EE4305 Assignment 1

Adil Bin Azmoon

A0155858M

Q1a.

f(x,y) = (1-x)2+100(y-x2)2

Therefore:

fx = 2(x-1)-400(y-x2)

fy = 200y-200x2

Using Steepest Descent Method,

w(k+1) = w(k) + ηg(k), where η = 0.001 and g(k) = ∇f(w(k))

Using the following Python script,

**n = 0.001**

**def derivX(x,y):**

**answer = (2\*(x-1)) - (400\*x\*(y-(x\*\*2)))**

**return(answer)**

**def derivY(x,y):**

**answer = 200\*(y-(x\*\*2))**

**return(answer)**

**def deltaX(x,y):**

**answer = x - (n\*derivX(x,y))**

**return answer**

**def deltaY(x,y):**

**answer = y - (n\*derivY(x,y))**

**return answer**

**def f(x,y):**

**answer = ((1-x)\*\*2) + 100\*((y-(x\*\*2))\*\*2)**

**return answer**

**def steepestdescent(x,y,n):**

**if (f(x,y)<.1):**

**return 0**

**else:**

**newX = deltaX(x,y)**

**newY = deltaY(x,y)**

**print("Wave :" + str(n))**

**print("x: " + str(newX))**

**print("y: " + str(newY))**

**print("f(x,y): " + str(f(newX, newY)))**

**return steepestdescent(newX,newY,(n+1))**

**steepestdescent(0,0.5,0)**

f(x,y) < 0.1 at iteration 1040, where

x = 0.6842420004628382

y = 0.46668078428343124

f(x,y) = 0.09993001755394404

if a larger learning rate of 0.2 is used, the weights would be adjusted to drastically after each iteration, causing it to have to be adjusted back in the opposite direction after each iteration.

Q1b.

fyy = 200

fxx = 1200x2 – 400y + 2

fxy = fyx = -400x

Using Newton’s method,

∆w(k) = -H-1(k)g(k), where H(k) is the Hessian Matrix of f(x,y) and g(k) = ∇f(w(k))

**def derivX(x,y):**

**answer = (2\*(x-1)) - (400\*x\*(y-(x\*\*2)))**

**return(answer)**

**def derivY(x,y):**

**answer = 200\*(y-(x\*\*2))**

**return(answer)**

**def derivXX(x,y):**

**answer = 2-(400\*(y-(3\*(x\*\*2))))**

**return(answer)**

**def derivYY(x,y):**

**answer = 200**

**return(answer)**

**def derivXY(x,y):**

**answer = -400\*x**

**return(answer)**

**def derivYX(x,y):**

**# print(answer)**

**return(answer)**

**def deltaX(x,y):**

**multiplier = -1/((derivXX(x,y)\*derivYY(x,y))-(derivXY(x,y)\*derivYX(x,y)))**

**answer = multiplier \* ((derivYY(x,y)\*derivX(x,y)) + (-(derivXY(x,y))\*derivY(x,y)))**

**print("DeltaX: " + str(answer))**

**return answer**

**def deltaY(x,y):**

**multiplier = -1/((derivXX(x,y)\*derivYY(x,y))-(derivXY(x,y)\*derivYX(x,y)))**

**answer = multiplier \* ((-(derivYX(x,y))\*derivX(x,y)) + (derivXX(x,y)\*derivY(x,y)))**

**print("DeltaY: " + str(answer))**

**return answer**

**def f(x,y):**

**answer = ((1-x)\*\*2) + 100\*((y-(x\*\*2))\*\*2)**

**return answer**

**def recursion(x,y,n):**

**if (f(x,y)<0.1):**

**return 0**

**else:**

**print("Wave :" + str(n))**

**newX = x + deltaX(x,y)**

**newY = y + deltaY(x,y)**

**print("x: " + str(newX))**

**print("y: " + str(newY))**

**print("f(x,y): " + str(f(newX, newY)))**

**print()**

**return recursion(newX,newY,(n+1))**

**recursion(0,0.5,1)**

f(x,y) < 0.1 at iteration 3, where

x = 0.9799025478636456

y = 0.9602089927935581

f(x,y) = 0.0004039075823841144