HW 2- Gradient Descent and the First Weekly Challenge

ENGR 3H

Due 11:59 PM, Thursday, April 18

1 Analytic Gradient Descent (20 Points)

Consider the function:

$$z = 0.02x^4 - 5x^2y + 0.01y^4 - 0.3x^3y + 0.005y^6 - x^3. \tag{1}$$

We're going to find its global minimum in a couple different ways.

1.1 Sanity Check

First, let's find what the global minimum *should* be. Compute the gradient of z and use any method you want (such as a root finder in MATLAB) to find the minimum. Show your work and note when/where you used programs to solve a step for you.

1.2

Now let's see if we can find it with gradient descent. Since this is just a polynomial function, we can find the gradient analytically without much trouble. Write a function called `analytic_grad` that takes as input the x- and y-coordinates and returns the gradient, then use that gradient in a script (`analytic_minimum.m` to find the minimum. You should use the origin as your starting point, but you should adjust the learning rate α and stopping condition as necessary to achieve a good result.

Create a plot that shows the path your script took to reach the minimum. First, create a surface plot of the function z. This can be done by first generating a mesh grid of x and y values, then generating their corresponding z values–see the MATLAB documentation for details (to keep things cleaner, I recommend setting 'EdgeColor' to 'None'). Then plot the x, y, and z values of the steps your script took to reach the minimum.

2 Numeric Gradient Descent (30 Points)

We'll now consider a slightly more complicated function:

$$z_2 = \frac{z}{0.0003x^8 + 0.002y^1 + 1}, \tag{2}$$

where z is defined in (1). While we could still find the analytic gradient for this system, we will instead approximate the gradient via finite differences. Suppose the coordinate is defined as $\vec{x} = [x_1, x_2, ..., x_n]^T$ and $z_2 = f(\vec{x})$. Then we can approximate the gradient with respect to x_i as:

$$\nabla_{x_i} f \approx \frac{f(\vec{x} + \delta x_i) - f(\vec{x} - \delta x_i)}{2\delta x_i}, \tag{3}$$

where δx_i denotes a small perturbation in the ith coordinate.

2.1

Implement numeric gradient descent to find the global minimum of z_2. Try initializing the descent from two different coordinates: $(-4, 4)$ and $(4, 4)$. What happens? What are some ways you can avoid the issue you run into?

2.2 The Weekly Challenge!

Write a function (`find_minimum.m` that takes as input a function handle $@f$ and an initial point $\vec{x}$ and returns a best guess for the global minimum. Try to make your code as efficient and accurate as possible! Consider implementing strategies for avoiding local minima, selecting a learning rate, etc. You can validate your code by running it on z and z_2 to confirm you get what you're expecting to get. The code should work for any size space (not necessarily just 2-D or 3-D!)