ECE-GY 6143 INTRO TO MACHINE LEARNING

HOMEWORK-2

Pratyush Shukla (ps4534)

Question 1:

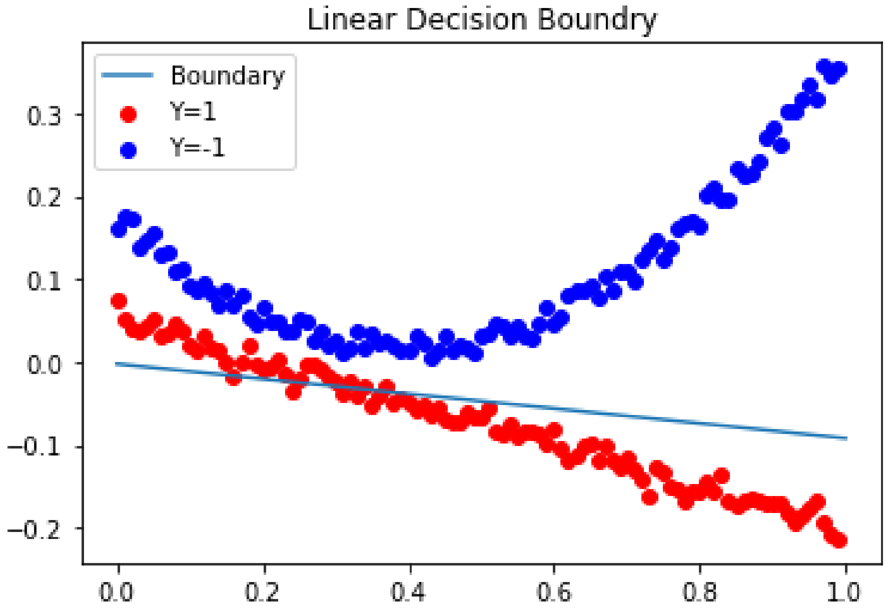
Code –

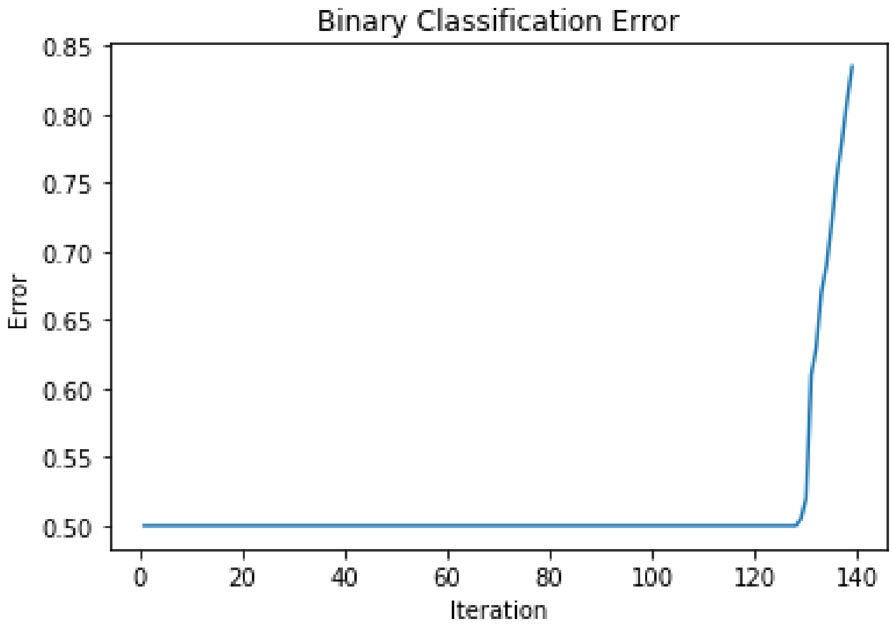
```
import scipy.io
import numpy as np
import matplotlib.pyplot as plt

data = scipy.io.loadmat('data3.mat')
data = data['data']

X = data[:,0:2]
y = data[:,-1]

X = np.hstack((X, np.ones((len(X), 1))))

theta = np.random.rand(3,1)
theta1 = np.ones((3,1)) * 15
eta = 0.4
tol = 0.002

loss =[]
err =[]
itr = []
grad = []

iteration=0
while np.linalg.norm(theta - theta1) > tol:
    iteration += 1
    itr.append(iteration)

    f = np.dot(X, theta)
    pLoss = np.zeros(len(f))
    gradient = np.zeros((len(f), 3))
```

