CS412 – Machine Learning
Neural Networks
100pts

1) **50pts** - In this problem, we will apply the **steepest descent method** to minimize the function F(w) with respect to **w**=[x, y]. The algorithm needs to run for two steps. I.e. find **w_1, w_2** (corresponding to times t=1 and t=2) around the point **w_0**=[0, 0]. **You should use a "step size"   alpha = 0.1.** *a, b and c are constants.*

$$F(\mathbf{w}) = x^2 + y^2 - \cos(x-y)$$

Answer:

$F(w_0) = 0^2 + 0^2 - \cos(0-0) = -1$ *: Just to note at what F value we start (5pts)*

$\nabla F = [\delta F(w)/\delta x, \delta F(w)/\delta y] = [2x+\sin(x-y), 2y-\sin(x-y)]$ *: Compute the gradient (10pts)*

$\nabla F \mid w_0 = [2 * 0 + \sin(0-0), 2 * 0 + \sin(0-0)] = [0,0]$ *: This is the gradient **evaluated** at w_0 (10pts)*

$w_1 = w_0 - 0.1 \, \nabla F \mid w_0$ *:Update w0 to find w1 (5pts)*

$F(w_1) = 0^2 + 0^2 - \cos(0-0) = -1$

We are not minimizing nor maximizing *: just checking to see if we are indeed minimizing*

2) **50pts –** In this assignment we will carry out image classification tasks in Tensorflow and Keras. The images belong to plants. The PlantVillage dataset consists of 54303 healthy and unhealthy leaf images divided into 38 categories by species and disease. Develop and train Neural Network models that can make predictions. 10pts for preparing data.

    a. *20 pts.* The first task is to use two Dense layers with ReLu nonlinearity or so to say, two layer fully connected NN. The output layer is softmax with 38 categories. You need to use categorical cross entropy with learning rate 0.001.

    b. *20 pts.* The second task is to use two layer Convolutional NN with ReLu nonlinearity. The output layer is softmax with 38 categories. You need to use categorical cross entropy with learning rate 0.001.

Remark: All the implementations should be done on the notebook. You shouldn't use this document for the answers.

Reference:

1. Visit tensorflow dataset info page https://www.tensorflow.org/datasets/catalog/plant_village
2. Visit documentation for layer https://keras.io/layers/