Lab_3_Convex_Optimisation

August 24, 2021

1 Lab 3 : Convex Optimisation

Gradient Descent

Write the code following the instructions to obtain the desired results

2 Import all the required libraries

[ ]:

```
import numpy as np
import matplotlib.pyplot as plt
```

3 Find the value of x at which $f(x)$ is minimum :

1. Find x analytically
2. Write the update equation of gradient descent
3. Find x using gradient descent method

Example 1 : $f(x) = x^2 + x + 2$

Analytical :

$$\frac{d}{dx}f(x) = 2x + 1 = 0$$

$$\frac{d^2}{dx^2}f(x) = 2\ (Minima)$$

$$x = -\frac{1}{2}\ (analytical\ solution)$$

Gradient Descent Update equation :

$$x_{init} = 4$$

$$x_{updt} = x_{old} - \lambda(\frac{d}{dx}f(x)|x = x_{old})$$

$$x_{updt} = x_{old} - \lambda(2x_{old} + 1)$$

Gradient Descent Method :

Follow the below steps and write your code in the block below

1. Generate x, 1000 data points from -10 to 10

2. Generate and Plot the function $f(x) = x^2 + x + 2$
3. Initialize the starting point (x_{init}) and learning rate (λ)
4. Use Gradient descent algorithm to compute value of x at which the function $f(x)$ is minimum
5. Also vary the learning rate and initialisation point and plot your observations

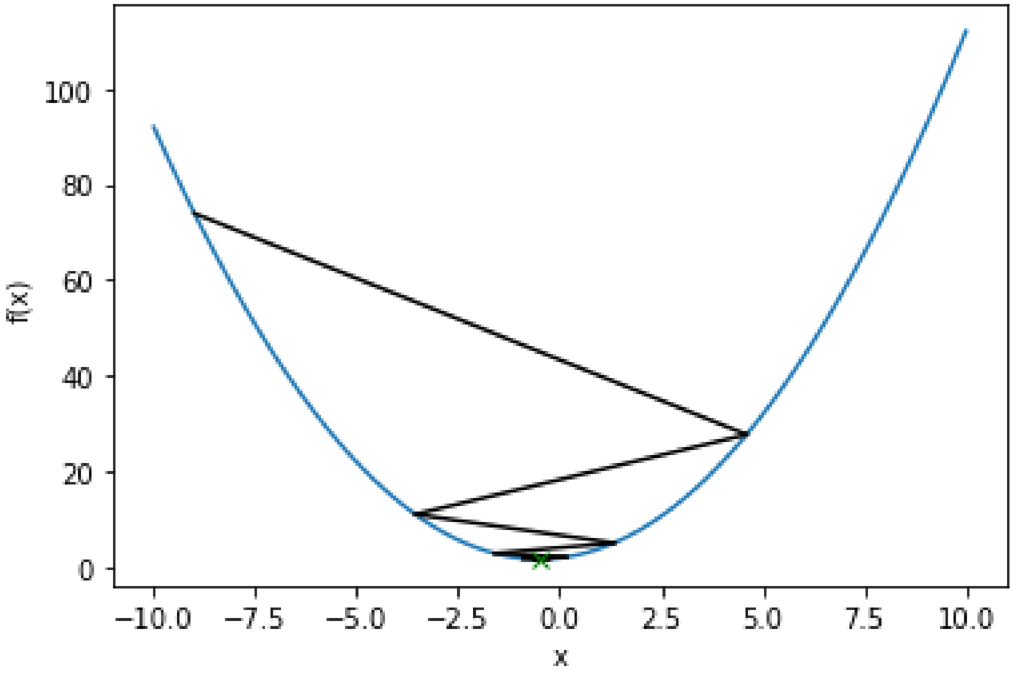
[ ]:

```
## Write your code here
```

```
Value of x at which the finction f(x) is minimum : -0.5000002435350299
```

