CS 451 – Computational Intelligence
Spring' 2025
Assignment # 3 – More Optimization with Reinforcement Learning and Self Organizing Maps, and PSO

Objective: This assignment provides students with hands-on experience in self-organizing maps, reinforcement learning, and particle swarm optimization (PSO). Students will use these techniques to experiment with data visualization, agent navigation in a grid-world, and non-convex function optimization.

Question 1- [25 points] Clustering COVID data using SOM

a) In this question, you will apply Self-Organizing Maps (SOM) on a world dataset of your choice (such as world poverty data, climate data, economy data etc.) to build an effective visualization of that dataset. You have to make sure that there are multiple attributes in that dataset that make the cluster analysis meaningful.

b) For visualization, you will come up with a mechanism to map the clusters to RGB values so that they appear on the grid in different colors and gradual convergence of SOM could be seen over iterations. Each color represents a cluster and its varying shades denotes nearby/similar clusters.

Figure 1- Convergence of SOM

You will use the '**sum of squared distance'** between input and weight vectors to find the best matching unit (BMU). Both radius and learning rate will decay over time using the **exponential decay**. Here are some starting parameters to begin with:

- Size of SOM grid: 10* 10
- Initial learning rate: 0.4
- Initial radius: half of the size of grid

c) You will further enhance your visualization by showing each country on the world map using the color taken from the respective cluster on the SOM grid. This will result in a visualization similar to the following:

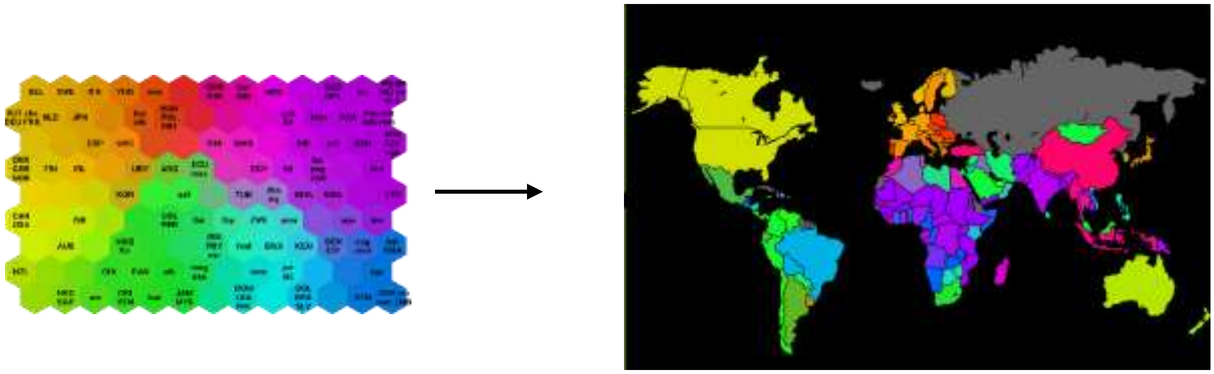


Figure 2- SOM clustering of world poverty data and its visualization on world map

Note: The sample is taken from World Poverty Map showing clustering based on poverty related dataset.) The GeoPandas library can be used for plotting. Its dataset of '*naturalearth_lowres*' gives contours of countries.

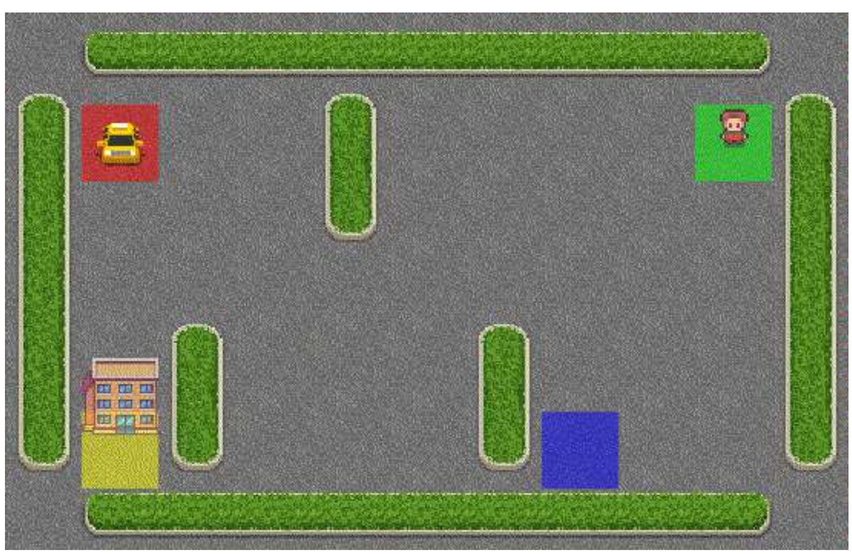
Grading:

