Gradient Descent

EE24BTECH11026 - G.Srihaas

Question

Find all points of local maxima and local minima of the function f given by

$$f(x) = x^3 - 3x + 3. \tag{1}$$

Theoretical Method

We can find the local maximum and minimum of the polynomial function $f(x)$ using the first and second derivatives.
The first derivative helps to identify the stationary points, i.e., the points where the function is maximum or minimum.
The second derivative helps to identify if we get a maximum or minimum at the said stationary points.

$$
\begin{aligned}
f(x) &= x^3 - 3x + 3, && (2)\\
f'(x) &= 3x^2 - 3 = 3(x-1)(x+1), && (3)\\
f'(x) &= 0, && (4)\\
x &= 1, && (5)\\
x &= -1. && (6)
\end{aligned}
$$

Hence, the stationary points are -1 and 1.

Second Derivative Test

Now, let's find the second derivative:

$$f''(x) = 6x, \tag{7}$$
$$f''(-1) = 6(-1) = -6 < 0, \tag{8}$$
$$f''(1) = 6(1) = 6 > 0. \tag{9}$$

Thus:

- At $x = -1$, $f''(-1) < 0$, so there is a local maximum.
- At $x = 1$, $f''(1) > 0$, so there is a local minimum.

$\therefore$ The point of local maximum is $x = -1$, and the point of local minimum is $x = 1$.

Computational Logic: Gradient Descent Method

Gradient descent is an iterative optimization algorithm used to find the minimum or maximum of a function by iteratively adjusting the input value in the direction of the steepest descent (for minimum) or ascent (for maximum).
Here we will apply the gradient descent method to find the local minima and maxima of the cubic function:

$$f(x) = x^3 - 3x + 3 \tag{10}$$

Gradient Descent Method

The update rule for gradient descent is given by the difference equation:

$$x_{n+1} = x_n - \eta . f'(x_n) \tag{11}$$

where:

- x_n is the current value of x,
- x_{n+1} is the updated value of x,
- η is the learning rate (a small positive constant),
- $f'(x_n)$ is the derivative of the function $f(x)$ evaluated at x_n, also called the gradient.

We initialize the algorithm with starting points near these values and observe convergence to the critical points.

Gradient Descent Algorithm

- Initialize x_0
- Update x iteratively using the formula:

$$x_{n+1} = x_n - \eta.(3x_n^2 - 3) \tag{12}$$
$$x_{n+1} = (-3\eta)x_n^2 + x_n + 3\eta \tag{13}$$

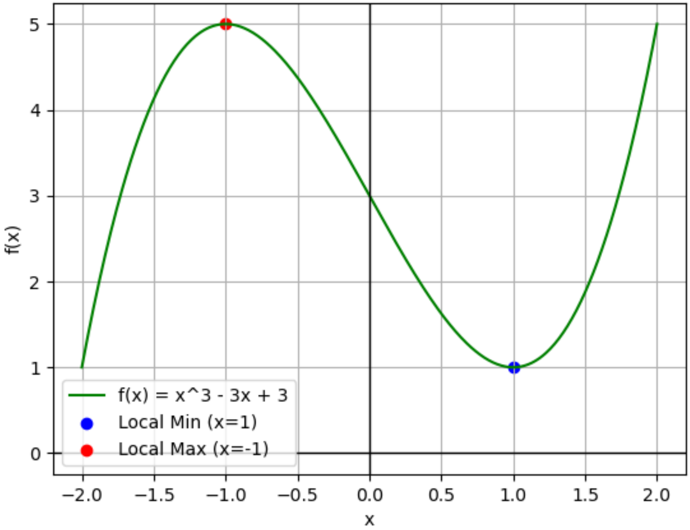
  until the change in x is smaller than a chosen threshold.
- The algorithm stops when $|x_{n+1} - x_n|$ is sufficiently small.

1. Finding minimum
   - At each step,the algorithm computes the gradient (Derivatives) of the function at the current point.
   - The gradient points in the direction of the steepest ascent, so you move in the opposite direction of the gradient to decrease the function's value.
   - You update the current point by subtracting the gradient (scaled by a learning rate, which controls how large a step is taken) and this process is repeated until the algorithm converges to a point where the gradient is nearly zero, indicating a local minimum.
2. Finding maximum
   - Repeat the same with negative of the function,the minimum of negative function would be the maximum of positive.

Stationary Points Obtained

- The local minimum occurs at $x = 1$.
- The local maximum occurs at $x = -1$.

Plot



f(x) = x^3 - 3x + 3
Local Min (x=1)
Local Max (x=-1)
f(x)
x
5
4
3
2
1
0
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

C-code

https://github.com/Srihaas15/EE1003/12.6.ex-29/codes/gradient$_{d}escent.c$

Python-code

https://github.com/Srihaas15/EE1003/12.6.ex-29/codes/plot$_{g}radient_{d}escent.py$