Implement Gradient Descent in Python

Rohan Joseph Follow

Jun 3, 2018 · 3 min read

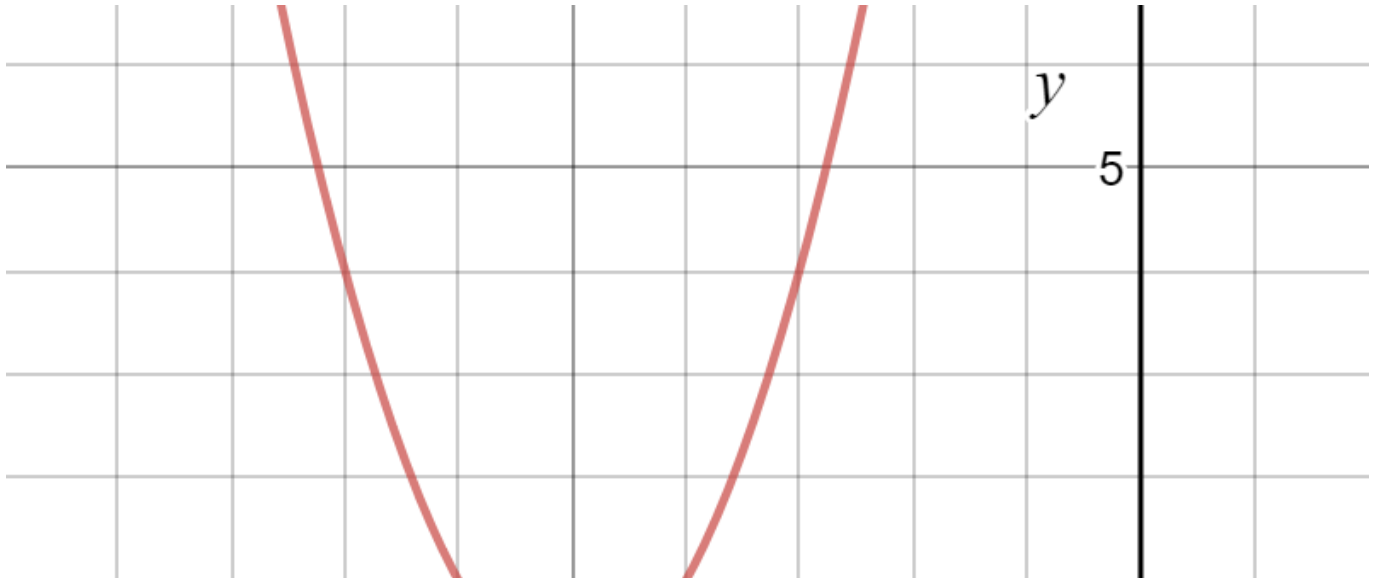
What is gradient descent ?

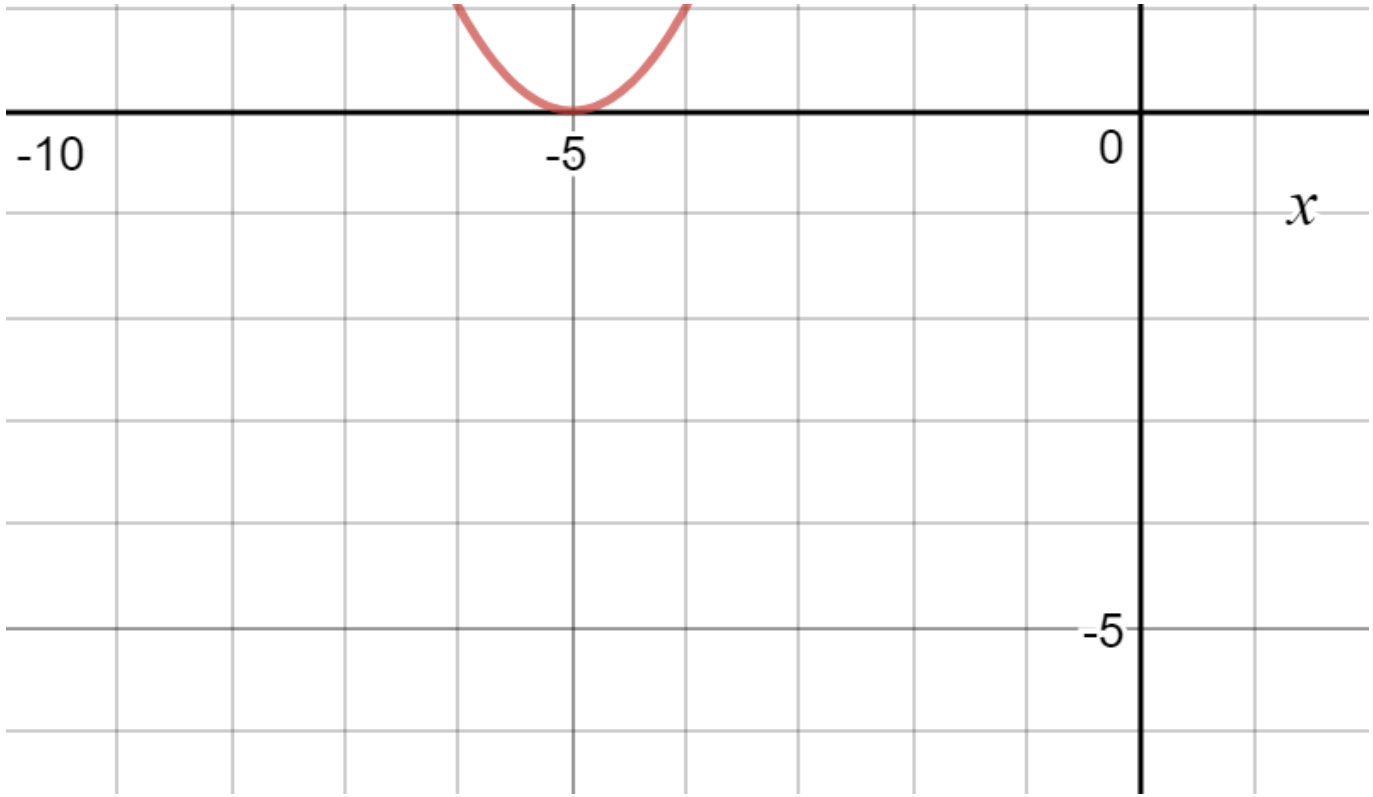
It is an optimization algorithm to find the minimum of a function. We start with a random point on the function and move in the **negative direction** of the **gradient of the function** to reach the **local/global minima**.

