

# Topics in Deep Learning

## Lab Assignment 02



*By*

**Tilak Nanavati**  
**201811001**

### Problem Statement :

1. Write the gradient descent optimization code for minimizing the following functions and find the values of  $x$  for which  $f(x)$  is minimum.

a.  $f(x) = x^2$

b.  $f(x) = (x+1)^2$

c.  $f(x_1, x_2) = x_1^2 + x_2^2$

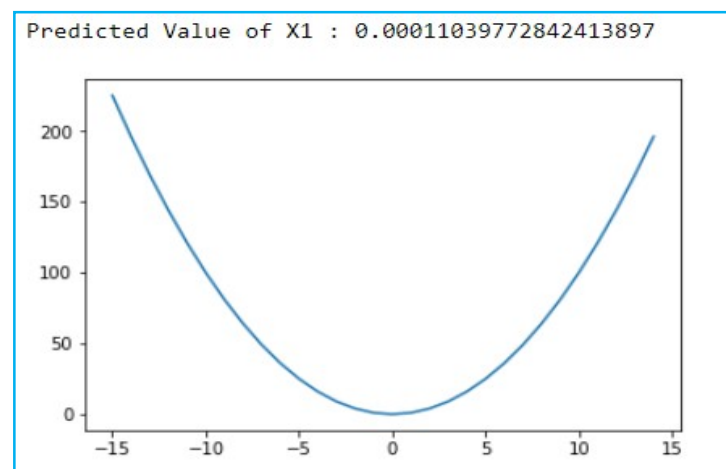
Solutions:

#### 1.a

```
#f(x) = x^2
import matplotlib.pyplot as plt
alpha = 0.01
x = [i*i for i in range(-15,15)]
y = [i for i in range(-15,15)]
#print(x)
plt.plot(y,x)

x1 = 1
for i in range(451):
    x1 = x1 - alpha*(2*x1)
print("Predicted Value of X1 :",x1)
```

Result :



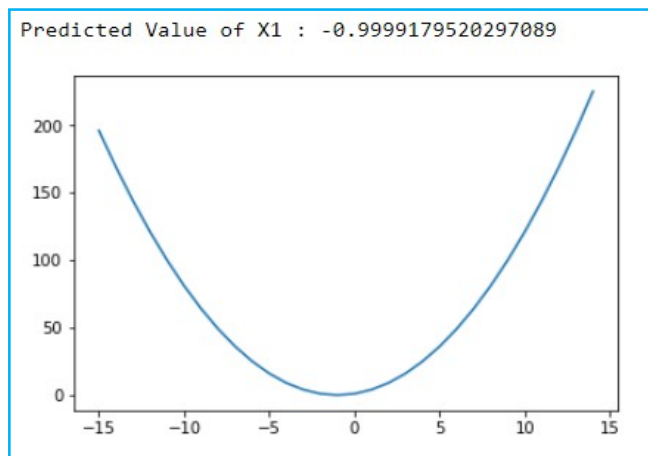
## 1.b

```
#f(x) = x^2
import matplotlib.pyplot as plt
alpha = 0.01
x = [(i+1)**2 for i in range(-15,15)]
y = [i for i in range(-15,15)]
#print(x)
plt.plot(y,x)

x1 = 1
for i in range(500):
    x1 = x1 - alpha*(2*(x1+1))

print("Predicted Value of X1 :",x1)
```

**Result :**



## 1.c

```
import numpy as np
def f(x, y):
    return ((x**2+y**2))

x1 = [i for i in range(-15,15)]
x2 = [i for i in range(-15,15)]

X, Y = np.meshgrid(x1,x2)
Z = f(X,Y)
alpha = 0.01

#print(Z)
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap='viridis', edgecolor='green')
ax.set_xlabel('x1')
ax.set_ylabel('x2')
ax.set_zlabel('y');
```

