72017912C
COMPBEB326

GROUP B : ASSIGNMENT No. 4

Page No.
Date

Title: Gradient Descent Algorithm

Objective: To implement Gradient Descent Algorithm for finding the local minima of a given function.

Problem Statement:

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 $x=2$.

Theory:

Gradient Descent:

Gradient descent is known as one of the most commonly used optimization algorithm to train machine learning models by means of minimizing errors between actual & expected results. Further, gradient descent is also used to train Neural Networks.

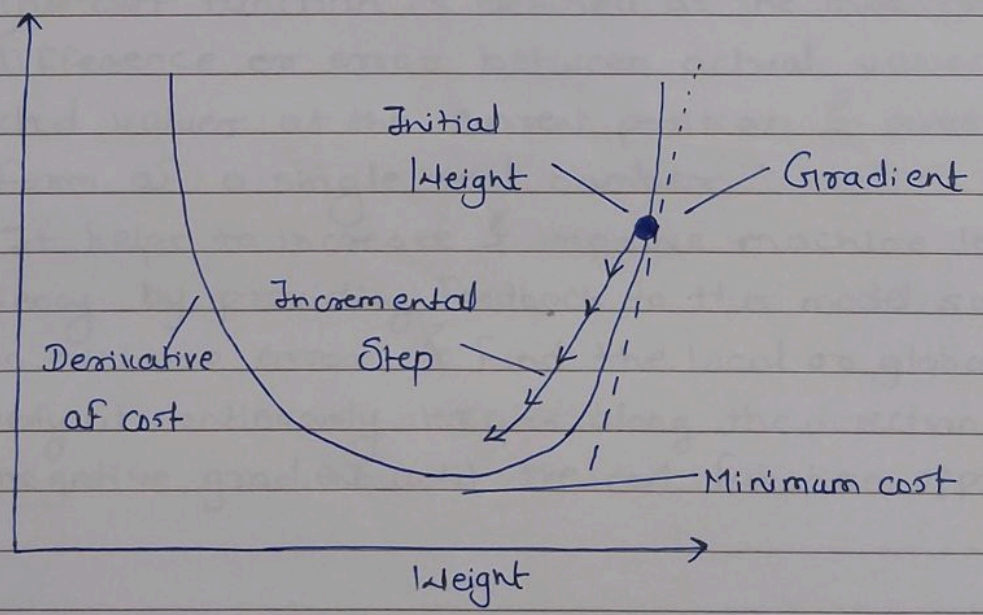
In mathematical terminology, Optimization algorithm refers to the task of minimizing /maximizing an objective function $f(x)$ parameterized by x. Similarly, in machine learning, optimization is the task of minimizing the cost function parameterized by the model's parameters. The main objective of gradient descent is to minimize the convex function using iteration of parameter updates. Once these machine learning models are optimized, these models can be used as powerful tools for Artificial intelligence & various computer science applications.

Page No.
Date

Gradient descent was initially discovered by "Augustin-Louis Cauchy" in mid of 18th century. Gradient Descent is defined as one of the most commonly used iterative optimization algorithms of machine learning to train the machine learning & deep learning models. It helps in finding the local minimum of a function.

The best way to define the local minimum or local maximum of a function using gradient descent is as follows.

- If we move towards a negative gradient or away from the gradient of the function at the current point, it will give the local minimum of that function.
- Whenever we move towards a positive gradient or towards the gradient of the function at the current point, we will get the local maximum of that function.



This entire procedure is known as Gradient Descent, which is also known as Steepest descent. The main objective of using a gradient descent algorithm is to minimize the cost function using iteration. To achieve this goal, it performs two steps iteratively:

1) Calculates the first-orders derivative of the function to compute the gradient or slope of that function.
2) Move away from the direction of the gradient, which means slope increased from the current point by alpha times, where Alpha is defined as Learning rate. It is a tuning parameter in the optimization process which helps to decide the length of the steps.

Cost Function:

The cost function is defined as the measurement of difference or error between actual values & expected values at the current position & present in the form of a single real number.

It helps to increase & improve machine learning efficiency by providing feedback to this model so that it can minimize error & find the local or global minimum. Furtherly, it continuously iterates along the direction of the negative gradient until the cost function approaches zero.

Page No.
Date

Types of Gradient Descent:

1) Batch Gradient Descent
2) Stochastic Gradient Descent
3) Mini Batch Gradient Descent

Conclusion:

Successfully implemented Gradient Descent Algorithm for finding the local minimum of a function.

17/10/22

In [1]:
```
current_x = 2 # The algorithm starts at x=3
rate = 0.01 # Learning rate
precision = 0.000001 #This tells us when to stop the algorithm
previous_step_size = 1
max_iters = 10000 # maximum number of iterations
iters = 0 #iteration counter
df = lambda x: 2*(x+3) #Gradient of our function
```

In [2]:
```
while previous_step_size > precision and iters < max_iters: #When Previous Step
    previous_x = current_x #Store current x value in prev_x
    current_x = current_x - rate * df(previous_x) #Grad descent
    previous_step_size = abs(current_x - previous_x) #Change in x
    iters = iters+1 #iteration count
    print("Iteration",iters,"\nX value is",current_x) #Print iterations

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

```
Iteration 1
X value is 1.9
Iteration 2
X value is 1.8019999999999998
Iteration 3
X value is 1.70596
Iteration 4
X value is 1.6118408
Iteration 5
X value is 1.519603984
Iteration 6
X value is 1.42921190432
Iteration 7
X value is 1.3406276662336
Iteration 8
X value is 1.253815112908928
Iteration 9
X value is 1.1687388106507495
Iteration 10
```

In [ ]: