In [1]:
```
from random import *
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
b = []
w0 = uniform(-0.25, 0.25)
w1 = uniform(-1, 1)
w2 = uniform(-1, 1)
b.append([w0,w1,w2])
wopt = np.asarray(b)
print(wopt)
```

```
[[ 0.07398882 -0.15041019  0.28511816]]
```

In [3]:
```
# n = 100
```

In [4]:
```
a = []
for i in range(0,100):
    x1 = uniform(-1, 1)
    x2 = uniform(-1, 1)
    a.append([1, x1, x2])
    i += 1
S = np.asarray(a)
#print(S)
```

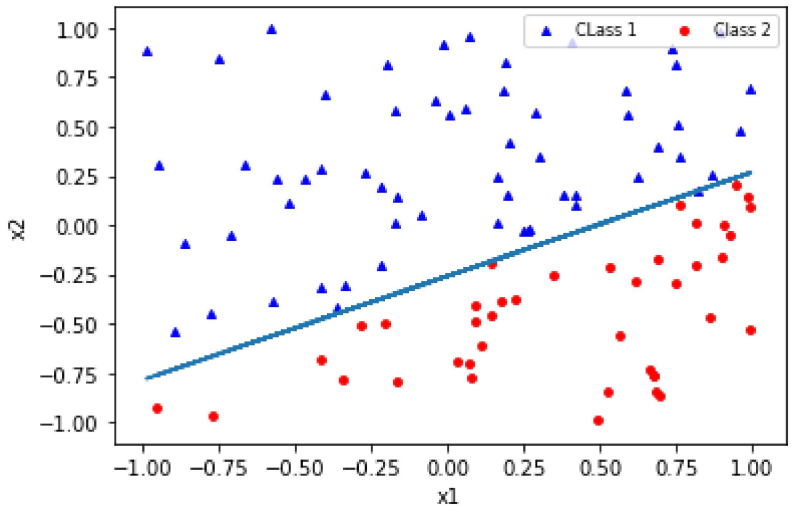
In [5]:
```
wTopt = np.transpose(wopt)
#print(wopt[0][1])
```

In [6]:
```
S1 = []
S2 = []
for i in range(0,100):
    if np.matmul(S[i], wTopt) >= 0:
        S1.append(S[i])
    else:
        S2.append(S[i])
```

In [7]:
```
print(len(S1))
print(len(S2))
```

```
59
41
```

In [8]:
```
fig, ax = plt.subplots()
xs = [x[1] for x in S1]
ys = [y[2] for y in S1]

# produce a legend with the unique colors from the scatter
scatter1 = plt.scatter(xs, ys, color ='blue', marker  = '^', s = 16)




xs = [x[1] for x in S2]
ys = [y[2] for y in S2]
scatter2 = plt.scatter(xs, ys, color ='red', s = 15)

plt.legend((scatter1, scatter2),
           ('CLass 1', 'Class 2'),
           scatterpoints=1,
           loc='upper right',
           ncol=3,
           fontsize=8)

xs = [x[1] for x in S]
ys = [y[2] for y in S]
#print(len(xs))
y = []
for i in range(len(S)):
    y.append(-((wopt[0][0]+(wopt[0][1]*xs[i]))/wopt[0][2]))




plt.plot(xs, y)

plt.xlabel('x1')
plt.ylabel('x2')
plt.show()
```



In [9]:
```
# PTA
```

In [10]:
```
b = []
wa = uniform(-1, 1)
wb = uniform(-1, 1)
wc = uniform(-1, 1)
b. append([wa,wb,wc])
w = np.asarray(b)
t = w
print(w)
```

```
[[ 0.07611122  0.26930747 -0.57670416]]
```

In [11]:
```
# Epoch 0
m = []
e = []
wT = np.transpose(w)
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1
```

In [12]:
```
# Learing Rate 1
n = 1

# Epoch 1
count = 0
epoch = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
        w =  w + n*S1[i]

    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
        w =  w - n*S2[i]
    i += 1
wT= np.transpose(w)
e.append(epoch)
m.append(count)
m
```

Out[12]:
```
[77]
```

In [13]:
```
while count != 0:
    epoch += 1
    count = 0
    for i in range(len(S1)):
        if np.matmul(S1[i], wT) < 0:
            w =  w + n*S1[i]
            count += 1
        i += 1
    for i in range(len(S2)):
        if np.matmul(S2[i], wT) >= 0:
            w =  w - n*S2[i]
            count += 1
        i += 1
    m.append(count)
    e.append(epoch)
    wT = np.transpose(w)
print("Final weights for learing rate 1 " + str(w))
```

```
Final weights for learing rate 1 [[ 11.07611122 -22.19536636  42.95259398]]
```