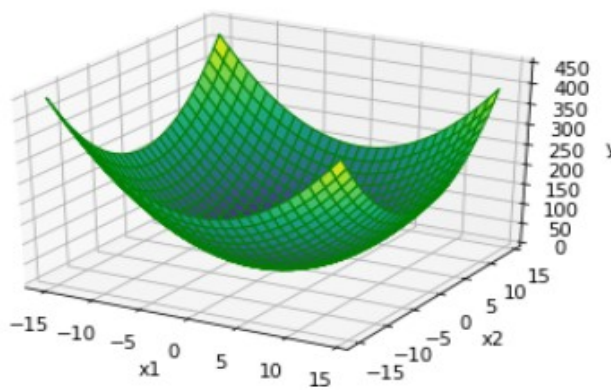
```
new_x1=1  
new_x2=1
```

```
for i in range(450):  
    new_x1 = new_x1 - alpha*(2*new_x1)  
    new_x2 = new_x2 - alpha*(2*new_x2)
```

```
print("Predicted Value of X1:",new_x1)
```

```
print("Predicted Value of X2:",new_x2)
```

```
Predicted Value of X1: 0.00011265074328993772  
Predicted Value of X2: 0.00011265074328993772
```



## 2. Check the convexity of above functions between the arbitrary interval [a,b] by using following inequality, (t has to be between 0 and 1)

**Solution :**

```
def f1(a,b,t):
    return (t*a + (1-t)*b)**2

def f2(a,b,t):
    #print("*****",t*((a)**2))
    return (t*((a)**2)+(1-t)*(b**2))

x = np.arange(0,1,0.01)

flag = 0
count = 0
for t in x:
    count+=1
    print(count,": L.H.S :",f1(1,-1,t)," |   R.H.S :",f2(1,-1,t))

    if(f1(1,-1,t)>f2(1,-1,t)):
        flag = 1
        break

print()
if(flag==0):
    print("The Given Function is : Convex")
else:
    print("The Given Function is : Non-Convex")
```

**Result :**

```
1 : L.H.S : 0.0 |   R.H.S : 0.0
2 : L.H.S : 0.0004000000000000007 |   R.H.S : 0.04
3 : L.H.S : 0.0016000000000000003 |   R.H.S : 0.08
4 : L.H.S : 0.00360000000000000064 |   R.H.S : 0.12
5 : L.H.S : 0.0064000000000000012 |   R.H.S : 0.16
6 : L.H.S : 0.0100000000000000018 |   R.H.S : 0.2
7 : L.H.S : 0.0144000000000000026 |   R.H.S : 0.24
8 : L.H.S : 0.0196000000000000034 |   R.H.S : 0.28
9 : L.H.S : 0.0255999999999999973 |   R.H.S : 0.32
10 : L.H.S : 0.032399999999999998 |   R.H.S : 0.36
11 : L.H.S : 0.039999999999999998 |   R.H.S : 0.4
12 : L.H.S : 0.048399999999999999 |   R.H.S : 0.44
13 : L.H.S : 0.0576 |   R.H.S : 0.48
14 : L.H.S : 0.067600000000000001 |   R.H.S : 0.52
15 : L.H.S : 0.078400000000000001 |   R.H.S : 0.56
16 : L.H.S : 0.090000000000000002 |   R.H.S : 0.6
17 : L.H.S : 0.102400000000000005 |   R.H.S : 0.64
18 : L.H.S : 0.115600000000000005 |   R.H.S : 0.68
19 : L.H.S : 0.12959999999999999 |   R.H.S : 0.72
20 : L.H.S : 0.144399999999999992 |   R.H.S : 0.76
21 : L.H.S : 0.159999999999999992 |   R.H.S : 0.8
22 : L.H.S : 0.17639999999999995 |   R.H.S : 0.84
23 : L.H.S : 0.19359999999999997 |   R.H.S : 0.88
```

