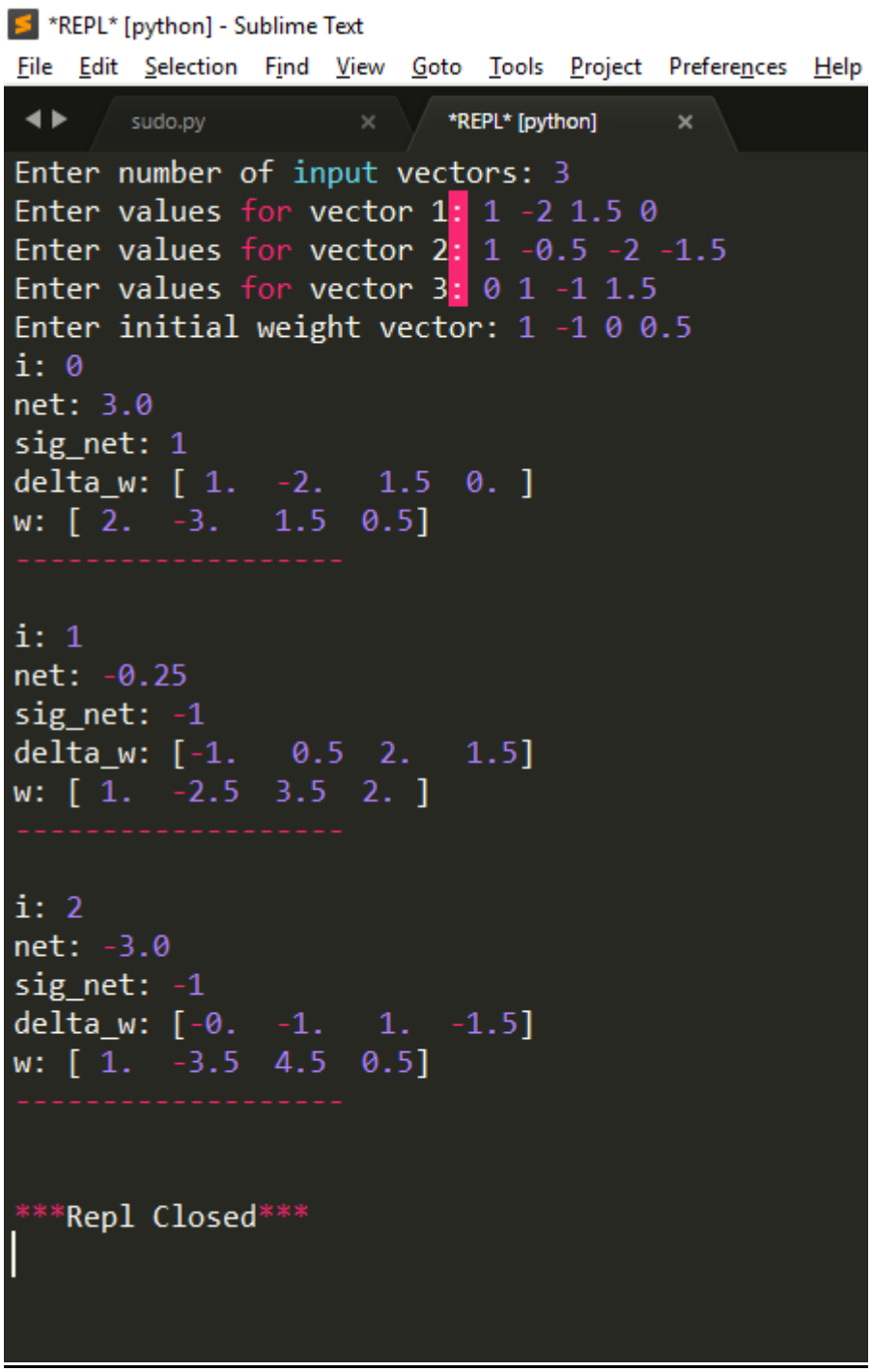
```
# -*- coding: utf-8 -*-
"""
@author: chirag
"""

import numpy as np
import time

def threshold(x):
	if x >=0:
		return 1
	else:
		return -1

def print_func(loop_var, net, sig_net, w, delta_w):
		print("i: "+ str(loop_var))
		print("net: "+ str(net))
		print("sig_net: "+ str(sig_net))
		print("delta_w: "+ str(delta_w))
		print("w: "+ str(w))
		print("-------------------\n")

def compute():
	try:
		n = int(input("Enter number of input vectors: "))
		x = []
		r = 1 #Learning constant(c)
		for i in range(0,n):
			raw_str1 = str(input("Enter values for vector " + str(i+1) + ": "))
			input_vector = raw_str1.split(' ')
			#print(input_vector)
			ip_list = []
			for ele in input_vector:
				ip_list.append(float(ele))
```

```
                #print(ip_list)
                np_list = np.array(ip_list, dtype=np.float64)
                x.append(np_list)
        raw_str3 = str(input("Enter initial weight vector: "))
        w = raw_str3.split(' ')
        w_list = []
        for ele in w:
                w_list.append(float(ele))
        #np_wlist = np.array(w_list, dtype=np.float64)
        #print(np_wlist)
        #if len(np_wlist) != n:
        #       print("Init Weight Vector Error..")
        delta_w = 0
        for i in range(0,n):
                net = np.transpose(np.asarray(w_list)).dot(np.asarray(x[i]))
                #print(net)
                sig_net = threshold(net)
                #print(sig_net)
                delta_w = r * sig_net * x[i]
                #print(delta_w)
                w_list = np.add(np.asarray(w_list),delta_w)
                print_func(i, net, sig_net, w_list, delta_w)
    except Exception as e:
        print("Error.. "+(str(e)))

if __name__ == '__main__':
    compute()
```

Output:


REPL [python] - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
sudo.py
REPL [python]
Enter number of input vectors: 3
Enter values for vector 1: 1 -2 1.5 0
Enter values for vector 2: 1 -0.5 -2 -1.5
Enter values for vector 3: 0 1 -1 1.5
Enter initial weight vector: 1 -1 0 0.5
i: 0
net: 3.0
sig_net: 1
delta_w: [1. -2. 1.5 0.]
w: [2. -3. 1.5 0.5]
i: 1
net: -0.25
sig_net: -1
delta_w: [-1. 0.5 2. 1.5]
w: [1. -2.5 3.5 2.]
i: 2
net: -3.0
sig_net: -1
delta_w: [-0. -1. 1. -1.5]
w: [1. -3.5 4.5 0.5]
Repl Closed