In [14]:
```
# Testing
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
print(count)
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1

count
```

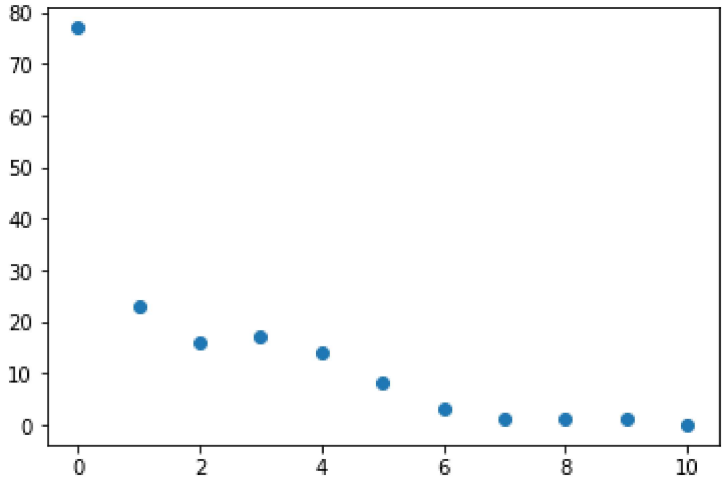
```
0
```

Out[14]:
```
0
```

In [15]:
```
print(e)
print(m)
plt.scatter(e,m)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[77, 23, 16, 17, 14, 8, 3, 1, 1, 1, 0]
```

Out[15]: <matplotlib.collections.PathCollection at 0x196a7c3d0f0>



In [16]:
```
# Learning Rate 10
n = 10
w = t
wT = np.transpose(w)
e= []
m =[]
# Epoch 1
count = 0
epoch = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
        w =  w + n*S1[i]

    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
        w =  w - n*S2[i]
    i += 1
wT= np.transpose(w)
e.append(epoch)
m.append(count)
```

In [17]:
```
while count != 0:
    epoch += 1
    count = 0
    for i in range(len(S1)):
        if np.matmul(S1[i], wT) < 0:
            w =  w + n*S1[i]
            count += 1
        i += 1
    for i in range(len(S2)):
        if np.matmul(S2[i], wT) >= 0:
            w =  w - n*S2[i]
            count += 1
        i += 1
    m.append(count)
    e.append(epoch)
    wT = np.transpose(w)
print("Final weights for learing rate 10 " + str(w))
```

```
Final weights for learing rate 10 [[ 110.07611122 -221.60765386  432.9090093
3]]
```

In [18]:
```
# Testing
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
print(count)
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1

count
```

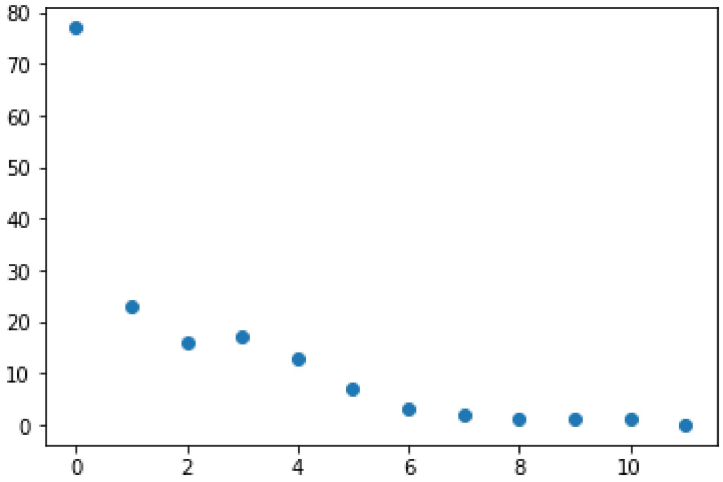
```
0
```

Out[18]:
```
0
```

In [19]:
```
print(e)
print(m)
plt.scatter(e,m)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
[77, 23, 16, 17, 13, 7, 3, 2, 1, 1, 1, 0]
```

