Title : Implement Gradient Descent Algorithm to find the local minima of a function.

For example, find the local minima of the function $y=(x+3)^2$ starting from the point x=2.

Prerequesite:

1.Data preprocessing

2. Gradient Descent functioning.

Theory:

Gradient descent is a prominent optimization approach in machine learning for minimizing a model's loss function. In layman's terms, it entails repeatedly changing the model's parameters until the ideal range of values is discovered that minimizes the loss function. The method operates by making tiny steps in the direction of the loss function's negative gradient, or, more specifically, the path of steepest descent. The learning rate, a hyperparameter that regulates the algorithm's trade-off between speed and accuracy, affects the size of the steps. Many machine learning methods, including linear regression, logistic regression, and neural networks, to mention a few, employ gradient descent. Its primary application is in model training, where the goal is to minimize the difference between the target variable's anticipated and actual values. In this post, we will look at implementing a gradient descent in Python to find a local minimum.

How to implement it :

- First, we import the necessary libraries.
- Defining the functions and derivative of it.
- Next, we will apply the gradient descent function.
- After applying the function, we will set the parameters to find the local minima,
- At last, we will plot the graph of output.

Conclusion

In conclusion, to find the local minimum of a function, Python makes use of the effective optimization process known as gradient descent. Gradient descent updates the input value repeatedly in the direction of the steepest fall until it achieves the lowest by computing the derivative of the function at each step.