[ ]:

```
Text(0, 0.5, 'f(x)')
```



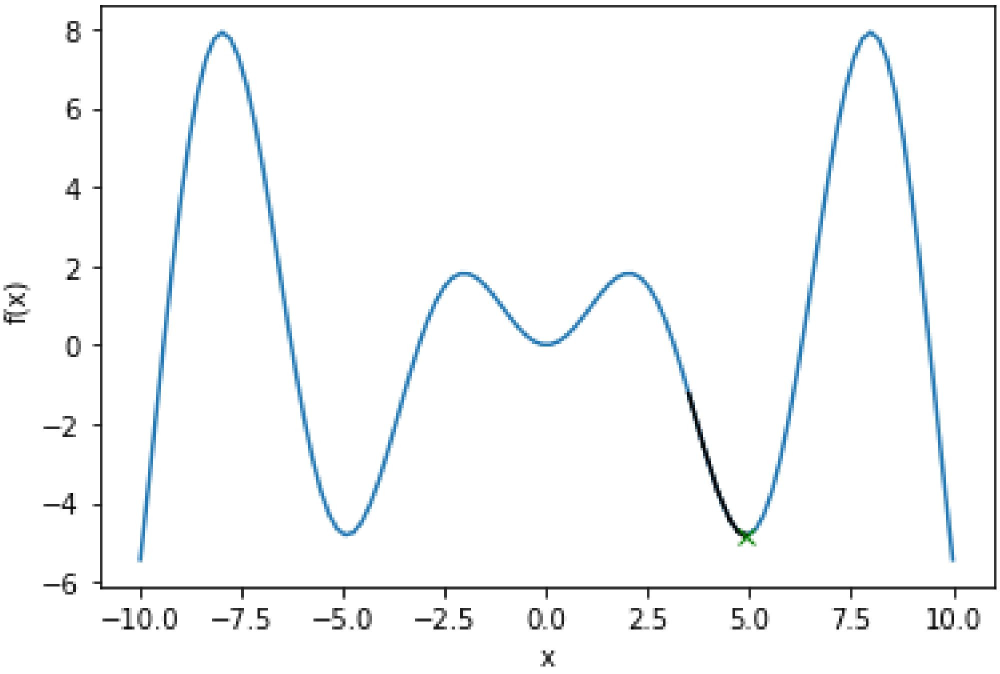
Example 2 : $f(x) = xsinx$
Analytical : Find solution analytically
Gradient Descent Update equation : Write Gradient descent update equations
Gradient Descent Method :
Follow the below steps and write your code in the block below

1. Generate x, 1000 data points from -10 to 10
2. Generate and Plot the function $f(x) = x^2 + x + 2$
3. Initialize the starting point (x_{init}) and learning rate (λ)
4. Use Gradient descent algorithm to compute value of x at which the function $f(x)$ is minimum
5. Also vary the learning rate and initialisation point and plot your observations

[ ]:
```
## Write your code here
```

```
The value of x ata which function f(x) is minimum is : 4.913179571739345
```

[ ]:
```
Text(0, 0.5, 'f(x)')
```



4 Find the value of x and y at which $f(x, y)$ is minimum :

Example 1 : $f(x, y) = x^2 + y^2 + 2x + 2y$

Gradient Descent Method :

Follow the below steps and write your code in the block below

1. Generate x and y, 1000 data points from -10 to 10
2. Generate and Plot the function $f(x, y) = x^2 + y^2 + 2x + 2y$
3. Initialize the starting point (x_{init}, y_{init}) and learning rate (λ)
4. Use Gradient descent algorithm to compute value of x and y at which the function $f(x, y)$ is minimum
5. Also vary the learning rate and initialisation point and plot your observations

