Assignment 2

Megh Manoj Bhalerao 16EE234

0.1 Question 1

0.1.1 Here we plot the randomly distributed samples across the two regions

0.1.2 The first region is defined and the samples are plotted and after that the second region is just the extension of the first region by scaling and shifting. The regions are plotted below for different shift values along the y axis

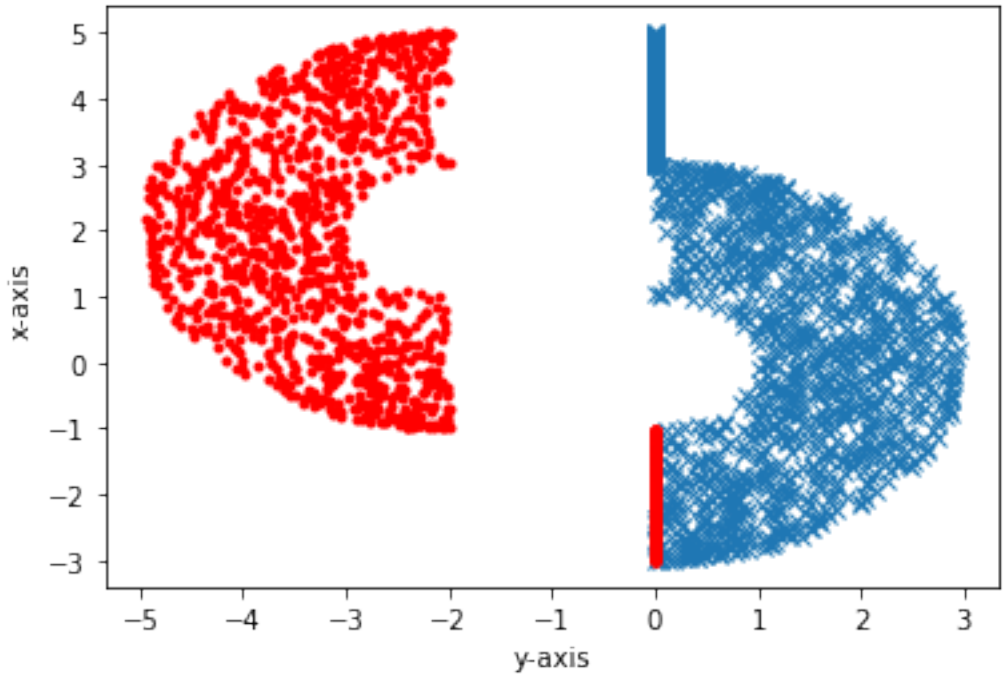
[103]:
```
import numpy as np
from random import random
from math import sqrt
import matplotlib.pyplot as plt
from pylab import *
from scipy.ndimage.interpolation import shift
```

[89]:
```
def plot_dis(r,w,d):
    x = np.linspace(-r-w/2,2*r + w/2,1500)
    # Setting the values of yA
    spacing = (3*r + w)/1500
    print(spacing)
    yA = np.zeros(1500)
    yB = np.zeros(1500)
    index = 0
    for i in x[0:int(w/spacing)]:
        yA[index] = 0 + random()*(sqrt((r+w/2)**2 - i**2))
        index = index+1
    for i in x[int(w/spacing):int(2*r/spacing)]:
        yA[index] = sqrt((r-w/2)**2 - i**2) + random()*(sqrt((r+w/2)**2 - i**2)␣
 ↪- sqrt((r-w/2)**2 - i**2))
        index = index+1
    for i in x[int(2*r/spacing):int(2*(r+w/2)/spacing)]:
        yA[index] = 0 + random()*(sqrt((r+w/2)**2 - i**2))
        index = index+1
    return x, yA
w = 2
r = 2
d = 2
spacing = (3*r + w)/1500
x,yA = plot_dis(w,r,d)
plt.plot(yA,x,'x')
plt.xlabel('y-axis')
yB =  -d -yA
yB = shift(yB,(r)/spacing)
plt.plot(yB,x,'r.')
```

```
plt.ylabel('x-axis')
```

```
0.005333333333333333
```