24 : L.H.S : 0.2115999999999995 | R.H.S : 0.92  
 25 : L.H.S : 0.2304 | R.H.S : 0.96  
 26 : L.H.S : 0.25 | R.H.S : 1.0  
 27 : L.H.S : 0.2704000000000003 | R.H.S : 1.04  
 28 : L.H.S : 0.2916 | R.H.S : 1.08  
 29 : L.H.S : 0.3136000000000005 | R.H.S : 1.12  
 30 : L.H.S : 0.3364000000000001 | R.H.S : 1.16  
 31 : L.H.S : 0.3600000000000001 | R.H.S : 1.2  
 32 : L.H.S : 0.38440000000000013 | R.H.S : 1.24  
 33 : L.H.S : 0.4096000000000002 | R.H.S : 1.28  
 34 : L.H.S : 0.4356000000000002 | R.H.S : 1.32  
 35 : L.H.S : 0.4624000000000002 | R.H.S : 1.36  
 36 : L.H.S : 0.49000000000000027 | R.H.S : 1.4000000000000001  
 37 : L.H.S : 0.5184 | R.H.S : 1.44  
 38 : L.H.S : 0.5476 | R.H.S : 1.48  
 39 : L.H.S : 0.5776 | R.H.S : 1.52  
 40 : L.H.S : 0.6084 | R.H.S : 1.56  
 41 : L.H.S : 0.6400000000000001 | R.H.S : 1.6  
 42 : L.H.S : 0.6724000000000001 | R.H.S : 1.6400000000000001  
 43 : L.H.S : 0.7055999999999998 | R.H.S : 1.68  
 44 : L.H.S : 0.7395999999999998 | R.H.S : 1.72  
 45 : L.H.S : 0.7743999999999999 | R.H.S : 1.76  
 46 : L.H.S : 0.8099999999999998 | R.H.S : 1.8  
 47 : L.H.S : 0.8463999999999998 | R.H.S : 1.84  
 48 : L.H.S : 0.8835999999999999 | R.H.S : 1.8800000000000001  
 49 : L.H.S : 0.9216 | R.H.S : 1.92  
 50 : L.H.S : 0.9603999999999999 | R.H.S : 1.96  
 51 : L.H.S : 1.0 | R.H.S : 2.0  
 52 : L.H.S : 1.0404 | R.H.S : 2.04  
 53 : L.H.S : 1.0816000000000001 | R.H.S : 2.08  
 54 : L.H.S : 1.1236000000000002 | R.H.S : 2.12  
 55 : L.H.S : 1.1664 | R.H.S : 2.16  
 56 : L.H.S : 1.2100000000000002 | R.H.S : 2.2  
 57 : L.H.S : 1.2544000000000002 | R.H.S : 2.24  
 58 : L.H.S : 1.2996000000000003 | R.H.S : 2.2800000000000002  
 59 : L.H.S : 1.3456 | R.H.S : 2.32  
 60 : L.H.S : 1.3923999999999999 | R.H.S : 2.36  
 61 : L.H.S : 1.44 | R.H.S : 2.4  
 62 : L.H.S : 1.4884 | R.H.S : 2.44  
 63 : L.H.S : 1.5376 | R.H.S : 2.48  
 64 : L.H.S : 1.5876000000000001 | R.H.S : 2.52  
 65 : L.H.S : 1.6384 | R.H.S : 2.56  
 66 : L.H.S : 1.6900000000000002 | R.H.S : 2.6  
 67 : L.H.S : 1.7424000000000002 | R.H.S : 2.64  
 68 : L.H.S : 1.7956000000000003 | R.H.S : 2.68  
 69 : L.H.S : 1.8496000000000004 | R.H.S : 2.72  
 70 : L.H.S : 1.9044000000000003 | R.H.S : 2.7600000000000002  
 71 : L.H.S : 1.9600000000000004 | R.H.S : 2.8000000000000003  
 72 : L.H.S : 2.0164 | R.H.S : 2.84  
 73 : L.H.S : 2.0736 | R.H.S : 2.88  
 74 : L.H.S : 2.1315999999999997 | R.H.S : 2.92  
 75 : L.H.S : 2.1904 | R.H.S : 2.96  
 76 : L.H.S : 2.25 | R.H.S : 3.0  
 77 : L.H.S : 2.3104 | R.H.S : 3.04  
 78 : L.H.S : 2.3716 | R.H.S : 3.08  
 79 : L.H.S : 2.4336 | R.H.S : 3.12

```

80 : L.H.S : 2.4964000000000004 | R.H.S : 3.16
81 : L.H.S : 2.5600000000000005 | R.H.S : 3.2
82 : L.H.S : 2.6244000000000005 | R.H.S : 3.24
83 : L.H.S : 2.6896000000000004 | R.H.S : 3.2800000000000002
84 : L.H.S : 2.7556000000000003 | R.H.S : 3.3200000000000003
85 : L.H.S : 2.8223999999999996 | R.H.S : 3.36
86 : L.H.S : 2.8899999999999997 | R.H.S : 3.4
87 : L.H.S : 2.9583999999999997 | R.H.S : 3.44
88 : L.H.S : 3.0276 | R.H.S : 3.48
89 : L.H.S : 3.0976 | R.H.S : 3.52
90 : L.H.S : 3.1684 | R.H.S : 3.56
91 : L.H.S : 3.24 | R.H.S : 3.6
92 : L.H.S : 3.3124000000000002 | R.H.S : 3.64
93 : L.H.S : 3.3856 | R.H.S : 3.68
94 : L.H.S : 3.4596000000000005 | R.H.S : 3.72
95 : L.H.S : 3.5344000000000007 | R.H.S : 3.7600000000000002
96 : L.H.S : 3.6100000000000003 | R.H.S : 3.8000000000000003
97 : L.H.S : 3.6864 | R.H.S : 3.84
98 : L.H.S : 3.7636 | R.H.S : 3.88
99 : L.H.S : 3.8415999999999997 | R.H.S : 3.92
100 : L.H.S : 3.9204 | R.H.S : 3.96

