```
 1  #Alon Shmilovich, id 034616359, alonsh, Jerusalem College of Engineering JCE
 2
 3  import matplotlib.pyplot as plt
 4  import numpy as np
 5  import copy
 6
 7  #Initiailize:
 8  x = np.array([[1,0,0],[1,0,1],[1,1,0],[1,1,1]])
 9  z = np.array([[1],[1],[1],[0]])
10
11  weights = [0,0,0] #Here the right weights will be held
12  prev_weights=[0,0,0] #For comparisons
13
14  i=0 #For iterations on matrix x
15  num=0 #Bounds
16  n=0 #Network for output
17  counter=0 #Counter for comparisons
18  threshold = 0.5 #This threshold will help us decide n
19  learning_rate = 0.1 #Learning correction
20  c=np.array([0,0,0]) #Results for x * weights
21
22  while num<100: #Set an upper bound
23      if i>=4: #4 iterations for matrix x
24          i=0
25
26      c = x[i, 0:3] * weights
27
28      sum=np.sum(c)
29
30      if (sum > threshold):
31          n=1
32      else:
33          n=0
34
35      error = int(z[i] - n)
36
37      correction = learning_rate * error
38
39      prev_weights = copy.copy(weights)
40
41      weights += x[i,0:3] * correction
42
43      if (prev_weights == weights).all():
44          counter+=1
45          if counter==3:
46              break
47      else:
48          counter = 0
49
50      print "Weights # ",num, weights
51
```

```
52      i+=1
53      num+=1
54
55      y1 = (-weights[1]*3 - weights[0])/(weights[2] - threshold)
56      print "First point: (3,",y1,")"
57
58      y2 = (-weights[1]*(-2) - weights[0])/(weights[2] - threshold)
59      print "First point: (-2,",y2,")"
60
61      X=[3,-2]
62      Y=[y1,y2]
63
64      plt.title('NAND Perceptron')
65      plt.xlabel('X1')
66      plt.ylabel('X2')
67      plt.plot([0, 0, 1, 1], [0, 1, 0, 1], 'ro', ms=10)
68      plt.axis([-1, 2, -1, 2])
69      plt.plot([1], [1], 'bo', ms=10)
70      plt.plot(X, Y)
71      plt.show()
72
```