import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import pandas as pd

def step\_function(x):

 if x<0:

 return -1

 else:

 return 1

#training\_set = [((-1,-1), 0), ((0.5,0.5), 0),((1,0.25), 0),((0,-1), 0)]#, ((0,1), 1), ((1,0), 1), ((1,1),1)]

training\_set = [((-.1,-.1),-1), ((-.1,.3),-1), ((0.2,0),-1), ((0,1), 1), ((1,0), 1), ((1,1),1)]

num\_of\_samples = 6;

plt.figure(0)

x1 = [training\_set[i][0][0] for i in range(num\_of\_samples)]

x2 = [training\_set[i][0][1] for i in range(num\_of\_samples)]

y = [training\_set[i][1] for i in range(num\_of\_samples)]

print("x1 =",x1)

print("x2 =",x2)

print("y =",y)

df = pd.DataFrame({'x1':x1, 'x2':x2, 'y':y})

sns.lmplot("x1","x2", data=df, hue='y', fit\_reg=False, markers=["o", "s"])

w = np.random.rand(2)

errors = []

alpha = .5

epoch = 30

b = 0

for i in range(epoch):

 for x, y in training\_set:

 u = sum(x\*w) + b

 print("x=",x)

 print("w=",w)

 print("u=",u)

 y\_hat = step\_function(u)

 error = y - y\_hat

 errors.append(error)

 for index, value in enumerate(x):

 print("index=",index)

 print("vlaue=",value)

 w[index] += alpha \* error \* value

 b += alpha \* error

a = [0,-b/w[1]]

c = [-b/w[0],0]

plt.plot(a,c)

plt.figure(2)

plt.ylim([-2,2])

plt.plot(errors)