```

**The Given Function is : Convex**

## 2. Defining the Convexity for the function $f(x)=(x+1)**2$

```

#Defining both the Functions again for  $f(x) = (x+1)**2$ 
import numpy as np
def f1(a,b,t):
    return (t*a + (1-t)*b + 1)**2

def f2(a,b,t):
    #print("****",t*((a)**2))
    return (t*((a+1)**2)+(1-t)*((b+1)**2))

x = np.arange(0,1,0.01)

flag = 0
count = 0
for t in x:
    count+=1
    print(count,": L.H.S :",f1(1,-1,t)," | R.H.S :",f2(1,-1,t))

    if(f1(1,-1,t)>f2(1,-1,t)):
        flag = 1
        break
print()
if(flag==0):
    print("The Given Function  $f(x) = (x+1)**2$  is : Convex")
else:
    print("The Given Function  $f(x) = (x+1)**2$  is : Non-Convex")

```

**Result :**

```
1 : L.H.S : 0.0 | R.H.S : 0.0
2 : L.H.S : 0.0004000000000000007 | R.H.S : 0.04
3 : L.H.S : 0.0016000000000000003 | R.H.S : 0.08
4 : L.H.S : 0.00360000000000000064 | R.H.S : 0.12
5 : L.H.S : 0.00640000000000000012 | R.H.S : 0.16
6 : L.H.S : 0.01000000000000000018 | R.H.S : 0.2
7 : L.H.S : 0.01440000000000000026 | R.H.S : 0.24
8 : L.H.S : 0.01960000000000000034 | R.H.S : 0.28
9 : L.H.S : 0.0255999999999999973 | R.H.S : 0.32
10 : L.H.S : 0.032399999999999998 | R.H.S : 0.36
11 : L.H.S : 0.039999999999999998 | R.H.S : 0.4
12 : L.H.S : 0.048399999999999999 | R.H.S : 0.44
13 : L.H.S : 0.0576 | R.H.S : 0.48
14 : L.H.S : 0.0676000000000000001 | R.H.S : 0.52
15 : L.H.S : 0.0784000000000000001 | R.H.S : 0.56
16 : L.H.S : 0.0900000000000000002 | R.H.S : 0.6
17 : L.H.S : 0.1024000000000000005 | R.H.S : 0.64
18 : L.H.S : 0.1156000000000000005 | R.H.S : 0.68
19 : L.H.S : 0.129599999999999999 | R.H.S : 0.72
20 : L.H.S : 0.144399999999999992 | R.H.S : 0.76
21 : L.H.S : 0.159999999999999992 | R.H.S : 0.8
22 : L.H.S : 0.176399999999999995 | R.H.S : 0.84
23 : L.H.S : 0.193599999999999997 | R.H.S : 0.88
24 : L.H.S : 0.211599999999999995 | R.H.S : 0.92
25 : L.H.S : 0.2304 | R.H.S : 0.96
26 : L.H.S : 0.25 | R.H.S : 1.0
27 : L.H.S : 0.2704000000000000003 | R.H.S : 1.04
28 : L.H.S : 0.2916 | R.H.S : 1.08
29 : L.H.S : 0.3136000000000000005 | R.H.S : 1.12
30 : L.H.S : 0.3364000000000000001 | R.H.S : 1.16
31 : L.H.S : 0.3600000000000000001 | R.H.S : 1.2
32 : L.H.S : 0.38440000000000000013 | R.H.S : 1.24
33 : L.H.S : 0.4096000000000000002 | R.H.S : 1.28
34 : L.H.S : 0.4356000000000000002 | R.H.S : 1.32
35 : L.H.S : 0.4624000000000000002 | R.H.S : 1.36
36 : L.H.S : 0.49000000000000000027 | R.H.S : 1.400000000000000001
37 : L.H.S : 0.5184 | R.H.S : 1.44
38 : L.H.S : 0.5476 | R.H.S : 1.48
39 : L.H.S : 0.5776 | R.H.S : 1.52
40 : L.H.S : 0.6084 | R.H.S : 1.56
41 : L.H.S : 0.6400000000000000001 | R.H.S : 1.6
