```python
import numpy as np
import random
import copy

def logreg(N):
    p1x = random.uniform(-1,1) #pick 2 random points and find the equation of line going through them
    p1y = random.uniform(-1,1)
    p2x = random.uniform(-1,1)
    p2y = random.uniform(-1,1)
    f = np.array([(-(p2x-p1x)+p1x*(p2y-p1y)),(p1y-p2y),(p2x-p1x)]) #correct f in terms of (w0, wx, wy)

    points = 2*np.random.random_sample(2*N)-1 #generate N in-sample points
    y = np.zeros([N])
    w = np.array([0.0,0.0,0.0])
    w1 = np.array([0.0,0.0,0.0])

    for i in range(N): #classify each point
        y[i] = np.sign(np.dot(f,np.array([1,points[2*i],points[2*i+1]])))

    first = True
    epoch = 0
    while first or np.linalg.norm(w-w1) >= 0.01:
        first = False
        order = np.random.permutation(N) #randomize order of points in epoch
        w1 = copy.copy(w)
        for i in range(len(order)):
            x = np.array([1,points[2*order[i]],points[2*order[i]+1]])
            yi = y[order[i]]
            dive = yi*x/(1+np.exp(yi*np.dot(w,x))) #calculate direction of minimum error
            w += 0.01*dive #descend in direction of min error
        epoch += 1

    sample = 2*np.random.random_sample(10000)-1 #generate 5000 sample points
    error = 0
    for i in range(5000):
        xs = np.array([1,sample[2*i],sample[2*i+1]])
        ys = np.dot(f,xs)
        error += np.log(1+np.exp(-ys*np.dot(w,xs))) #calculate cross entropy error for 5000 points

    return [error/5000,epoch]

avgeot = 0
avgepoch = 0
for i in range(100):
    results = logreg(100)
    avgeot += results[0]
    avgepoch += results[1]

print(avgeot/100)
print(avgepoch/100)
```