.
d) Use **heatmaps** to show clustering effects.
e) Experiment with different learning rates and neighborhood functions.

3. Use **Kohonen SOM** to detect anomalies in network traffic with a coding language of your choice.

a) Load iris dataset used in previous lab assignment.
b) Normalize the data and train a **2D SOM**.
c) After training, measure **distance from BMU** to identify anomalies.
d) Highlight anomalies using **outlier detection** techniques.
e) Try to extend the work for **KDD Cup 99** dataset for anomaly detection

Hint:

- Normalizing the Data: Normalizing is straightforward, ensuring that all features have the same scale. Python, R, or MATLAB has libraries and functions for this.
- Training a 2D SOM: Kohonen Self-Organizing Maps can be implemented using available libraries in Python such as: MiniSom, somoclu, or sompy
- MATLAB: SOM Toolbox Training involves creating a 2D grid (e.g., 10x10) of neurons and adjusting weights to map the dataset.
- Measuring Distance from BMU: After training, the Best Matching Unit (BMU) represents each input. The distance from the BMU for each data point can indicate normal data points (closer distance) versus potential anomalies (farther away). This is a key step in anomaly detection.
- Outlier Detection Techniques: You can apply statistical outlier detection methods to identify and highlight anomalies, such as: Z-score, IQR (Interquartile Range), Isolation Forest for extending the project.