42 : L.H.S : 0.6724000000000000001 | R.H.S : 1.640000000000000001
43 : L.H.S : 0.705599999999999998 | R.H.S : 1.68
44 : L.H.S : 0.739599999999999998 | R.H.S : 1.72
45 : L.H.S : 0.774399999999999999 | R.H.S : 1.76
46 : L.H.S : 0.809999999999999998 | R.H.S : 1.8
47 : L.H.S : 0.846399999999999998 | R.H.S : 1.84
48 : L.H.S : 0.883599999999999999 | R.H.S : 1.880000000000000001
49 : L.H.S : 0.9216 | R.H.S : 1.92
50 : L.H.S : 0.960399999999999999 | R.H.S : 1.96
51 : L.H.S : 1.0 | R.H.S : 2.0
52 : L.H.S : 1.0404 | R.H.S : 2.04
53 : L.H.S : 1.0816000000000000001 | R.H.S : 2.08
54 : L.H.S : 1.1236000000000000002 | R.H.S : 2.12
```



```

55 : L.H.S : 1.1664 | R.H.S : 2.16
56 : L.H.S : 1.2100000000000002 | R.H.S : 2.2
57 : L.H.S : 1.2544000000000002 | R.H.S : 2.24
58 : L.H.S : 1.2996000000000003 | R.H.S : 2.2800000000000002
59 : L.H.S : 1.3456 | R.H.S : 2.32
60 : L.H.S : 1.3923999999999999 | R.H.S : 2.36
61 : L.H.S : 1.44 | R.H.S : 2.4
62 : L.H.S : 1.4884 | R.H.S : 2.44
63 : L.H.S : 1.5376 | R.H.S : 2.48
64 : L.H.S : 1.5876000000000001 | R.H.S : 2.52
65 : L.H.S : 1.6384 | R.H.S : 2.56
66 : L.H.S : 1.6900000000000002 | R.H.S : 2.6
67 : L.H.S : 1.7424000000000002 | R.H.S : 2.64
68 : L.H.S : 1.7956000000000003 | R.H.S : 2.68
69 : L.H.S : 1.8496000000000004 | R.H.S : 2.72
70 : L.H.S : 1.9044000000000003 | R.H.S : 2.7600000000000002
71 : L.H.S : 1.9600000000000004 | R.H.S : 2.8000000000000003
72 : L.H.S : 2.0164 | R.H.S : 2.84
73 : L.H.S : 2.0736 | R.H.S : 2.88
74 : L.H.S : 2.1315999999999997 | R.H.S : 2.92
75 : L.H.S : 2.1904 | R.H.S : 2.96
76 : L.H.S : 2.25 | R.H.S : 3.0
77 : L.H.S : 2.3104 | R.H.S : 3.04
78 : L.H.S : 2.3716 | R.H.S : 3.08
79 : L.H.S : 2.4336 | R.H.S : 3.12
80 : L.H.S : 2.4964000000000004 | R.H.S : 3.16
81 : L.H.S : 2.5600000000000005 | R.H.S : 3.2
82 : L.H.S : 2.6244000000000005 | R.H.S : 3.24
83 : L.H.S : 2.6896000000000004 | R.H.S : 3.2800000000000002
84 : L.H.S : 2.7556000000000003 | R.H.S : 3.3200000000000003
85 : L.H.S : 2.8223999999999996 | R.H.S : 3.36
86 : L.H.S : 2.8899999999999997 | R.H.S : 3.4
87 : L.H.S : 2.9583999999999997 | R.H.S : 3.44
88 : L.H.S : 3.0276 | R.H.S : 3.48
89 : L.H.S : 3.0976 | R.H.S : 3.52
90 : L.H.S : 3.1684 | R.H.S : 3.56
91 : L.H.S : 3.24 | R.H.S : 3.6
92 : L.H.S : 3.3124000000000002 | R.H.S : 3.64
93 : L.H.S : 3.3856 | R.H.S : 3.68
94 : L.H.S : 3.4596000000000005 | R.H.S : 3.72
95 : L.H.S : 3.5344000000000007 | R.H.S : 3.7600000000000002
96 : L.H.S : 3.6100000000000003 | R.H.S : 3.8000000000000003
97 : L.H.S : 3.6864 | R.H.S : 3.84
98 : L.H.S : 3.7636 | R.H.S : 3.88
99 : L.H.S : 3.8415999999999997 | R.H.S : 3.92
100 : L.H.S : 3.9204 | R.H.S : 3.96