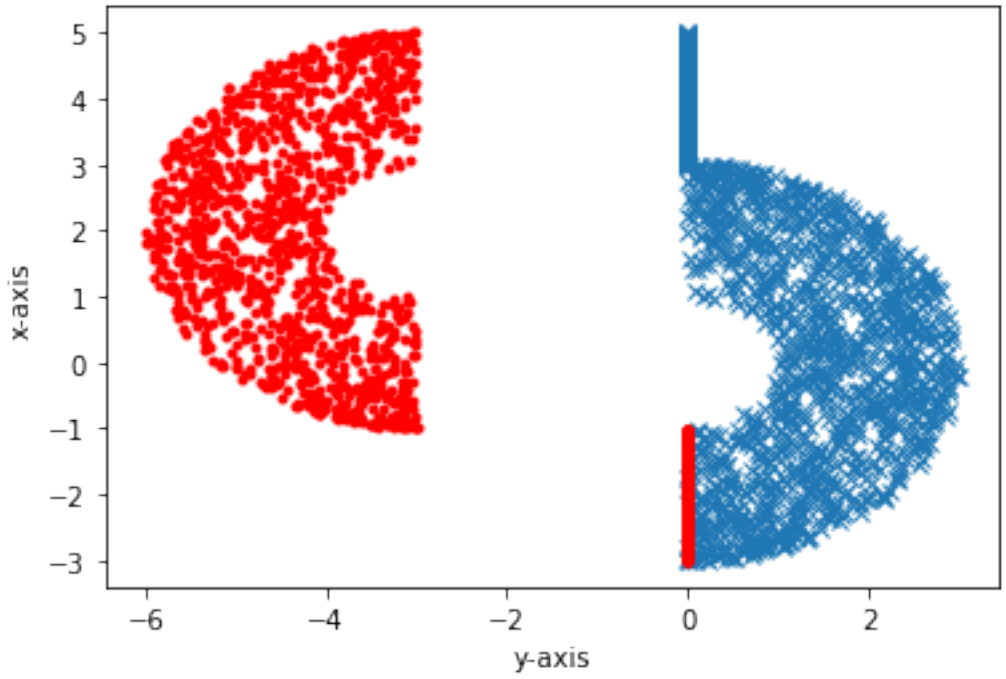
[89]:
```
Text(0, 0.5, 'x-axis')
```



[90]:
```
w = 2
r = 2
d = 3
spacing = (3*r + w)/1500
x,yA = plot_dis(w,r,d)
plt.plot(yA,x,'x')
yB =  -d -yA
yB = shift(yB,(r)/spacing)
plt.plot(yB,x,'r.')
plt.ylabel('x-axis')
plt.xlabel('y-axis')
```

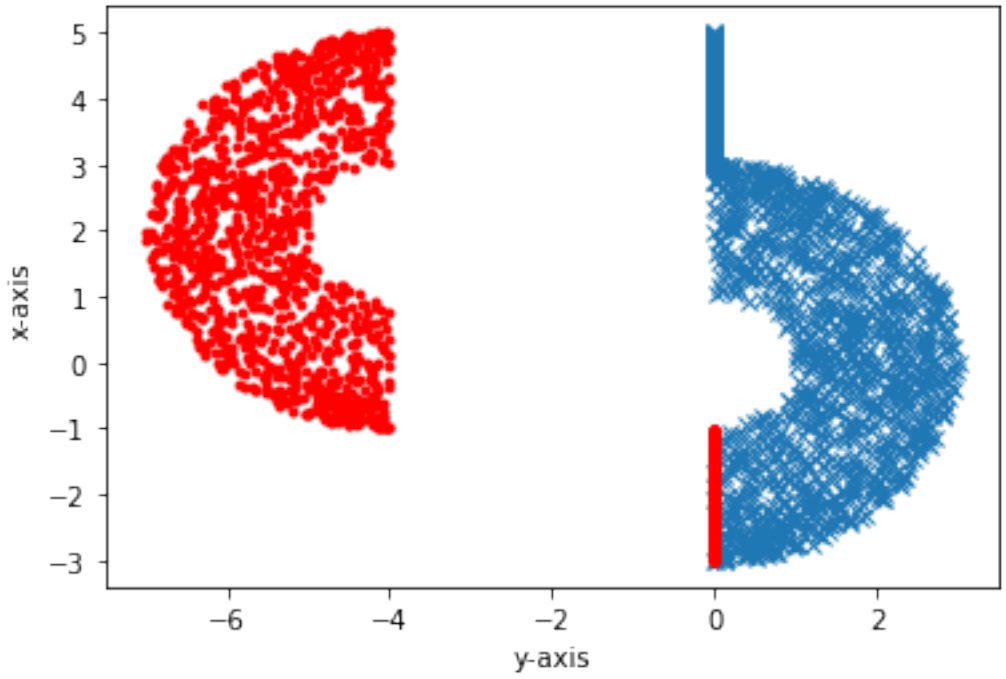
```
0.005333333333333333
```

[90]:
```
Text(0.5, 0, 'y-axis')
```



[95]:
```
w = 2
r = 2
d = 4
spacing = (3*r + w)/1500
x,yA = plot_dis(w,r,d)
plt.plot(yA,x,'x', label = "Region A")
yB =  -d -yA
yB = shift(yB,(r)/spacing)
plt.plot(yB,x,'r.', label = "Region B")
plt.ylabel('x-axis')
plt.xlabel('y-axis')
plt.show()
```

```
0.005333333333333333
```