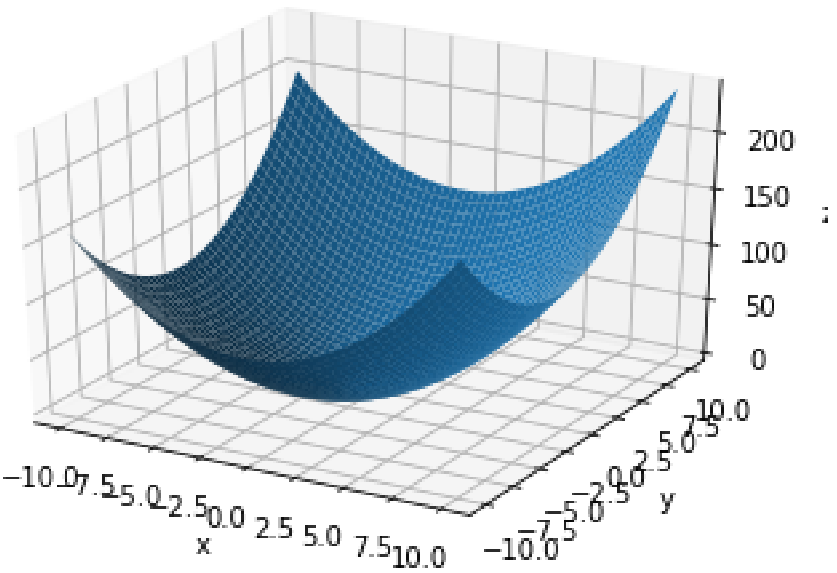
[ ]:
```
## Write your code here (Ignore the warning)
```

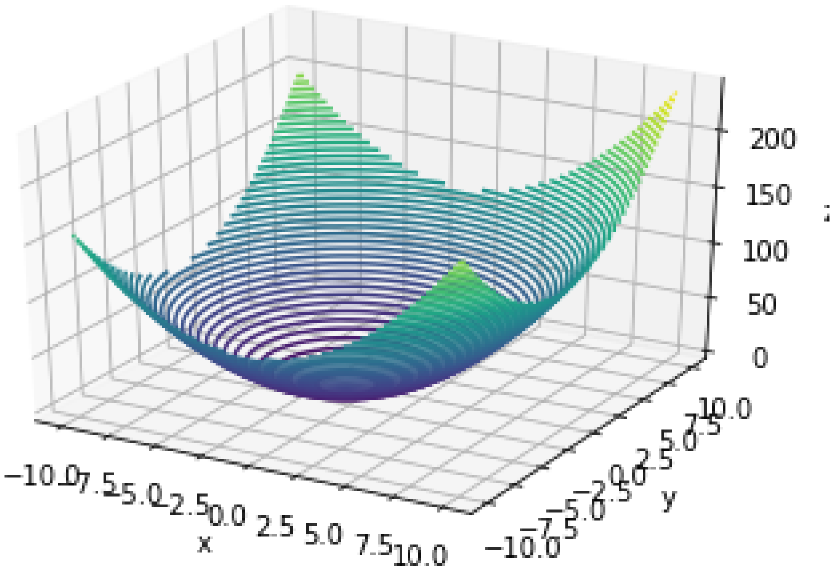
```
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py:136:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
```