```

**The Given Function is : Convex**

**3. Obtain the 10 target (output) datapoints for the following equation for uniformly spaced input variable in the range [0,10].**

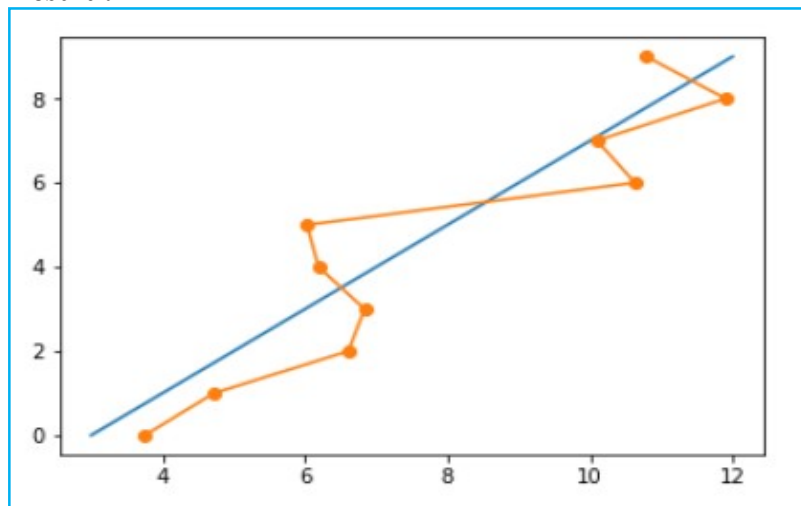
**Solution :**

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

y = [x+3 for x in range(0,10)]
x1 = []
x1 = [x for x in range(0,10)]
y2 = []
plt.plot(y,x1)
noise = np.random.normal(0,1,10)
for i in range(len(x1)):
    y2.append(y[i] + (noise[i]*((-1)**(i))))