Out[19]: <matplotlib.collections.PathCollection at 0x196a7ca9080>



In [20]:
```
# Learning Rate 0.1
n = 0.1
w = t
wT = np.transpose(w)
e= []
m =[]
# Epoch 1
count = 0
epoch = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
        w =  w + n*S1[i]

    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
        w =  w - n*S2[i]
    i += 1
wT= np.transpose(w)
e.append(epoch)
m.append(count)
```

In [21]:
```
while count != 0:
    epoch += 1
    count = 0
    for i in range(len(S1)):
        if np.matmul(S1[i], wT) < 0:
            w =  w + n*S1[i]
            count += 1
        i += 1
    for i in range(len(S2)):
        if np.matmul(S2[i], wT) >= 0:
            w =  w - n*S2[i]
            count += 1
        i += 1
    m.append(count)
    e.append(epoch)
    wT = np.transpose(w)
print("Final weights for learing rate 0.1 " + str(w))
```

```
Final weights for learing rate 0.1 [[ 1.17611122 -2.33942677  4.49016557]]
```

In [22]:
```
# Testing
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
print(count)
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1

count
```

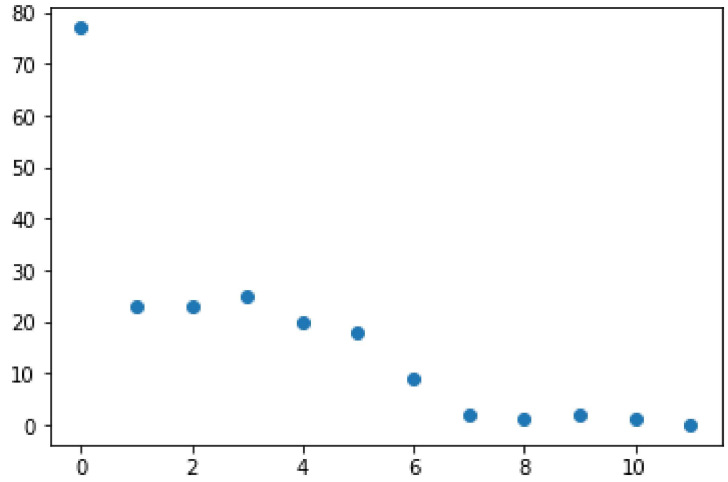
```
0
```

Out[22]:
```
0
```

In [23]:
```
print(e)
print(m)
plt.scatter(e,m)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]
[77, 23, 23, 25, 20, 18, 9, 2, 1, 2, 1, 0]
```

Out[23]: <matplotlib.collections.PathCollection at 0x196a7d11e80>



In [ ]:

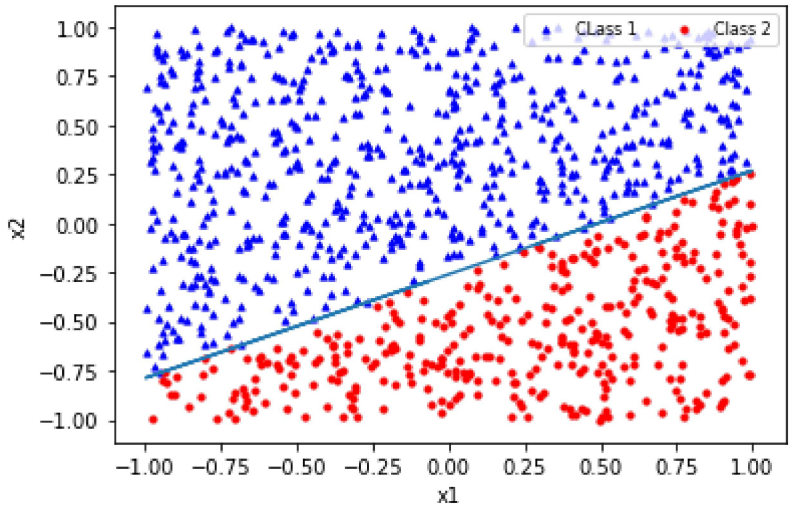
In [24]:
```
# n = 1000

a = []
for i in range(0,1000):
    x1 = uniform(-1, 1)
    x2 = uniform(-1, 1)
    a.append([1, x1, x2])
    i += 1
S = np.asarray(a)
#print(S)
```

In [25]:
```
wTopt = np.transpose(wopt)
```

In [26]:
```
S1 = []
S2 = []
for i in range(0,1000):
    if np.matmul(S[i], wTopt) >= 0:
        S1.append(S[i])
    else:
        S2.append(S[i])
print(len(S1))
print(len(S2))
```

```
638
362
```