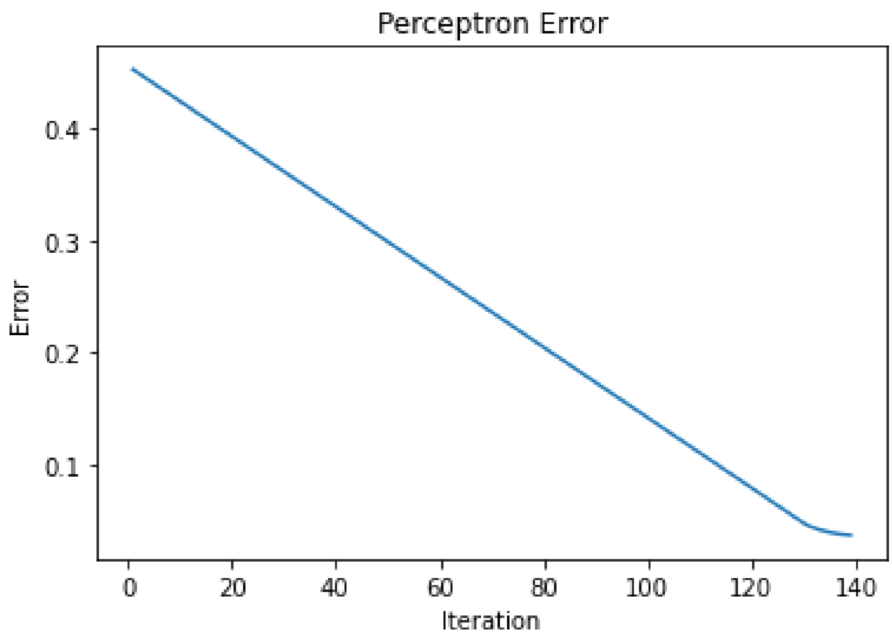
```
    misclassified = 0

    for i in range(len(f)):
        if f[i] * y[i] < 0:
            misclassified = misclassified + 1
            pLoss[i] = y[i] * f[i]
            gradient[i] = y[i] * X[i]

    loss.append(-1 * (1/len(X)) * sum(pLoss))

    gradientFinal = -(1/len(X))*sum(gradient)
    gradientFinal = gradientFinal.reshape((3,1))
    grad.append(gradientFinal)

    err.append((1/len(X)) * misclassified)

    theta1 = theta
    theta = theta1 - gradientFinal*eta

a = X[:, 0]
b = X[:, 1]

plt.figure()
plt.title('Linear Decision Boundry')
for i in range(len(y)):
    if y[i] == 1:
        plt.scatter(a[i], b[i], color='red')
    else:
        plt.scatter(a[i], b[i], color='blue')

xl = np.linspace(0, 1, 150)
```

```
g = (-1*theta[0]/theta[1])*xl - (theta[2]/theta[1])
plt.plot(xl, g)
plt.legend(['Boundary','Y=1', 'Y=-1'])
plt.show()


plt.figure()
plt.xlabel('Iteration')
plt.ylabel('Error')
plt.title('Binary Classification Error')
plt.plot(itr, err)
plt.show()


plt.figure()
plt.xlabel('Iteration')
plt.ylabel('Error')
plt.title('Perceptron Error')
plt.plot(itr, loss)
plt.show()
```