plt.plot(y2,x1,'o-')
#print(y2)
```

**Result :**



**Fig.**  $y = x + 3$  line in Blue and non linear structure having Orange tone.

Now we are doing the Gradient Descent to find the actual Curve or line and for that we are taking the hypothesis as  $y = \theta_0 + \theta_1 x$ ;

**Solution :**

```
theta = [0,0]
m = len(x1)
alpha = 0.01

for it in range(1,1000):
    cm = [0,0]
    for i in range(m):

        xi = [1]
```

```
xi.append(x1[i])

c = 0
diff = 0

for j in range(len(xi)):
    c = c + xi[j]*theta[j]

#print("c:",c)

diff = (c - y[i])

for j in range(len(xi)):
    cm[j] += diff*xi[j]

for j in range(len(xi)):
    cm[j] += diff*xi[j]

for i in range(len(theta)):
    theta[i] = theta[i] - (1/m)*(cm[i])*alpha

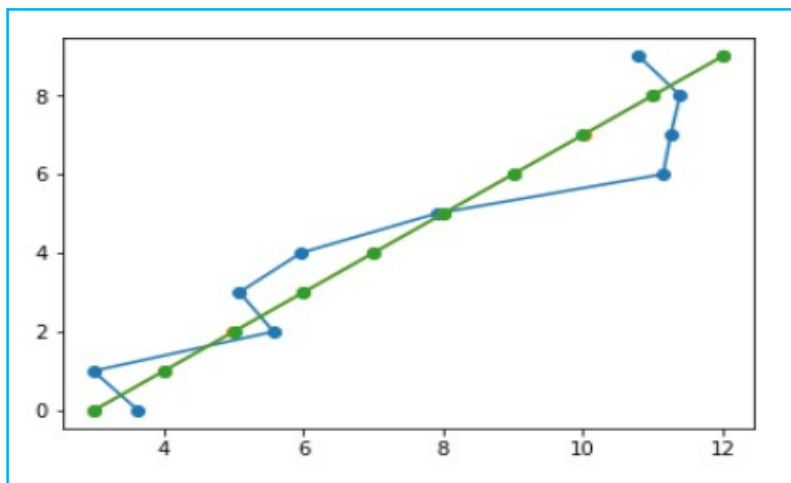
print("Predicted Theta:",theta)
plt.plot(y2,x1,'o-')

y3 = []

for i in range(len(x1)):
    y3.append(theta[1]*x1[i]+theta[0])

plt.plot(y3,x1,'-o')
plt.plot(y,x1,'-o')
```

**Result :**



**Fig. The obtained line and Line with Noise and the actual line.**

#### 4. Finding the Thetas for the above problem using Psuedo-Inverse Method.

**Solution :**

#Finidng theta Using Normalization Method.

```
from numpy.linalg import inv
```

```
#print(x1)
```

```
X = []
```

```
for i in range(len(x1)):
```

```
    c =[1]
```

```
    c.append(x1[i])
```

```
    X.append(c)
```

```
Z = np.dot(np.transpose(X),X)
```

```
#print(Z)
```

```
Z1 = np.dot(np.transpose(X),y)
```

```
theta_n = np.dot(inv(Z),Z1)
```

```
print("Predicted Theta(Using Normalization Method) :",list(theta_n))
```

**Result :**

**Predicted Theta(Using Normalization Method) : [3.0, 1.0]**