Homer descending !

Example by hand :

Question : Find the local minima of the function $y=(x+5)^2$ starting from the point $x=3$





Solution : We know the answer just by looking at the graph. $y = (x+5)^2$ reaches it's minimum value when $x = -5$ (i.e when $x=-5, y=0$). Hence $x=-5$ is the local and global minima of the function.

Now, let's see how to obtain the same numerically using gradient descent.

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 (Why?). But wait, how much to move? For that, we require a learning rate. Let us assume the **learning rate → 0.01**

Step 3 : Let's perform 2 iterations of gradient descent

Initialize Parameters :

$$X_0 = 3$$

$$Learning\ rate = 0.01$$

$$\frac{dy}{dx} = \frac{d}{dx}(x+5)^2 = 2 * (x+5)$$

Iteration 1 :

$$X_1 = X_0 - (learning\ rate) * (\frac{dy}{dx})$$

$$X_1 = 3 - (0.01) * \big(2 * (3+5)\big) = 2.84$$

Iteration 2 :

$$X_2 = X_1 - (learning\ rate) * (\frac{dy}{dx})$$

$$X_2 = 2.84 - (0.01) * \big(2 * (2.84 + 5)\big) = 2.6832$$

Step 4 : We can observe that the X value is slowly decreasing and should converge to -5 (the local minima). However, how many iterations should we perform?

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 !

Gradient descent in Python :

Step 1 : Initialize parameters

```
cur_x = 3 # 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+5) #Gradient of our function
```

Step 2 : Run a loop to perform gradient descent :

i. Stop loop when difference between x values from 2 consecutive iterations is less than 0.000001 or when number of iterations exceeds 10,000

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

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

Output : From the output below, we can observe the x values for the first 10 iterations- which can be cross checked with our calculation above. The algorithm runs for 595 iterations before it terminates. The code and solution is embedded below for reference.

```
Iteration 1
X value is 2.84
```

```
X value is 2.84
Iteration 2
X value is 2.6832
Iteration 3
X value is 2.529536
Iteration 4
X value is 2.37894528
Iteration 5
X value is 2.2313663744
Iteration 6
X value is 2.0867390469119997
Iteration 7
X value is 1.9450042659737599
Iteration 8
X value is 1.8061041806542846
Iteration 9
X value is 1.669982097041199
Iteration 10
X value is 1.5365824551003748
```

. . .

Connect on LinkedIn.

```
X value is -4.277241855298307
Iteration 120
X value is -4.291697018192341
Iteration 121
X value is -4.305863077828494
Iteration 122
```

```
X value is -4.319745816271924
Iteration 123
X value is -4.333350899946486
Iteration 124
X value is -4.3466838819475555
Iteration 125
X value is -4.359750204308605
Iteration 126
X value is -4.372555200222433
Iteration 127
X value is -4.385104096217984
Iteration 128
X value is -4.3974020142936245
Iteration 129
X value is -4.409453974007752
Iteration 130
X value is -4.421264894527597
Iteration 131
X value is -4.432839596637045
Iteration 132
X value is -4.444182804704305
Iteration 133
X value is -4.4552991486102185
```

Gradient_descent.ipynb hosted with ♡ by GitHub view raw

Deep Learning Machine Learning Data Science Analytics Python