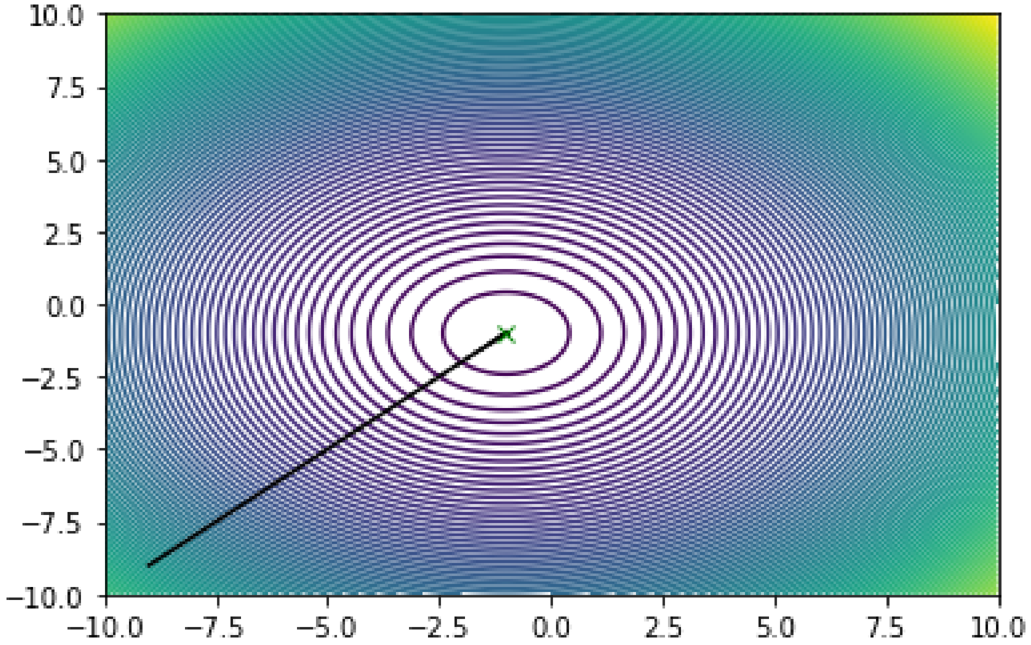
```
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray
  return array(a, dtype, copy=False, order=order, subok=True)

--------------------------------------------------------------------------------
-------
The value of x and y at which the function f(x,y) is minimum is :
-1.0000025711008707,-1.0000025711008707
--------------------------------------------------------------------------------
-------
```

[ ]:

```
[<matplotlib.lines.Line2D at 0x7ff55c0e3790>]
```







Example 2 : $f(x,y) = xsin(x) + ysin(y)$
Gradient Descent Method :
Follow the below steps and write your code in the block below

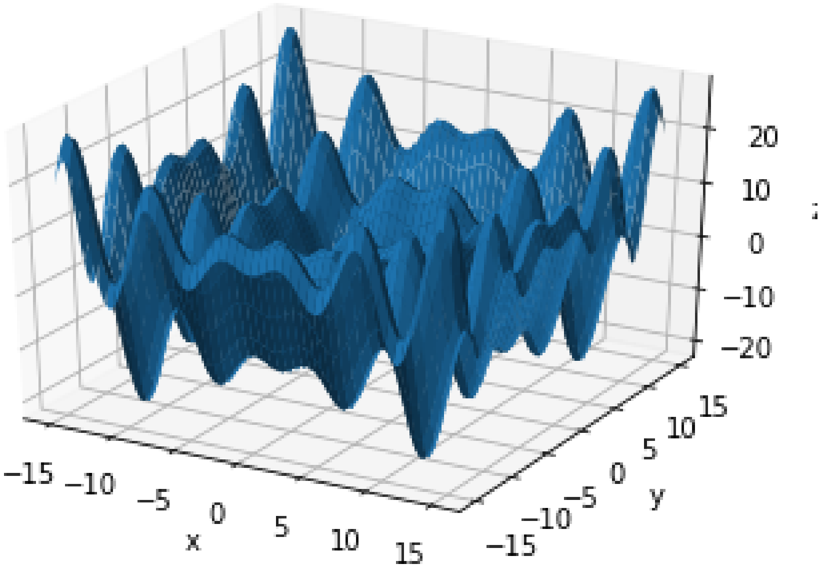
1. Generate x and y, 1000 data points from -10 to 10
2. Generate and Plot the function $f(x,y) = xsin(x) + ysin(y)$