0.2 Question 2

0.2.1 Here we plot the output of the activation function for different values of lambda (The learning rate) and g (the simple weight matrix times the input)

0.2.2 After that we do the contour plot of the curve which is obtaibed from taking different values of w0, w1, w2 and lambda on the plane made by x1 and x2

Imports needed for different functionalities

[97]:
```
from mpl_toolkits import mplot3d
import matplotlib.pyplot as plt
from matplotlib import *
from matplotlib.pyplot import *
import numpy as np
```

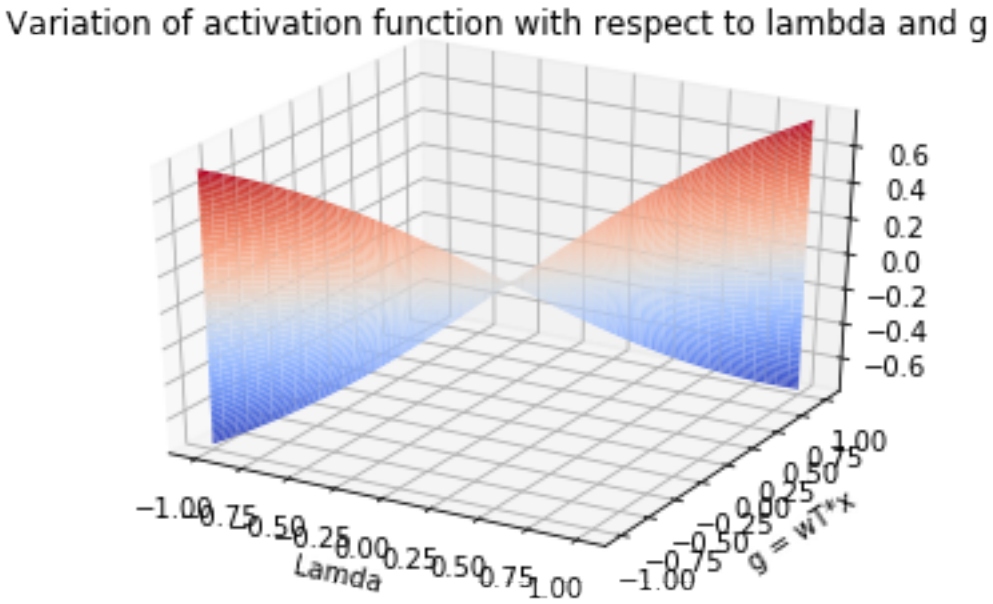
Plot of lambda vs g

[98]:
```
# Defining the activtion function that is to be used
def act(x):
    return np.tanh(x)
```

[99]:
```
#Defining the values of lambda and g for the plot The
lam = np.linspace(-1,1,1000)
g = np.linspace(-1,1,1000)
L,G = np.meshgrid(lam,g)
f = act(L*G)
# Defineing how to plot the function
```