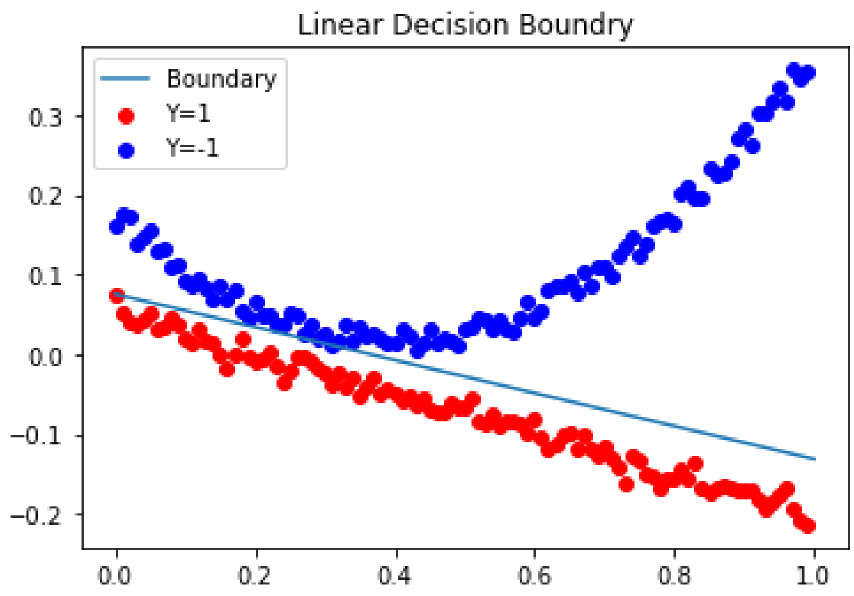
For step size = 0.1, the plots are as follows –

