# Machine Learning Laboratory(EE-527)

Assignment-3

February 3, 2020

1. (a) This is an Optimisation problem where you need to find the global minima of the given objective function using **Particle Swarm Optimisation (PSO)** technique. PSO solves an optimisation problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions. The goal is to find a solution **a** for which $f(a) \leq f(b)$ for all **b** in the search-space, which would mean **a** is the global minimum.

   (b) Take the objective function as:

$$f(x,y) = -20 * e^{-0.2*\sqrt{0.5(x^2+y^2)}} - e^{0.5*((\cos 2\pi x)+(\cos 2\pi y))} + e + 20$$

   (c) Take $b_{lo} = -10, b_{up} = 10, \omega = 0.05, \phi_p = 0.5, \phi_g = 1$

   (d) Plot the contour of the given function and show the scatter plot of the particles at each iteration.

Here, let S be the number of particles in the swarm, each having a position $x_i \in \mathbb{R}^2$ in the search-space and a velocity $v_i \in \mathbb{R}^2$. Let $p_i$ be the best known position of particle i and let g be the best known position of the entire swarm. The values $b_{lo}$ and $b_{up}$ represents the lower and upper boundaries of the search-space. The program terminates when **maxItr** number of iterations are performed.

The algorithm for Particle Swarm Optimization is as follows.

**Algorithm 1:** Particle Swarm Optimization

**for** *each particle i = 1, ..., S* **do**
  Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$ ;
  Initialize the particle's best known position to its initial position: $p_i \leftarrow x_i$ ;
  **if** $f(p_i) < f(g)$ **then**
    update the swarm's best known position: $g \leftarrow p_i$
  Initialize the particle's velocity: $v_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$
itr = 0;
**while** $itr < maxItr$ **do**
  **for** *each particle i = 1, ..., S* **do**
    **for** *each dimension d = 1, ..., n* **do**
      Pick random numbers: $r_p, r_g \sim U(0, 1)$
      Update the particle's velocity:
      $v_{i,d} \leftarrow \omega * v_{i,d} + \phi_p * r_p * (p_{i,d} - x_{i,d}) + \phi_g * r_g * (g_d - x_{i,d})$
  Update the particle's position: $x_i \leftarrow x_i + v_i$
  **if** $f(x_i) < f(p_i)$ **then**
    Update the particle's best known position: $p_i \leftarrow x_i$
    **if** $f(p_i) < f(g)$ **then**
      Update the swarm's best known position: $g \leftarrow p_i$
  itr = itr + 1

2. Consider the particles with same initial position and velocity as in Question 1, but alongwith the movement of global best in random directions, also consider the update of global best using Gradient Descent algorithm. Update the global value with the one which is smaller between these two.