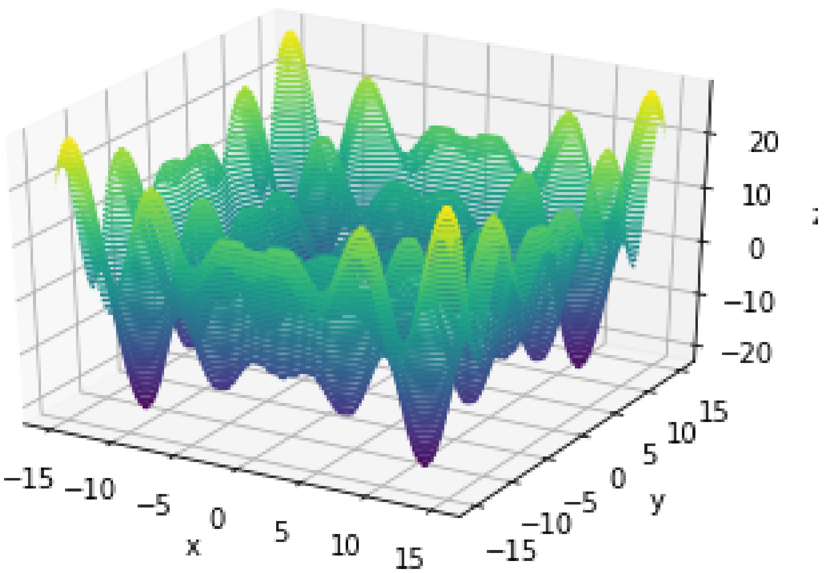
3. Initialize the starting point (x_{init},y_{init}) and learning rate (λ)
4. Use Gradient descent algorithm to compute value of x and y at which the function $f(x, y)$ is minimum
5. Also vary the learning rate and initialisation point and plot your observations

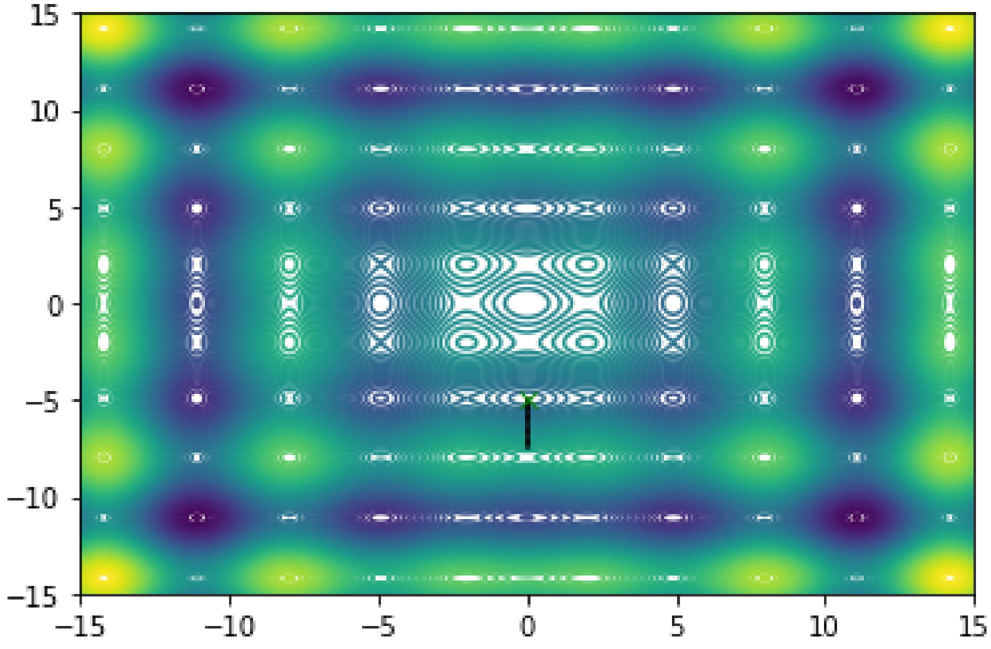
[ ]:
```
## Write your code here (Ignore the warning)
```

```
/usr/local/lib/python3.7/dist-packages/numpy/core/_asarray.py:136:
VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences
(which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths
or shapes) is deprecated. If you meant to do this, you must specify
'dtype=object' when creating the ndarray
  return array(a, dtype, copy=False, order=order, subok=True)

--------------------------------------------------------------------------------
-----------
The value of x and y at which the function f(x,y) is minimum :
0.0,-4.91318121457257
--------------------------------------------------------------------------------
-----------
```

[ ]:
```
[<matplotlib.lines.Line2D at 0x7ff55327e1d0>]
```




20
10
0
−10
−20
z
−15
−10
−5
0
5
10
15
x
15
10
5
0
−5
−10
−15
y


15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15