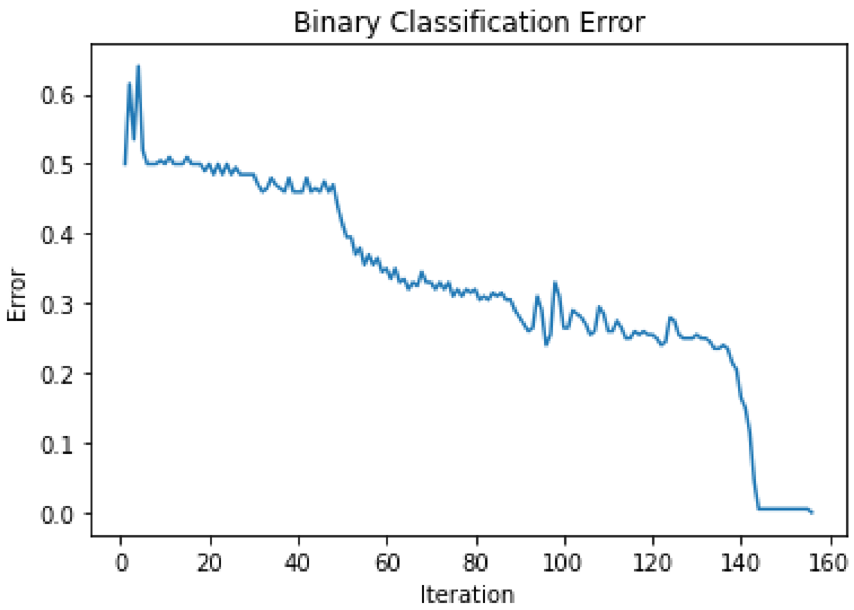


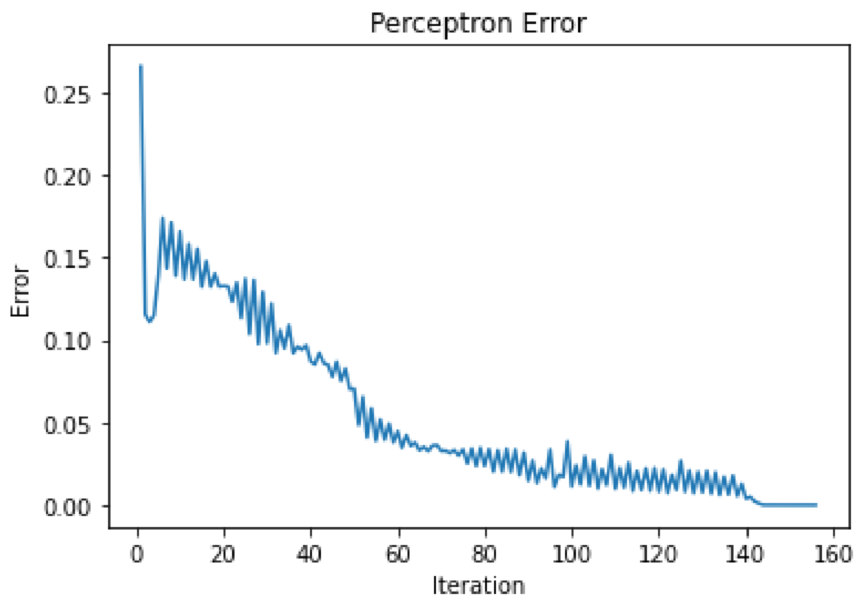




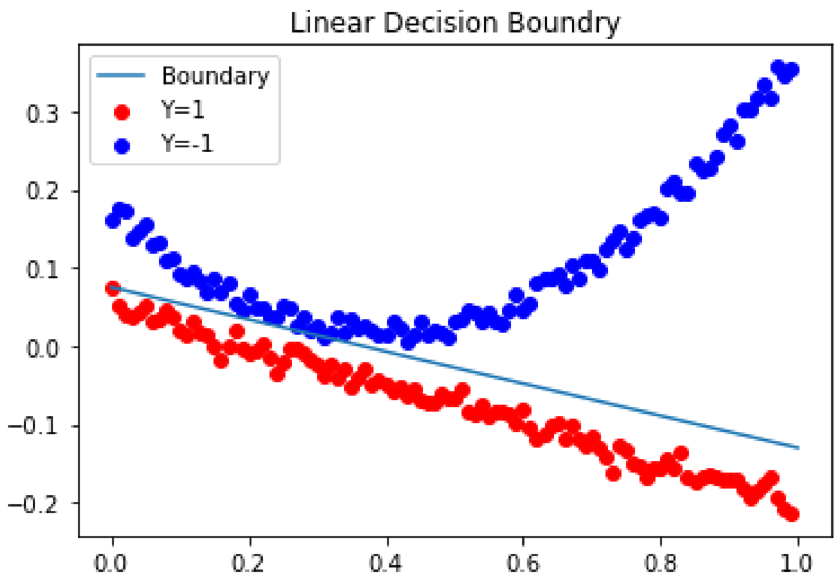
For step size = 0.09, the plots are as follows –