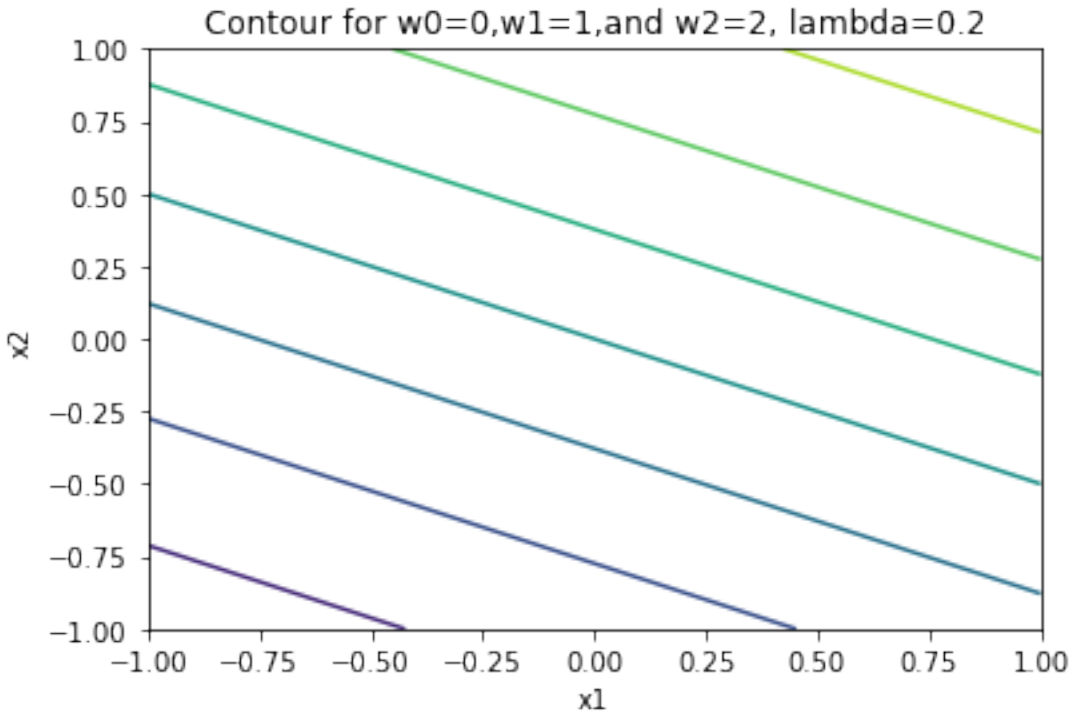
```
fig1 = plt.figure()
ax = fig1.gca(projection='3d')
ax.plot_surface(lam, g, f, cmap = cm.coolwarm)
title('Variation of activation function with respect to lambda and g')
xlabel('Lamda')
ylabel('g = wT*x')
```

[99]:
```
Text(0.5, 0, 'g = wT*x')
```



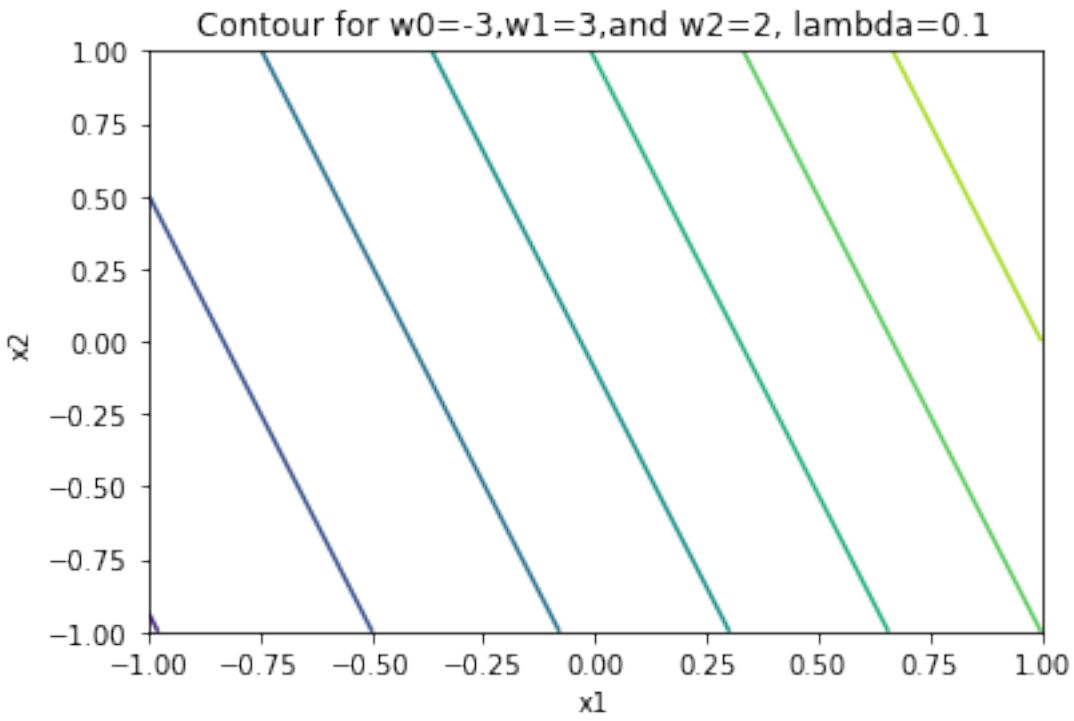
[100]:
```
# Plotting the contour plot for different values of the function f on the x1 and␣
 ↪x2 plane
x1 = np.linspace(-1,1,1000)
x2 = np.linspace(-1,1,1000)
X1, X2 = np.meshgrid(x1,x2)
# Now we take some example values of lambda, w0, w1, w2 and plot the contours
w0 = 0
w1 = 1
w2 = 2
lam_0 = 0.2
g = w0 + X1*w1 + w2*X2
f = act(lam_0*g)
fig1 = plt.figure()
ax = fig1.gca()
ax.contour(X1, X2, f)
title('Contour for w0='+str(w0)+',w1='+str(w1)+',and w2='+str(2)+', lambda=' +␣
 ↪str(lam_0))
xlabel('x1')
ylabel('x2')
```

