Implement Gradient Descent in Python

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**.

Example by hand :

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

Solution : We know the answer just by looking at the graph. y = (x+3)² reaches it's minimum value when x = -3 (i.e when x=-3, y=0). Hence x=-3 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 =2. Then, find the gradient of the function, dy/dx = 2*(x+3).

Step 2 : Move in the direction of the negative of the gradient. Stopping condition: 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:

$$x_0 = 2$$

Learning rate = 0.01

$$\frac{dy}{dx} = \frac{d}{dx}(x+3)^2 = 2 \times (x+3)$$

Iteration 1:

$$x_1 = x_0 - (learning\ rate) \times \left(\frac{dy}{dx}\right)$$

$$x_1 = 2 - (0.01) \times \big(2 \times (2+3)\big) = 1.9$$

Iteration 2:

$$x_2 = x_1 - (learning\ rate) \times \left(\frac{dy}{dx}\right)$$

$$x_2 = 1.9 - (0.01) \times \big(2 \times (1.9+3)\big) = 1.802$$

Step 4 : We can observe that the X value is slowly decreasing and should converge to -3 (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 = 2 # The algorithm starts at x=2
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
```

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

.

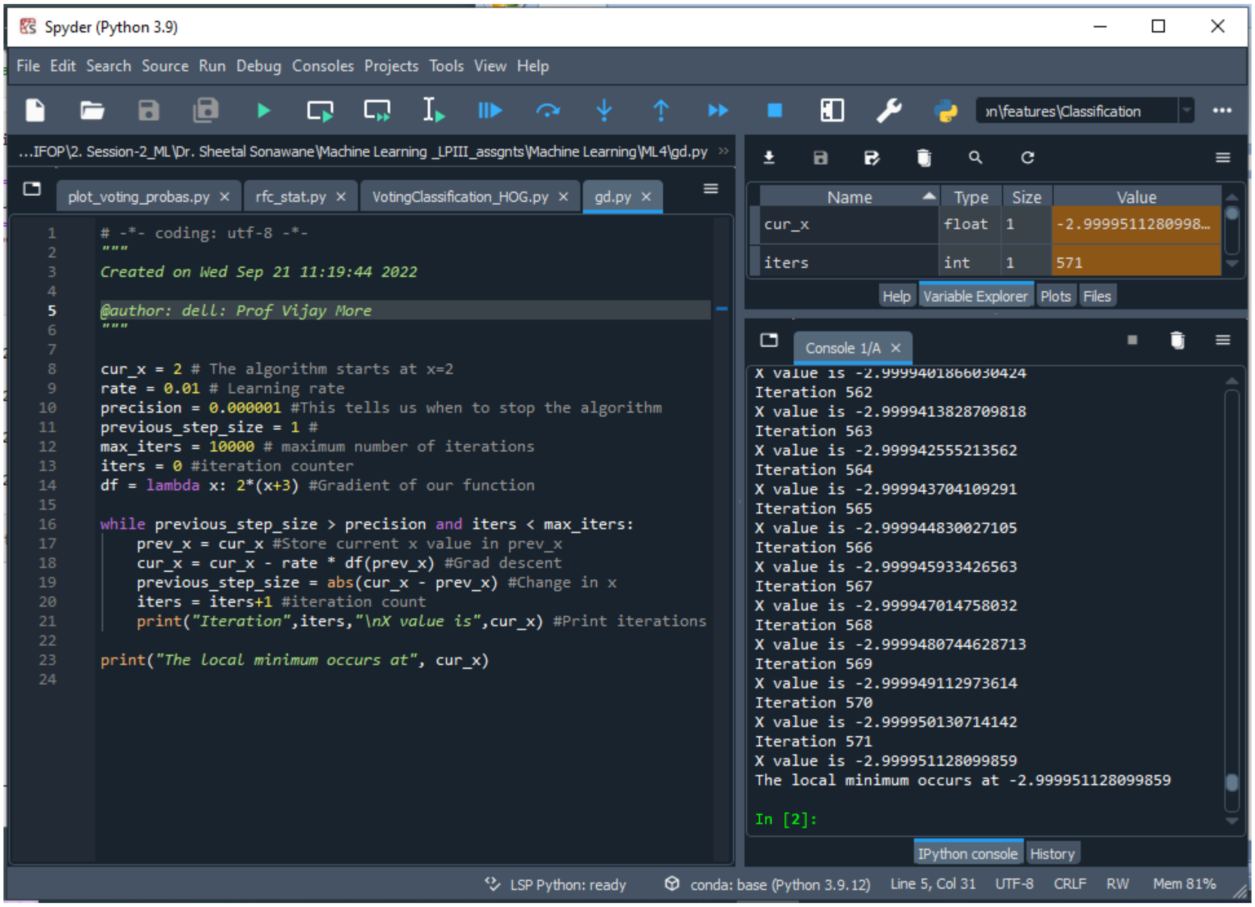
```
while previous_step_size > precision and iters < max_iters:
      prev_x = cur_x #Store current x value in prev_x
      cur_x = cur_x - rate * df(prev_x) #Grad descent
      previous_step_size = abs(cur_x - prev_x) #Change in x
      iters = iters+1 #iteration count
      print("Iteration",iters,"\nX 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 571 iterations before it terminates. The code and solution is embedded below for reference.

```
X value is -2.9999364493143186
Iteration 559
X value is -2.9999377203280324
Iteration 560
X value is -2.999938965921472
Iteration 561
X value is -2.9999401866030424
Iteration 562
X value is -2.9999413828709818
Iteration 563
X value is -2.999942555213562
Iteration 564
X value is -2.999943704109291
Iteration 565
X value is -2.999944830027105
Iteration 566
X value is -2.999945933426563
Iteration 567
X value is -2.999947014758032
```

Iteration 568
X value is -2.9999480744628713
Iteration 569
X value is -2.999949112973614
Iteration 570
X value is -2.999950130714142
Iteration 571
X value is -2.999951128099859
The local minimum occurs at -2.999951128099859

Screenshot of program and output



Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
…n\features\Classification
…IFOP\2. Session-2_ML\Dr. Sheetal Sonawane\Machine Learning _LPIII_assgnts\Machine Learning\ML4\gd.py
plot_voting_probas.py
rfc_stat.py
VotingClassification_HOG.py
gd.py
Help
Variable Explorer
Plots
Files
Console 1/A
IPython console
History
LSP Python: ready
conda: base (Python 3.9.12)
Line 5, Col 31
UTF-8
CRLF
RW
Mem 81%

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Sep 21 11:19:44 2022

@author: dell: Prof Vijay More
"""

cur_x = 2 # The algorithm starts at x=2
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

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)
```

Name	Type	Size	Value
cur_x	float	1	-2.9999511280998…
iters	int	1	571

```
X value is -2.9999401866030424
Iteration 562
X value is -2.9999413828709818
Iteration 563
X value is -2.999942555213562
Iteration 564
X value is -2.999943704109291
Iteration 565
X value is -2.999944830027105
Iteration 566
X value is -2.999945933426563
Iteration 567
X value is -2.999947014758032
Iteration 568
X value is -2.9999480744628713
Iteration 569
X value is -2.999949112973614
Iteration 570
X value is -2.999950130714142
Iteration 571
X value is -2.999951128099859
The local minimum occurs at -2.999951128099859

In [2]:
```

Source: https://towardsdatascience.com/implement-gradient-descent-in-python-9b93ed7108d1