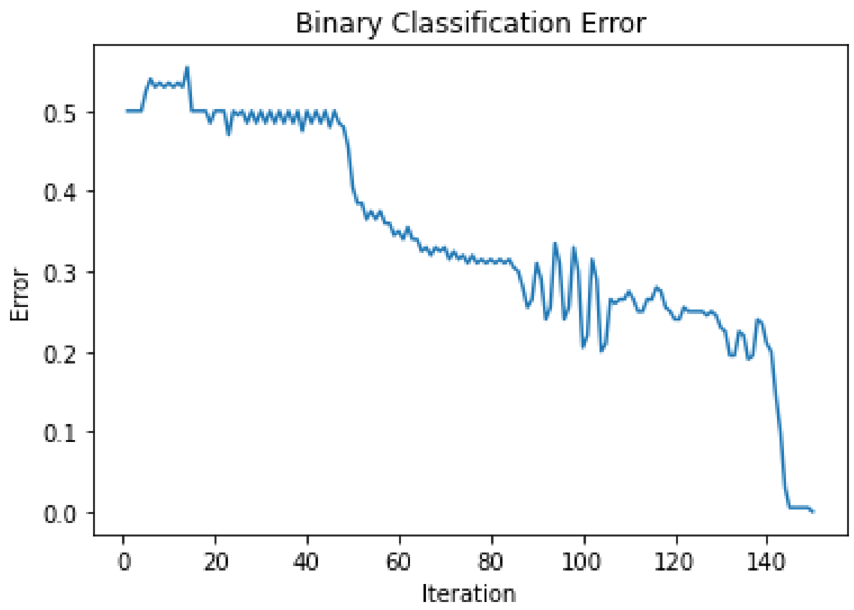


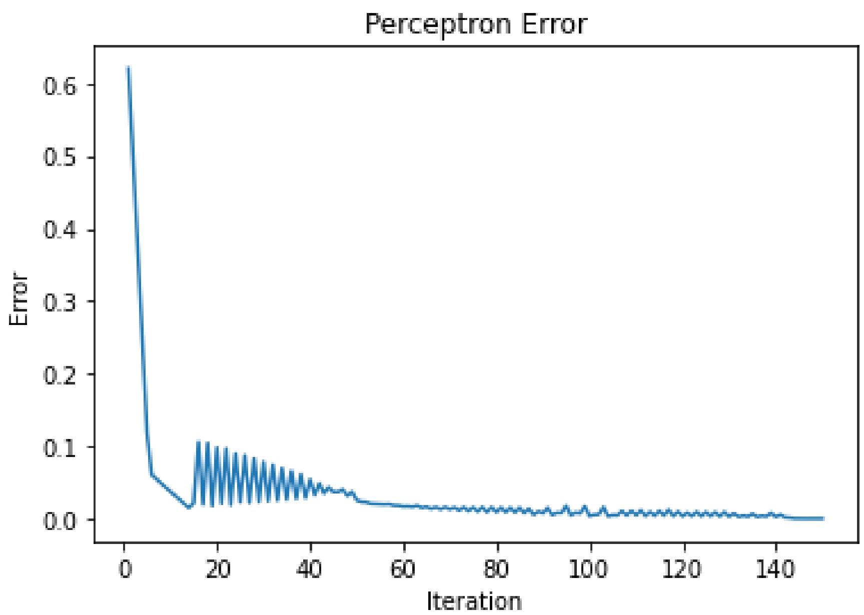




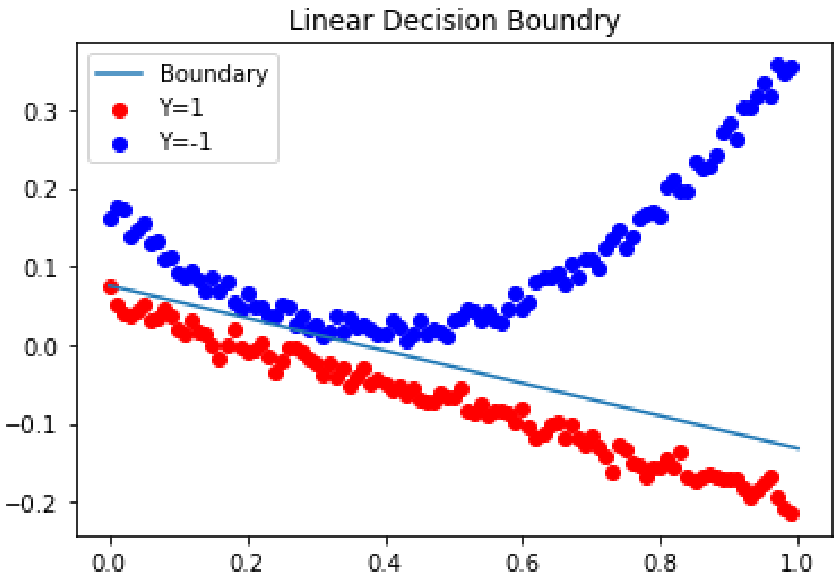
For step size = 0.4, the plots are as follows –

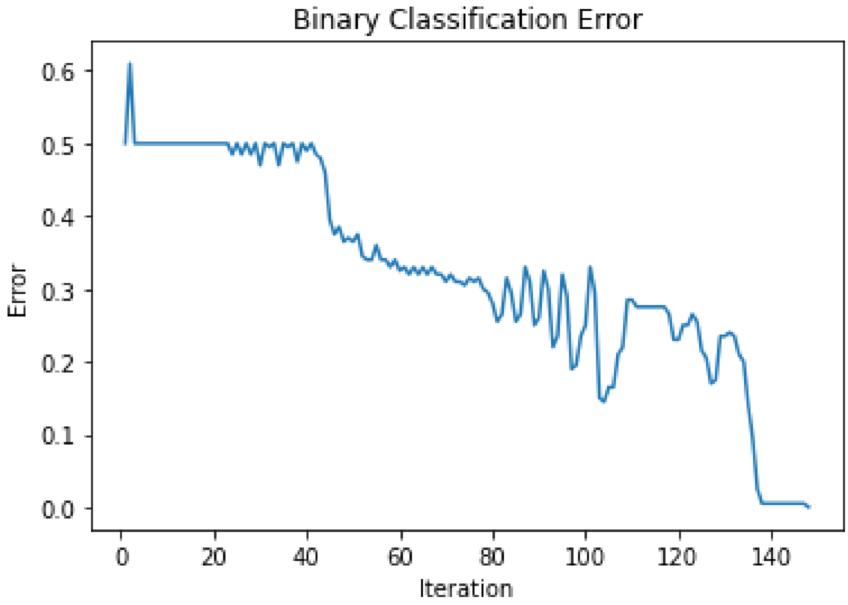


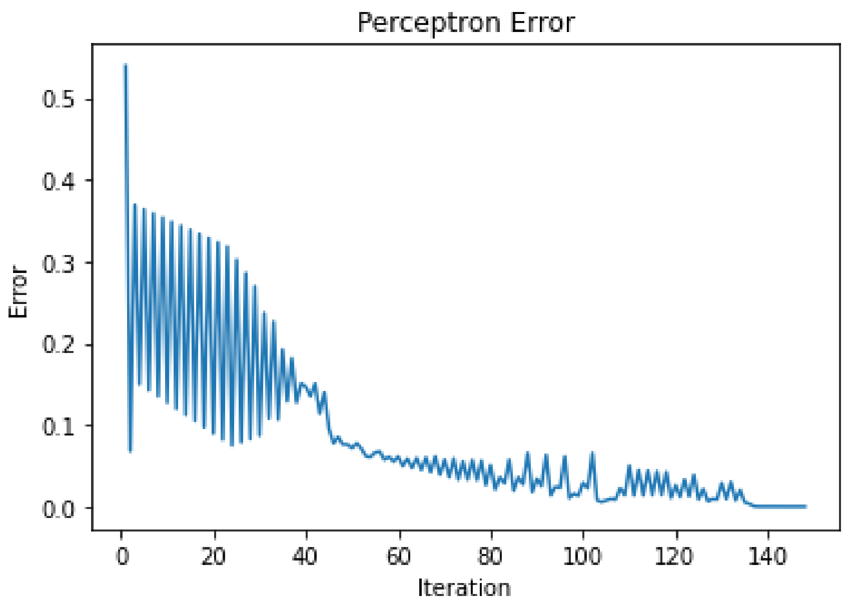




For step size = 1.5, the plots are as follows –







As illustrated in the figures, with increase in the step size, the errors gradually become volatile. This is due to the use of gradient descent algorithm, which is highly dependent on the learning rate i.e., step size. Low values of learning rate tend to not reach convergence while higher values will miss the convergence. Hence, the learning rate needs to be selected carefully for proper accuracy in a linearly separable data.

Question 2:

2a) $E = -\sum_i \left(t_i \log(x_i) + (1-t_i)\log(1-x_i)\right)$

$x_i = \frac{1}{1+e^{-s_i}}$ where $s_i = \sum_j y_j \omega_{ji}$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial x_i} = -\left(\frac{t_i}{x_i}\right) + \frac{1-t_i}{1-x_i} = \frac{x_i - t_i}{x_i(1-x_i)}$$

$$\frac{\partial x_i}{\partial s_i} = \frac{e^{-s_i}+1-1}{(1+e^{-s_i})^2} = x_i(1-x_i)$$

$$\frac{\partial s_i}{\partial w_{ji}} = y_j$$

So, $w_{ji}^{t+1} = w_{ji}^{t} - \eta_1 \frac{\partial E}{\partial w_{ji}}$

For input layer,

$$\frac{\partial E}{\partial y_j} = \sum_i \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial y_j}$$

$$\frac{\partial s_i}{\partial y_j} = w_{ji}$$

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$\frac{\partial y_j}{\partial s_j} = \frac{e^{-s_j}+1-1}{(1+e^{-s_j})^2} = y_j(1-y_j)$$