[100]:
```
Text(0, 0.5, 'x2')
```



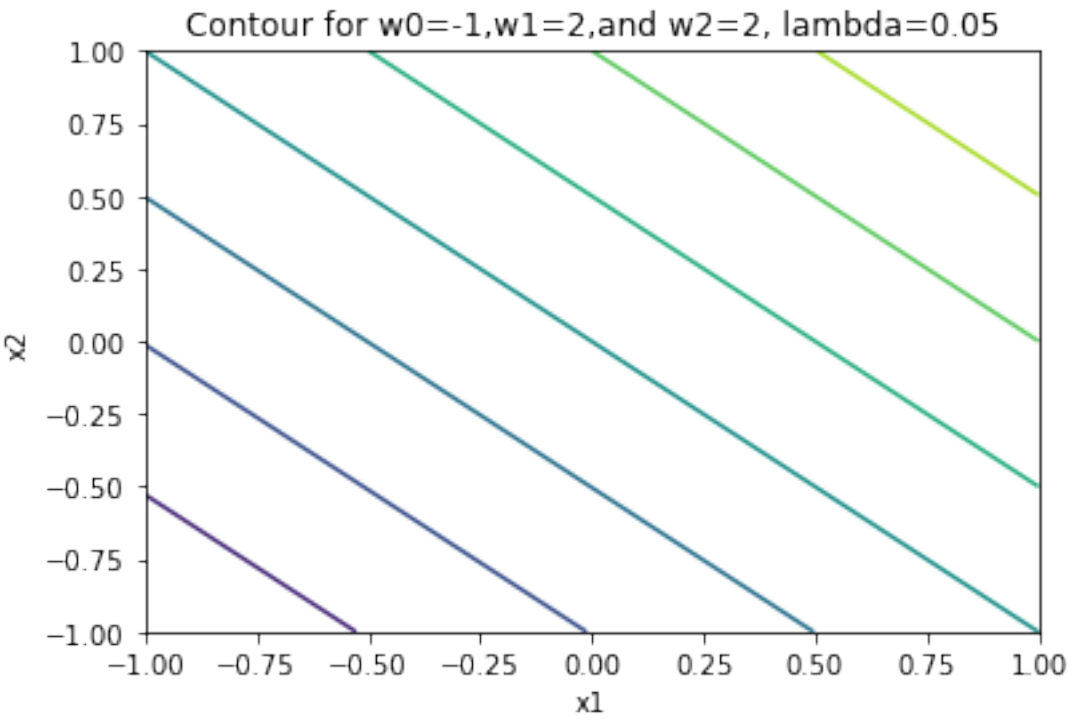
[101]:
```
w0 = -3
w1 = 3
w2 = 1
lam_0 = 0.1
g = w0 + X1*w1 + w2*X2
f = act(lam_0*g)
fig1 = plt.figure()
ax = fig1.gca()
ax.contour(X1, X2, f)
title('Contour for w0='+str(w0)+',w1='+str(w1)+',and w2='+str(2)+', lambda=' +␣
 ↪str(lam_0))
xlabel('x1')
ylabel('x2')
```

[101]:
```
Text(0, 0.5, 'x2')
```



[102]:
```
w0 = -1
w1 = 2
w2 = 2
lam_0 = 0.05
g = w0 + X1*w1 + w2*X2
f = act(lam_0*g)
fig1 = plt.figure()
ax = fig1.gca()
ax.contour(X1, X2, f)
title('Contour for w0='+str(w0)+',w1='+str(w1)+',and w2='+str(2)+', lambda=' +␣
 ↪str(lam_0))
xlabel('x1')
ylabel('x2')
```

[102]:
```
Text(0, 0.5, 'x2')
```


Contour for w0=-1,w1=2,and w2=2, lambda=0.05
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
x2
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
x1