In [27]:
```
fig, ax = plt.subplots()
xs = [x[1] for x in S1]
ys = [y[2] for y in S1]

# produce a legend with the unique colors from the scatter
scatter1 = plt.scatter(xs, ys, color ='blue', marker  = '^', s = 10)




xs = [x[1] for x in S2]
ys = [y[2] for y in S2]
scatter2 = plt.scatter(xs, ys, color ='red', s = 10)

plt.legend((scatter1, scatter2),
           ('CLass 1', 'Class 2'),
           scatterpoints=1,
           loc='upper right',
           ncol=3,
           fontsize=8)

xs = [x[1] for x in S]
ys = [y[2] for y in S]
#print(len(xs))
y = []
for i in range(len(S)):
    y.append(-((wopt[0][0]+(wopt[0][1]*xs[i]))/wopt[0][2]))




plt.plot(xs, y)

plt.xlabel('x1')
plt.ylabel('x2')
plt.show()
```



In [28]:
```
# Epoch 0
m = []
e = []
w = t
wT = np.transpose(w)
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1
```

In [29]:
```
# Learing Rate 1
n = 1

# Epoch 1
count = 0
epoch = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
        w =  w + n*S1[i]

    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
        w =  w - n*S2[i]
    i += 1
wT= np.transpose(w)
e.append(epoch)
m.append(count)
```

In [30]:
```
while count != 0:
    epoch += 1
    count = 0
    for i in range(len(S1)):
        if np.matmul(S1[i], wT) < 0:
            w =  w + n*S1[i]
            count += 1
        i += 1
    for i in range(len(S2)):
        if np.matmul(S2[i], wT) >= 0:
            w =  w - n*S2[i]
            count += 1
        i += 1
    m.append(count)
    e.append(epoch)
    wT = np.transpose(w)
print("Final weights for learing rate 1 " + str(w))
```

```
Final weights for learing rate 1 [[ 105.07611122 -215.03790194  409.4783406
]]
```

In [31]:
```
# Testing
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
print(count)
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1

count
```

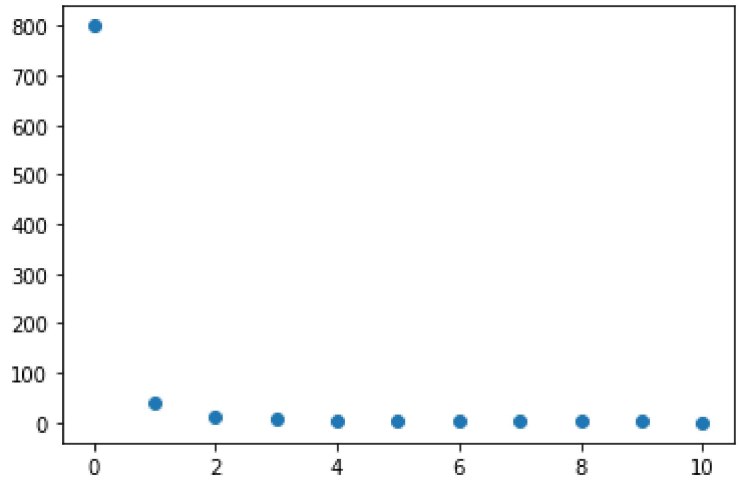
```
0
```

Out[31]:
```
0
```

In [32]:
```
print(e)
print(m)
plt.scatter(e,m)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[800, 41, 13, 7, 4, 2, 2, 2, 2, 2, 0]
```

Out[32]: <matplotlib.collections.PathCollection at 0x196a7e3f0f0>



In [33]:
```
# Learning Rate 10
n = 10
w = t
wT = np.transpose(w)
e= []
m =[]
# Epoch 1
count = 0
epoch = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
        w =  w + n*S1[i]

    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
        w =  w - n*S2[i]
    i += 1
wT= np.transpose(w)
e.append(epoch)
m.append(count)
```

In [34]:
```
while count != 0:
    epoch += 1
    count = 0
    for i in range(len(S1)):
        if np.matmul(S1[i], wT) < 0:
            w =  w + n*S1[i]
            count += 1
        i += 1
    for i in range(len(S2)):
        if np.matmul(S2[i], wT) >= 0:
            w =  w - n*S2[i]
            count += 1
        i += 1
    m.append(count)
    e.append(epoch)
    wT = np.transpose(w)
print("Final weights for learing rate 10 " + str(w))
```

```
Final weights for learing rate 10 [[ 1060.07611122 -2160.25697083  4093.18590
864]]
```

