Aakriti Singh 20190802006

Fundamentals of AI & ML

Monsoon Semester V 2021-22

Lab - 1

Date: 9 September 2021

Topic: Optimization Algorithm

AIM

To find the parameters or coefficients of a function where the function has a minimum value using gradient descent optimization algorithms.

THEORY

Gradient Descent is an optimization algorithm for finding a local or global minimum of a differentiable function. Gradient descent is simply used in machine learning to find the values of a function's parameters (coefficients) that minimize a cost function as far as possible.

A local minimum is a point where our function is lower than all neighboring points. It is not possible to decrease the value of the cost function by making infinitesimal steps.

A global minimum is a point that obtains the absolute lowest value of our function, but global minima are difficult to compute in practice.

The learning rate determines the size of the steps that are taken by the gradient descent algorithm.

EXPERIMENT

We take the equation $y = (x+5)^2$, and iterate to find its parameters

Step 1: Initialize x = -3. Then, find the gradient of the function, dy/dx = 2*(x+5).

Step 2: Move in the direction of the negative of the gradient. We assume a learning rate to define how much to move at each step.
Let us assume the **learning rate → 0.01**

Step 3: We perform iterations of gradient descent

$$x_1 = x_0 - (\text{learning rate}) * (dy/dx)$$
$$x_2 = x_1 - (\text{learning rate}) * (dy/dx)$$

Step 4: We can observe that the X value is slowly decreasing and should converge to -5 (the local minima).

Let us set a precision variable in our algorithm which calculates the difference between two consecutive “x” values. If the difference between x values from 2 consecutive iterations is lesser than the precision we set, stop the algorithm!

PROGRAM CODE

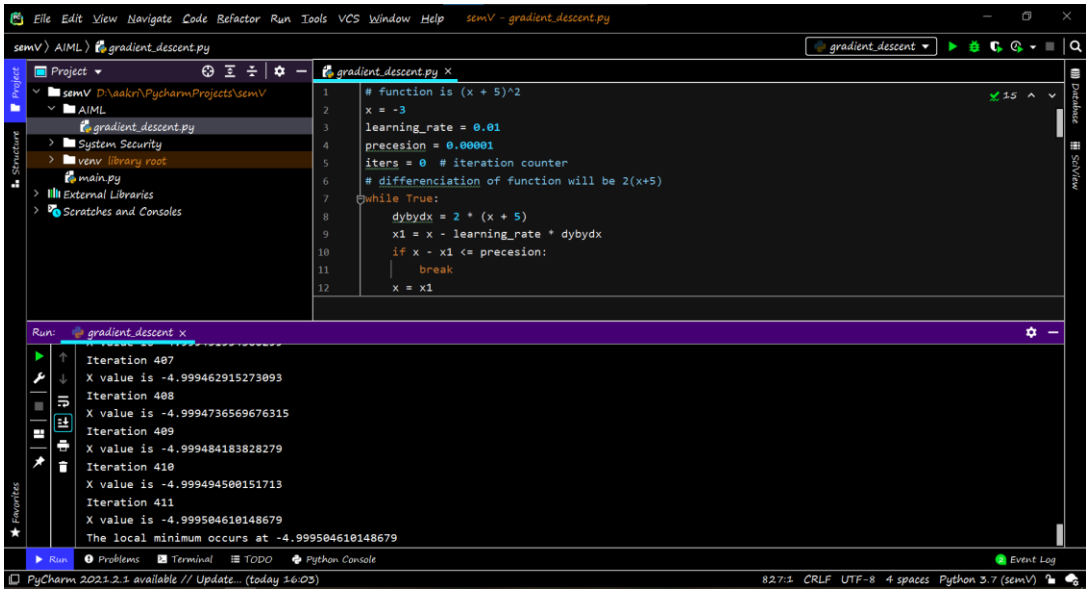
```
# function is (x + 5)^2

x = -3
learning_rate = 0.01
precesion = 0.00001
iters = 0  # iteration counter
# differenciation of function will be 2(x+5)

while True:
    dybydx = 2 * (x + 5)
    x1 = x - learning_rate * dybydx
    if x - x1 <= precesion:
        break
    x = x1
    iters = iters + 1  # iteration count
    print("Iteration", iters, "\nX value is", x)  # Print iterations

print("The local minimum occurs at", x)
```

OUTPUT



OBSERVATION

The algorithm runs for 411 iterations before it terminates.

Pros and cons of gradient descent

- A simple algorithm that is easy to implement and each iteration is cheap; we just need to compute a gradient
- However, it's often slow because many interesting problems are not strongly convex
- Cannot handle non-differentiable functions (biggest downside)

CONCLUSION

Gradient descent is a simple algorithm it can be applied to any function to optimize it. In this lab, we learnt how the gradient descent algorithm works and implemented it from scratch in python.