$$\frac{\partial s_j}{\partial w_{kj}} = z_k$$

So, $w_{kj}^{t+1} = w_{kj}^{t} - \eta_2 \frac{\partial E}{\partial w_{kj}}$

2b) $E = -\sum_i t_i \log(x_i)$

$$x_i = \frac{e^{s_i}}{\sum_{c=1}^{m} e^{s_c}}$$

$$\frac{\partial E}{\partial w_{ji}} = \sum_u \frac{\partial E}{\partial x_u}\frac{\partial x_u}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}}$$

$$\frac{\partial E}{\partial x_u} = -\frac{t_u}{x_u}$$

$$\frac{\partial x_u}{\partial s_i} = -(x_u)^2 \frac{e^{s_i}}{e^{s_u}}$$

$$\frac{\partial s_i}{\partial w_{ji}} = y_j$$

When $u=i$, $\frac{\partial x_i}{\partial s_i} = x_i - x_i^2$

$$\frac{\partial E}{\partial x_i} = -\frac{t_i}{x_i}$$

So, $w_{ji}^{t+1} = w_{ji}^{t} - \eta_1 \frac{\partial E}{\partial w_{ji}}$

For input layer,

$$\frac{\partial E}{\partial y_j} = \sum_i \frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial s_i}\frac{\partial s_i}{\partial y_j}$$

$$\frac{\partial s_i}{\partial y_j} = w_{ji}$$

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial s_j}\frac{\partial s_j}{\partial w_{kj}}$$

$$\frac{\partial y_j}{\partial s_j} = y_j - y_j^2$$

$$\frac{\partial s_j}{\partial w_{kj}} = z_k$$

So, $w_{kj}^{t+1} = w_{kj}^{t} - \eta_2 \frac{\partial E}{\partial w_{kj}}$

Question 3:

3) For a discrete distribution $\{p_k \mid k=1,2,\dots,N\}$

$$H = -\sum_{k=1}^{N} p_k \log p_k$$

$$L(p_k, \lambda_0) = -\sum_{k=1}^{N} p_k \log p_k + \lambda_0 \left(\sum_{k=1}^{N} p_k - 1\right.$$

$$\frac{\partial L}{\partial p_k} = 0 = -\log p_k - 1 + \lambda_0$$

$$\frac{\partial L}{\partial \lambda_0} = 0 = \sum_{k=1}^{N} p_k - 1$$

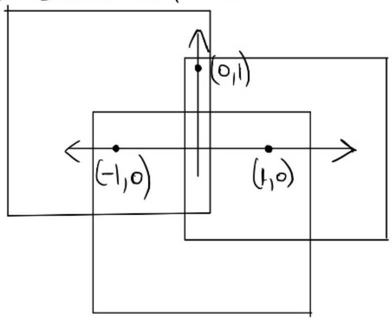
$$p_k = e^{-1+\lambda_0} \quad \text{with} \sum_{k=1}^{N} p_k = 1$$

$$p_k = \frac{1}{N+1}$$

Hence, the distribution that maximizes entropy is the Normal Distribution.

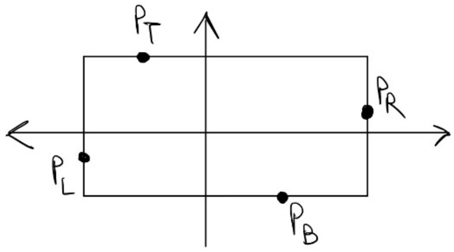
Question 4:

4) The VC-Dimension is 3.



The set of 3 co-ordinates (0,1), (1,0) & (-1,0) can be shattered by axis-aligned squares as shown above.

No set of 4 points can be fully shattered.



Let P_T be the highest point, P_B the lowest, P_L the leftmost & P_R the rightmost with the assumption they can be defined uniquely (no tie). Also assume without loss of generality that the difference d_{BT} of y-coordinates b/w P_T & P_B is greater than difference d_{LR} of x-coordinates b/w P_L & P_R.

Thus P_T & P_B cannot be labeled positively while P_L & P_R are labeled negatively.

Hence, VC-dimension is 3 & cannot be 4.