In [35]:
```
# Testing
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
print(count)
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1

count
```

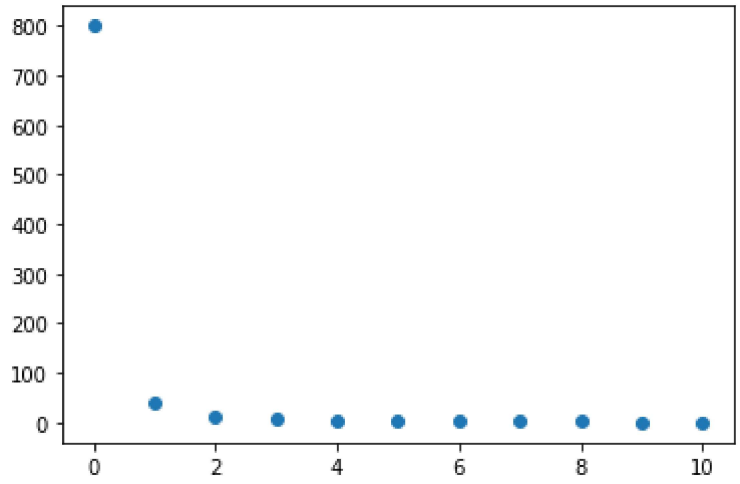
```
0
```

Out[35]:
```
0
```

In [36]:
```
print(e)
print(m)
plt.scatter(e,m)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[800, 41, 12, 8, 4, 2, 2, 2, 2, 1, 0]
```

Out[36]: <matplotlib.collections.PathCollection at 0x196a7e9eb70>



In [37]:
```
# Learning Rate 0.1
n = 0.1
w = t
wT = np.transpose(w)
e= []
m =[]
# Epoch 1
count = 0
epoch = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
        w =  w + n*S1[i]

    i += 1
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
        w =  w - n*S2[i]
    i += 1
wT= np.transpose(w)
e.append(epoch)
m.append(count)
```

In [38]:
```
while count != 0:
    epoch += 1
    count = 0
    for i in range(len(S1)):
        if np.matmul(S1[i], wT) < 0:
            w =  w + n*S1[i]
            count += 1
        i += 1
    for i in range(len(S2)):
        if np.matmul(S2[i], wT) >= 0:
            w =  w - n*S2[i]
            count += 1
        i += 1
    m.append(count)
    e.append(epoch)
    wT = np.transpose(w)
print("Final weights for learing rate 0.1 " + str(w))
```

```
Final weights for learing rate 0.1 [[ 10.47611122 -21.29689005  40.43167596]]
```

In [39]:
```
# Testing
count = 0
for i in range(len(S1)):
    if np.matmul(S1[i], wT) < 0:
        count += 1
    i += 1
print(count)
for i in range(len(S2)):
    if np.matmul(S2[i], wT) >= 0:
        count += 1
    i += 1

count
```

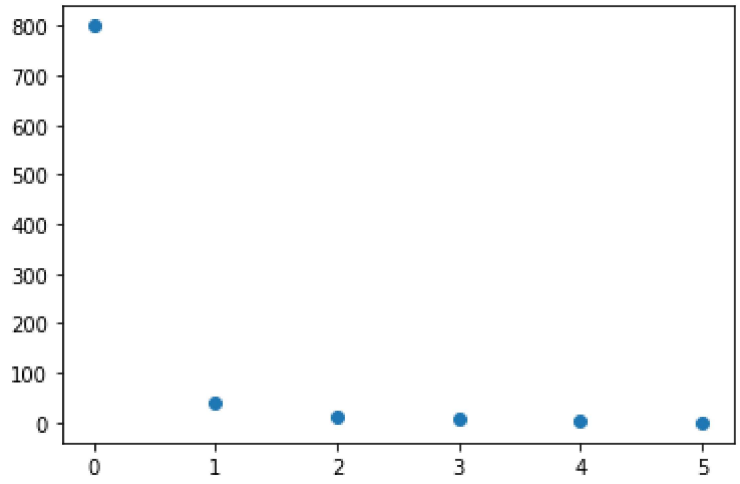
```
0
```

Out[39]:
```
0
```

In [40]:
```
print(e)
print(m)
plt.scatter(e,m)
```

```
[0, 1, 2, 3, 4, 5]
[800, 40, 13, 7, 2, 0]
```

Out[40]: <matplotlib.collections.PathCollection at 0x196a7f0a7b8>



In [ ]: