5.1

a) $P(y=1|x_1 \ldots x_d) = g\left(\sum_{i=1}^{d} w_i x_i\right) = g(\vec{w}.\vec{x})$

$$\mathcal{L} = \sum_t \log P(y_t|\vec{x}_t)$$

$$= \sum_t \log g(\vec{w}\cdot\vec{x}_t)^{y_t}(1-g(\vec{w}_t\cdot\vec{x}_t))^{(1-y_t)}$$

$$= \sum_t y_t \log g(\vec{w}.\vec{x}) + \sum_t (1-y_t)\log(1-(P(y_t|x_t))$$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial w_i} = \sum_t \frac{y_t\, g'(\vec{w}.\vec{x})}{g(\vec{w}.\vec{x})} x_{it}$$

$$- \sum_t \frac{(1-y_t)\, g'(\vec{w}.\vec{x})}{(1-g(\vec{w}.\vec{x}))} x_{it}$$

$$= \sum_t g'(\vec{w}.\vec{x})x_{it}\left(\frac{y_t}{g(\vec{w}.\vec{x})} - \frac{(1-y_t)}{(1-g(\vec{w}.\vec{x}))}\right)$$

$$= \sum_t \frac{g'(\vec{w}.\vec{x})\, x_{it}}{g(\vec{w}.\vec{x})(1-g(\vec{w}.\vec{x}))}\left(\cancel{g(w}\, y_t - g(\vec{w}.\vec{x})\right)$$

$$\boxed{\frac{\partial \mathcal{L}}{\partial w_i} = \sum_t \frac{g'(\vec{w}.\vec{x})x_{it}(y_t - P_t)}{P_t(1-P_t)}}$$

b) $$g(z) = \frac{1}{1+e^{-z}}$$

$$g'(z) = g(z)\,g(-z) \qquad \because g(z) = \sigma(z)$$

$$g(-z) = 1 - g(z)$$

$$\Rightarrow \frac{\partial \mathcal{L}}{\partial w_i} = \sum_{t=1}^{T} \frac{g'(\vec{w}\cdot\vec{x})}{g(\vec{w}\cdot\vec{x})(1-g(\vec{w}\cdot\vec{x}))} (y_t - P_t)\, x_{it}$$

$$= \sum_{t=1}^{T} \frac{\cancel{g(\vec{w}\cdot\vec{x})\, g(-\vec{w}\cdot\vec{x})}}{\cancel{g(\vec{w}\cdot\vec{x})(1-g(\vec{w}\cdot\vec{x}))}} (y_t - P_t)\, x_{it}$$

$$\boxed{\frac{\partial \mathcal{L}}{\partial w_i} = \sum_{t=1}^{T} x_{it}(y_t - P_t)}$$

5.2

$$P(y_t|\vec{x}_t) = \prod_{i=1}^{K} P_{it}^{y_{it}}$$

$$\mathcal{L} = \sum_t \log(P(y_t|\vec{x}_t))$$

$$= \sum_t \log \prod_{i=1}^{k} P_{it}^{y_{it}}$$

$$= \sum_t^T \sum_{i=1}^{k} \log P_{it}^{y_{it}} = \sum_t^T \sum_{i=1}^{k} y_{it} \log P_{it} \quad \ldots (i)$$

if $$h(x) = \frac{f(x)}{g(x)}$$

$$\Rightarrow \frac{d\,h(x)}{dx} = \frac{g(x) \cdot \frac{d f(x)}{dx} - f(x) \frac{d g(x)}{dx}}{(g(x))^2}$$

let, $$h(\vec{\omega}_i) = \frac{e^{\vec{\omega}_i \cdot \vec{x}}}{\sum_j e^{\vec{\omega}_j \cdot \vec{x}}}$$

$$\Rightarrow \frac{\partial h(\vec{\omega}_i)}{\partial \vec{\omega}_i} = \frac{\left(\sum_j e^{\vec{\omega}_j \cdot \vec{x}}\right) e^{\vec{\omega}_i \cdot \vec{x}} \vec{x} - e^{\vec{\omega}_i \cdot \vec{x}} \cdot e^{\vec{\omega}_i \cdot \vec{x}} \vec{x}}{\left(\sum_j e^{\vec{\omega}_j \cdot \vec{x}}\right)^2}$$

$$= \vec{x} P_{it}(1 - P_{it}) \quad \ldots (ii)$$

let, $h(\omega_i) = \dfrac{e^{\vec{\omega}_l \cdot \vec{x}}}{\sum_j e^{\vec{\omega}_j \cdot \vec{x}}}$ $\forall\ l \in \{1, \dots k\} \setminus \{i\}$

$$\Rightarrow \frac{\partial h(\vec{\omega}_i)}{\partial \vec{\omega}_i} = \frac{- e^{\vec{\omega}_l \cdot \vec{x}}\, e^{\vec{\omega}_i \cdot \vec{x}}\, \vec{x}}{\left(\sum_{j=1}^{k} e^{\vec{\omega}_j \cdot \vec{x}}\right)^2}$$

$$= - P_{lt} P_{it} \vec{x} \quad \dots (iii)$$

From (i)

$$\mathcal{L} = \sum_{t}^{T} \sum_{i=1}^{K} y_{it} \log P_{it}$$

$$\frac{\partial \mathcal{L}}{\partial \vec{\omega}_i} = \sum_{t}^{T} \left[\frac{y_{it}}{\cancel{P_{it}}} (1 - P_{it}) \cancel{P_{it}} \vec{x} - \sum_{l=1}^{K \setminus \{i\}} \frac{y_{lt}}{\cancel{P_{lt}}} \times \cancel{P_{lt}} P_{it} \vec{x} \right]$$

from (ii) & (iii)

$$= \sum_{t}^{T} \left[\left(y_{it} - \left(\sum_{l=1}^{K} y_{lt} \right) P_{it} \right) \vec{x} \right]$$

$$\boxed{\frac{\partial \mathcal{L}}{\partial \vec{\omega}_i} = \sum_{t}^{T} (y_{it} - P_{it}) \vec{x}}$$

5.3

a) $$x_{n+1} = x_n - \eta g'(x_n)$$

$$g(x) = \frac{\alpha}{2}(x - x_*)^2$$

$$\frac{d g(x)}{d x} = \alpha(x - x_*)$$

$$\Rightarrow \quad x_{n+1} = x_n - \eta\alpha(x_n - x_*)$$

$$x_{n+1} = x_n(1 - \eta\alpha) + \eta\alpha x_*$$

$$\varepsilon_n = x_n - x_*$$

$$= x_{n-1}(1 - \eta\alpha) + \eta\alpha x_* - x_*$$

$$= (x_{n-1} - x_*)(1 - \eta\alpha)$$

$$\boxed{\varepsilon_n = \varepsilon_{n-1}(1 - \eta\alpha)}$$

$$\Rightarrow \quad \boxed{\varepsilon_n = \varepsilon_0 (1 - \eta\alpha)^n}$$

b) Since the update rule needs to converge,

$$\left| \frac{\varepsilon_{n+1}}{\varepsilon_n} \right| < 1$$

$$\Rightarrow \left| \frac{\varepsilon_n (1-\eta\alpha)}{\varepsilon_n} \right| < 1$$

$$\Rightarrow \cancel{\varepsilon} \left| 1-\eta\alpha \right| < 1$$

$$\Rightarrow 1-\eta\alpha < 1 \quad \& \quad \eta\alpha - 1 < 1$$

$$\Rightarrow 0 < \eta \quad \& \quad \eta < \frac{2}{\alpha}$$

$$\Rightarrow \boxed{0 < \eta < \frac{2}{\alpha}}$$

For fastest conversion.

$$\varepsilon_1 = \varepsilon_0 (1-\eta\alpha) = 0$$

$$\Rightarrow \cancel{\eta} \boxed{\eta_{fast} = \frac{1}{\alpha}}$$

$$\because g''(x_n) = \alpha \Rightarrow \boxed{\eta_{fast} = \frac{1}{g''(x_n)}}$$

c) $x_{n+1} = x_n - \eta g'(x_n) + \beta(x_n - x_{n-1})$

$g'(x_n) = \alpha(x_n - x_*)$

$\Rightarrow x_{n+1} = x_n - \eta\alpha(x_n - x_*) + \beta(x_n - x_{n-1})$

$x_{n+1} = x_n(1 - \eta\alpha + \beta) + \eta\alpha x_* - \beta x_{n-1}$ ...(i)

$\varepsilon_{n+1} = x_{n+1} - x_*$

$= x_n(1 - \alpha\eta + \beta) + \alpha\eta x_* - \beta x_{n-1} - x_*$

$= x_n(1 - \alpha\eta + \beta) - (1 - \alpha\eta + \beta)x_* - \beta(x_{n-1} - x_*)$

$= (x_n - x_*)(1 - \alpha\eta + \beta) - \beta(x_{n-1} - x_*)$

$$\boxed{\varepsilon_{n+1} = \varepsilon_n(1 - \alpha\eta + \beta) - \beta\varepsilon_{n-1}}$$

d) $\varepsilon_{n+1} = (1 - \alpha\eta + \beta)\varepsilon_n - \beta\varepsilon_{n-1}$ ...(i)

$\varepsilon_n = \lambda^n \varepsilon_0$ ...(ii)

substituting (ii) in (i)

$\varepsilon_0 \lambda^{n+1} = (1 - \alpha\eta + \beta)\varepsilon_0 \lambda^n - \beta\varepsilon_0\lambda^{n-1}$

$\Rightarrow \lambda^2 = (1 - \alpha\eta + \beta)\lambda - \beta$ (iii)

Given;

$\alpha = 1$

$\eta = \frac{4}{9}$

$\beta = \frac{1}{9}$

substituting in (iii)

$\Rightarrow \lambda^2 = \left(1 - \frac{4}{9} + \frac{1}{9}\right)\lambda - \frac{1}{9}$

$\Rightarrow 9\lambda^2 - 6\lambda + 1 = 0$

$\Rightarrow (3\lambda - 1)^2 = 0$

$\Rightarrow \boxed{\lambda = \frac{1}{3}} = 0.\bar{3}$

For $\beta = 0$

$$\Rightarrow \lambda^2 = \left(1 - \frac{4}{9}\right)\lambda$$

$$\Rightarrow \boxed{\lambda = \frac{5}{9}} = 0.\bar{5}$$

We can see that

$$\lambda_{\beta=0} > \lambda_{\beta=\frac{1}{9}}$$

$\Rightarrow$ Rate of convergence of gradient descent with momentum parameter is faster compared to without momentum parameter.

5.4

a) $$x_{n+1} = x_n - \frac{g'(x)}{g''(x)} \quad \ldots (i)$$

$$g(x) = (x - x_*)^{2p}$$

$$g'(x) = 2p(x - x_*)^{2p-1}$$

$$g''(x) = 2p(2p-1)(x - x_*)^{2p-2}$$

$\Rightarrow$ Substituting in (i).

$$x_{n+1} = x_n - \frac{2p(x_n - x_*)^{2p-1}}{2p(2p-1)(x_n - x_*)^{2p-2}}$$

$$= x_n - \frac{x_n - x_*}{2p-1}$$

$$= \frac{(2p-2)x_n + x_*}{2p-1}$$

$$x_{n+1} = \frac{2x_n(p-1) + x_*}{(2p-1)} \quad \ldots (ii)$$

Given :-

$$\mathcal{E}_n = |x_n - x_*|$$

substituting x_n from (ii)

$$\Rightarrow \quad \mathcal{E}_n = \left| \frac{2x_{n-1}(p-1) + x_*}{(2p-1)} - x_* \right|$$

$$\Rightarrow \quad = \left| \frac{2x_{n-1}(p-1) - 2x_*(p-1)}{2p-1} \right|$$

$$= \frac{2(p-1)}{(2p-1)} \left| x_{n-1} - x_* \right|$$

$$\mathcal{E}_n = \frac{2(p-1)}{(2p-1)} \mathcal{E}_{n-1} \quad \cdots \text{(iii)}$$

Performing the substitution of (iii) for each smaller $\mathcal{E}_{n-i}$, n times $\Rightarrow$

$$\boxed{\mathcal{E}_n = \left(\frac{2(p-1)}{2p-1} \right)^n \mathcal{E}_0}$$

b) $\mathcal{E}_n \leq \delta \mathcal{E}_0$

from 5.4 (a), we have:-

$$\mathcal{E}_0 \left(\frac{2(p-1)}{2p-1}\right)^n \leq \delta \mathcal{E}_0$$

$$\Rightarrow \left(1 - \frac{1}{2p-1}\right)^n \leq \delta$$

Taking log both sides

$$n \log\left(1 - \frac{1}{2p-1}\right) \leq \log \delta$$

$$\Rightarrow -n \log\left(1 - \frac{1}{2p-1}\right) \geq \log \frac{1}{\delta} \quad \ldots (i)$$

$$\because \delta < 1 \Rightarrow \log \frac{1}{\delta} > 0$$

$$\Rightarrow -n \log\left(1 - \frac{1}{2p-1}\right) \geq 0 \quad \ldots (ii)$$

$$\Rightarrow \because \log z \leq z - 1 \quad \ldots (iii)$$

substituting (iii) in (ii)

$$\Rightarrow \quad -n\left(1-\frac{1}{2p-1}-1\right) \geq -n\log\left(1-\frac{1}{2p-1}\right) \quad \text{(iv)}$$

substituting in (i)

$$\Rightarrow \quad -n\left(-\frac{1}{2p-1}\right) \geq \log\frac{1}{\delta}$$

$$\Rightarrow \quad \frac{n}{2p-1} \geq \log\frac{1}{\delta}$$

$$\Rightarrow \quad \boxed{n \geq (2p-1)\log 1/\delta}$$

c) $g(x) = x_* \log(x_*/x) - x_* + x$

$$g'(x) = \frac{-x_*}{x_*} \times x \times \frac{x_*}{x^2} + 1$$

$$= 1 - \frac{x_*}{x}$$

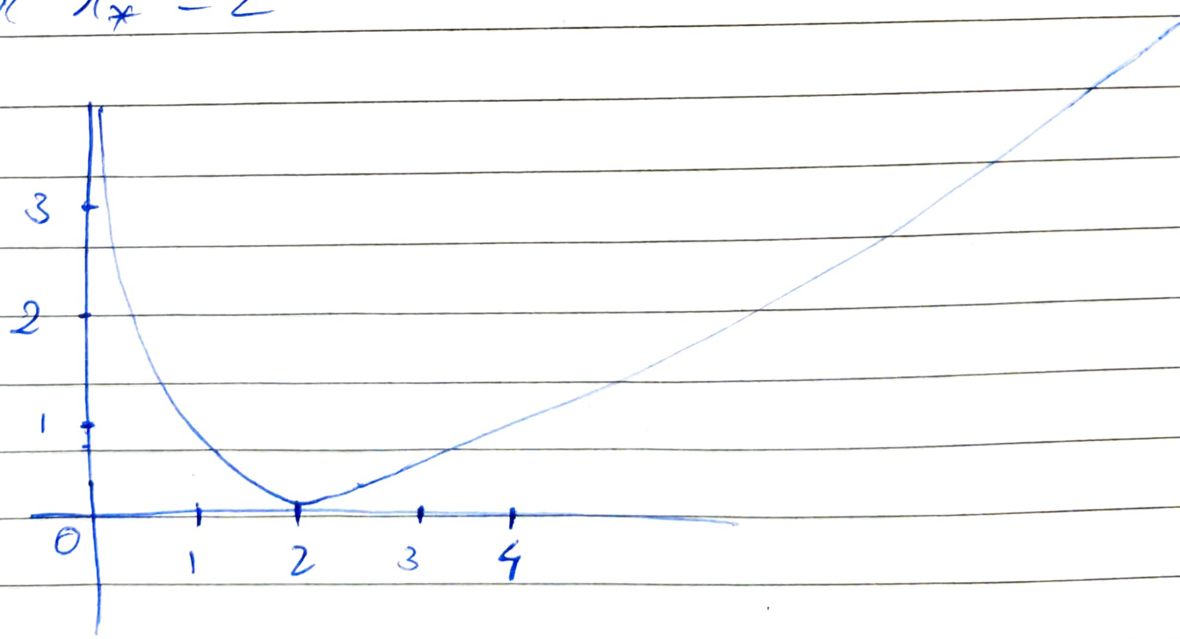
$$g''(x) = \frac{x_*}{x^2} \Rightarrow g''(x) > 0$$

$\Rightarrow g'(x) = 0$ gives min.

$\Rightarrow g'(x) = 0 \Rightarrow \boxed{x = x_*}$ gives minima

let $x_* = 2$

$\Rightarrow$



$|x-2| < 2 \Rightarrow 0 < x < 4$

a)
$$\rho_n = \frac{(x_n - x_*)}{x_*} = \frac{x_n}{x_*} - 1$$

$$\rho_0 = \frac{x_0}{x_*} - 1$$

$$x_{n+1} = x_n - \frac{g'(x_n)}{g''(x_n)}$$

$$= x_n - \frac{\left(1 - \frac{x_*}{x_n}\right)}{\frac{x_*}{x_n^2}}$$

$$= x_n - \frac{(x_n - x_*)\,x_n}{x_*}$$

$$x_{n+1} = \frac{2x_n x_* - x_n^2}{x_*} \quad \ldots (i)$$

Now,

$$\rho_n = \frac{x_n}{x_*} - 1$$

$$= \frac{1}{x_*}\left(\frac{2x_{n-1}x_* - x_{n-1}^2}{x_*}\right) - 1 \quad \text{from (i)}$$

$$= \frac{2x_{n-1}x_* - x_{n-1}^2 - x_*^2}{x_*^2}$$

$$= \frac{-(x_{n-1} - x_*)^2}{x_*^2}$$

$$f_n = -f_{n-1}^2$$

$$\Rightarrow f_n = -\left(-f_{n-2}^2\right)^2$$

$$= -f_{n-2}^4$$

$$\Rightarrow \cancel{\boxed{f_n = -f_0^{2n}}} \quad \boxed{f_n = -f_0^{2^n}}$$

Now, f_n decays when $|f_0| < 1$

$$\Rightarrow \left|\frac{x_0}{x_*} - 1\right| < 1$$

$$\Rightarrow -1 < \frac{x_0}{x_*} - 1 < 1$$

$$\Rightarrow \boxed{0 < x_0 < 2x_*}$$

Q 5.5

In [114]:

```
import random
import math
import matplotlib.pyplot as plt
import numpy as np
from sklearn.utils import shuffle
from sklearn.metrics import accuracy_score
```

In [4]:

```
def parseFile(filename):
    contents = []
    with open(filename) as f:
        for line in f:
            contents.append([int(item) for item in line.strip('\n').split()])
    return contents
```

In [5]:

```
train3file = 'train3.txt'
train5file = 'train5.txt'
test3file = 'test3.txt'
test5file = 'test5.txt'
```

In [25]:

```
train3 = parseFile(train3file)
train5 = parseFile(train5file)
test3 = parseFile(test3file)
test5 = parseFile(test5file)
```

In [26]:

```
ytrain3 = [0] * len(train3)
ytrain5 = [1] * len(train5)
ytest3 = [0] * len(test3)
ytest5 = [1] * len(test5)

train3.extend(train5)
test3.extend(test5)
ytrain3.extend(ytrain5)
ytest3.extend(ytest5)

x_train = train3
x_test = test3
y_train = ytrain3
y_test = ytest3
```

In [20]:

```
def listToNparray(listOfList):
    return np.array([np.array(xi) for xi in listOfList])
```

In [31]:

```
x_train = listToNparray(x_train)
x_test = listToNparray(x_test)
y_train = listToNparray(y_train)
y_test = listToNparray(y_test)
```

In [34]:

```
x_train, y_train = shuffle(x_train, y_train, random_state=0)
x_test, y_test = shuffle(x_test, y_test, random_state=0)
```

In [92]:

```
def sigmoid(x):
    return 1 / (1 + math.exp(-x))

sigmoid_vec = np.vectorize(sigmoid)
```

In [110]:

```
def gradient(y_data, x_data, w):
    arr = y_data - sigmoid_vec(x_train.dot(w))
    return np.sum((np.expand_dims(arr, axis=1) * x_data), axis=0)
```

In [111]:

```
def logLikelihood(y_data, x_data, w):
    vec_log = np.vectorize(math.log)
    sigma = sigmoid_vec(x_data.dot(w))
    return np.sum(y_data * vec_log(sigma) + ((1 - y_data) * vec_log(1 - sigma)), axis = 0)
```

In [146]:

```
def percentErrorRate(y_data, x_data, w):
    sigma = sigmoid_vec(x_data.dot(w))
    sigma[sigma >= 0.5] = 1
    sigma[sigma < 0.5] = 0
    return (1 - accuracy_score(y_data, sigma)) * 100
```

In [139]:

```
lr = 0.2/len(train3)
w = np.array([random.random() for i in range(64)])
```