The grading will be based on the following components:

Component	**Weight**
Selection of dataset	10%
Overall process of SOM	30%
Decay of radius and learning rate	10%
Initialization and convergence	10%
Visualization in SOM grid	20%
Visualization on World Map	20%

Question 2 [13 points]- Learning to navigate in Taxi-v3

You are building your RL based agent for the Taxi-v3 environment in the OpenAI gym. In this environment, an agent is tasked with controlling a taxi that must pick up and drop off passengers at specific locations on a grid.



The agent must learn to efficiently navigate the grid, obeying rules like picking up passengers at one location and dropping them off at another while avoiding obstacles. The state space consists of the taxi's position, the passenger's location, and the destination, while the action space involves movements like moving the taxi in one of the four cardinal directions, picking up passengers, or dropping them off. Taxi-v3 is widely used for teaching and experimenting with reinforcement learning techniques due to its simplicity and discrete state and action spaces.

A skeleton code is attached with the assignment that already creates a Taxi-v3 environment in gym and makes the agent perform selected actions. You need to write your code to learn the policy for action selection.

Grading:

The grading will be based on the following components:

Component	Weight
Overall process of value Iteration	35%
Value Update	15%
Stopping criteria	10%
Action selection from learned policy	20%
Execution of learned policy	20%

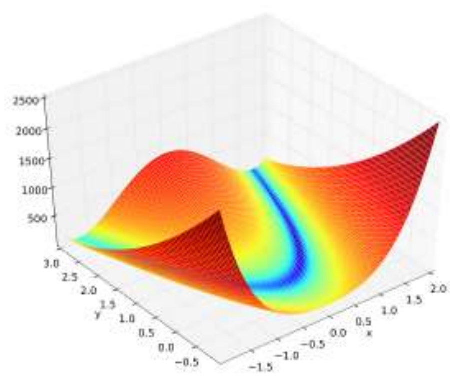
Q3 – [12 points] Particle Swarm Optimization for Continuous Optimization

In this question, you are required to use PSO to find the global minima of the following two functions:

a) Rosenbrock Function

$$f(x, y) = 100(x^2 - y)^2 + (1 - x)^2$$

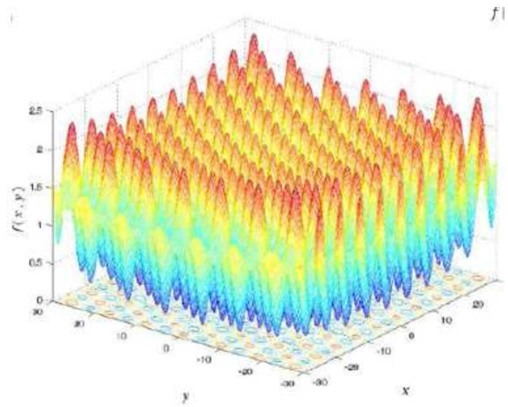
$$-2 < x < 2, -1 < y < 3$$



b) 2d- Greiwank's Function

$$f(x,y) = 1 + \frac{x^2}{4000} + \frac{y^2}{4000} - \cos(x)\cos\left(\frac{y}{\sqrt{2}}\right)$$

$$-30 < x < 30, \qquad -30 < y < 30$$



In both cases, you will plot the avg-so-far and best-so-far for each iteration and will report the final result. The grading will be based on the following components:

Component	Weight
Particle representation	10%
PSO implementation (velocity update, stopping criterion)	50%
Final Result	20%
Graph Plotting	20%

Submission: You will submit your properly commented code in the form of a separate python notebook for each question. No pdf report is required for this assignment.