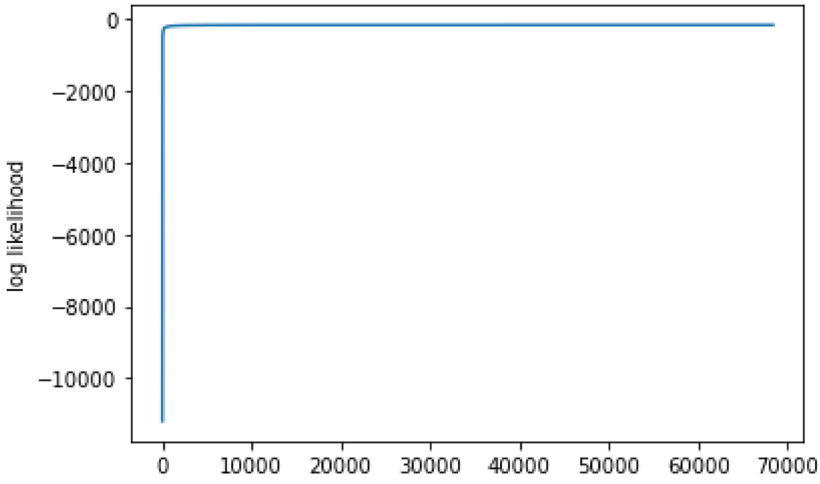
In [140]:

```
likelihoods = []
error_rates = []

while(True):
    error_rates.append(percentErrorRate(y_train, x_train, w))
    likelihoods.append(logLikelihood(y_train, x_train, w))
    new_w = w + (lr * gradient(y_train, x_train, w))
    if np.sum(abs(new_w - w)) < 0.000001:
        break
    w = new_w
```

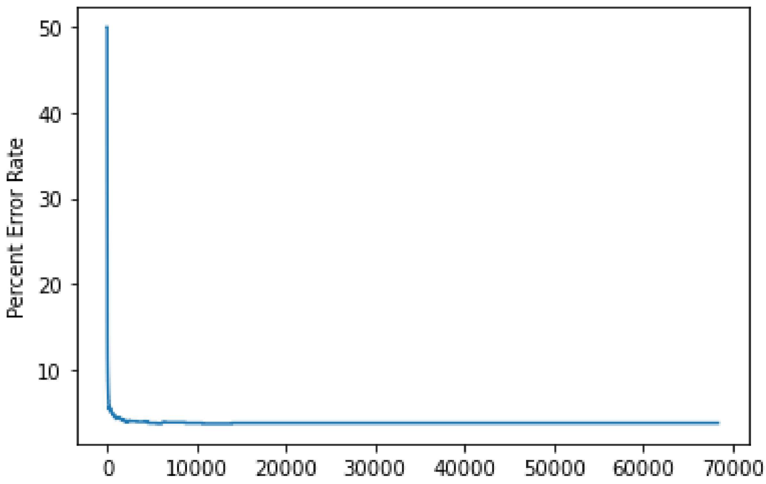
In [141]:

```
plt.plot(likelihoods)
plt.ylabel('log likelihood')
plt.show()
```



In [142]:

```
plt.plot(error_rates)
plt.ylabel('Percent Error Rate')
plt.show()
```



In [143]:

```
print('Training Data:')
print('Percent Error Rate: ', error_rates[-1])
print('Log Likelihood: {}', likelihoods[-1])
```

```
Training Data:
Percent Error Rate: 3.785714285714281
Log Likelihood: -160.69474592974012
```

In [144]:

```
w.reshape((8, 8))
```

Out[144]:

```
array([[-0.69894   , -1.79069835, -1.09593222, -1.55910935, -0.61281589,
        -1.19585413,  0.80501971,  1.98157463],
       [-0.30662466, -0.27510221,  0.33725763, -0.03486635, -0.70225478,
         1.00807412, -1.50060329, -1.51395157],
       [ 4.53778378,  1.39875248,  1.62982922,  0.09538458,  1.03742834,
        -2.47927927, -2.46688993, -2.94551655],
       [ 0.75360529,  0.36365432,  0.79404835, -0.36574538, -0.53219836,
        -2.81289441,  0.5334757 , -0.06481029],
       [ 0.66709125,  1.33469587,  0.11231855, -0.48299606, -0.63111966,
        -0.03003996, -0.67680811, -0.06051637],
       [ 1.3430501 , -0.30005544, -0.45764991, -0.22806255, -0.05443981,
        -1.17036269,  1.03806527, -1.89777205],
       [ 1.7596746 , -0.78112313,  1.42567803,  0.74172486,  0.5411414 ,
        -0.47610008,  0.12114978, -1.76621289],
       [ 0.74667644,  0.36042091,  0.78657701,  2.71799425,  0.4308032 ,
         0.75479189,  0.991657  , -0.63365613]])
```

In [147]:

```
print('Testing Data:')
print('Percent Error Rate: ', percentErrorRate(y_test, x_test, w))
print('Log Likelihood: ', logLikelihood(y_test, x_test, w))
```

```
Testing Data:
Percent Error Rate:  6.6250000000000036
Log Likelihood:  -127.17772193651534
```