Q.6.1.2

Ex 5.18

from D layer to k-dimensional output layer we introduce DMK (skip layer) which is

$$x_i, i=1 \cdots D \xrightarrow{w_{ji}^{(1)}} z_j, j=1 \cdots M \xrightarrow{w_{kj}^{(2)}} y_k, k=1 \cdots K$$

$$x_i \longrightarrow y_k$$

$$w_{ki}^{(3)}$$

so we can write y_k for forward propagation with skip layer.

$$y_k = \sum_{j=0}^{M} w_{kj}^{2} z_j + \underbrace{\sum_{i=0}^{D} w_{ki}^{(3)} x_i}_{\text{skip layer}}$$

The Derivation of $E(w)$ wrt $w_{ji}^{(1)}$ and $w_{kj}^{(1)}$ remain same since these weights are not dependent on the $w_{ki}^{(2)}$

so, differentiating w.r.t w_{ki} gives

$$\frac{\partial E_n}{w_{ki}^{(4)}} = (y_n - t_n) \partial_{w_{ki}}^{(3)} y_k$$

$$= (y_n - t_n) x_i \cong \delta_k x_i$$

Q. 6.2.1

Ex Q. 23.1

a) A fundamental theorm in Linear Algebra states that If V, W are finite dimensional Vector spaces, and let T be a Linear transformation from V to W, then the Image of T is a finite-dimensional Subspace of W and

$$\dim(V) = \dim(\text{null}(T)) + \dim(\text{image}(T)).$$

we can say $\dim(\text{null}(A)) \geq 1$. thus

$\exists\, v \neq 0$ such that $Av = A0 = 0$, Hence $\exists\, u \neq v \in \mathbb{R}^n$

$\Rightarrow$ $\boxed{Au = Av}$

b) let f be recovery function, let $u \neq v \in \mathbb{R}^n \Rightarrow Au = Av$

Hence $f(Au) = f(Av)$ so at least one of the vector u, v is not recovered.

Ex 23.3

let assume feature space is of finite dimensions let X where

$\phi(x_j)$ is j^{th} column

So, we can find spectral decomposition of $X^T X$.

$$\therefore \quad (X^T X)_{ij} = K(x_i, x_j)$$

We can find Efficient solution in case of $d >> m$, So the Eigen decomposition of $X^T X$ can also be find in polynomial time

let V be matrix with n leading Eigenvector of $X^T X$ as column.

and D be a diagonal $n \times n$ matrix whose diagonal consist of the corresponding Eigen values.

V is the matrix whose column are n leading Eigen vector of XX^T

so, for $x \in X$ the $U^T \phi(x)$ is $D^{-1/2} V^T X^T \phi(x)$

$$= D^{-1/2} V^T X^T \phi(x) = D^{-1/2} V^T \begin{bmatrix} K(x_1, x) \\ K(x_m, x) \end{bmatrix}$$

Ex 23.4

a) Note that for every Unit vector $w \in \mathbb{R}^d$, $i \in [m]$

$$(\langle w, x_i \rangle)^2 = tr(w^T x_i \cdot x_i^T w).$$

Hence, the Optimization problem here coincides with the optimization problem objective of $n=1$ PCA, Hence the Optimal solution of our varience Maximization problem is the first principle vector of $x_1 \dots x_m$.

(b)

$$w^* = \underset{\|w\|=1,\ \langle w, w_1\rangle > 0}{\operatorname{argmax}} \frac{1}{m}\sum_{i=1}^{m}(\langle w, x_i\rangle)^2$$

$$= \underset{\|w\|=1,\ \langle w, w_i\rangle = 0}{\operatorname{argmax}} tr\left(w^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T w\right)$$

PCA problem in case of $n=2$ is equivalent to finding a Unity matrix $W \in R^{d\times 2}$

$$\Rightarrow W^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T W \text{ is maximized}$$

w_1 & w_2 optimal matrix W's column and two first principal Vectors of $x_1 \dots x_m$

$$W^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T W$$

$$= w_1^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T w_i + w_2^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T w_i$$

Since w^* & w_1 are orthonormal, we get

$$= w_1^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T w_1 + w_2^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T w_2 \quad \text{——— (1)}$$

&

$$= w_1^T \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T w_1 + w^{*T} \frac{1}{m}\sum_{i=1}^{m} x_i x_i^T w^* \quad \text{——— (2)}$$

So, we can say that (1) $\geq$ (2)

Hence, we can conclude that $\boxed{w^* = w_2}$

Q.6.2.2

Ex 20.5

(a) we have,

$$C = \frac{1}{n}\left([I - v_1 v_1^T] X^T X [I - v_1 v_1^T]\right)$$

$$= \frac{1}{n}\left((X^T X - v_1 v_1^T X^T X)(I - v_1 v_1^T)\right)$$

$$= \frac{1}{n}\left[X^T X - v_1 (v_1^T X^T X) - (X^T X v_1) v_1^T + v_1 (v_1^T n \lambda_1 v_1) v_1^T\right]$$

$$= \frac{1}{n}\left[X^T X - n\lambda v_1 v_1^T - n\lambda_1 v_1 v_1^T + n\lambda v_1 v_1^T\right]$$

$$C \Rightarrow \frac{1}{n}\left[X^T X - n\lambda v_1 v_1^T\right] = \frac{1}{n} X^T X - \lambda v_1 v_1^T$$

Hence proved.

(b) since $\tilde{x}$ lives in d-1 subspace orthogonal to v_1 the vector u must be orthogonal to v_1. Hence

$u^T v_1 = 0$ & $u^T u = 1$ so, $u = v_2$

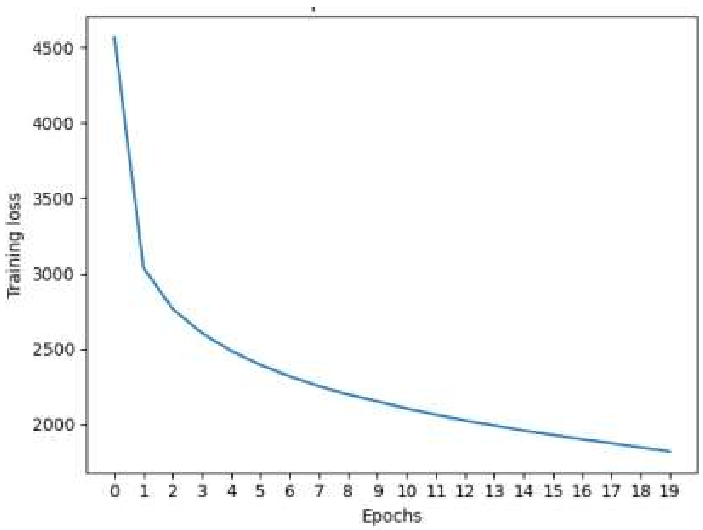
(c) we have

```
function [v, lambda] = Simple PCA(C, k, f)
d = length(C)
V = zeros(d, k)
for j = 1:k
    [lambda(1), V(:, J)] = f(C);
    C = C - lambda(1) * (V(:, J)); % depration
end.
```

Q .6 .1

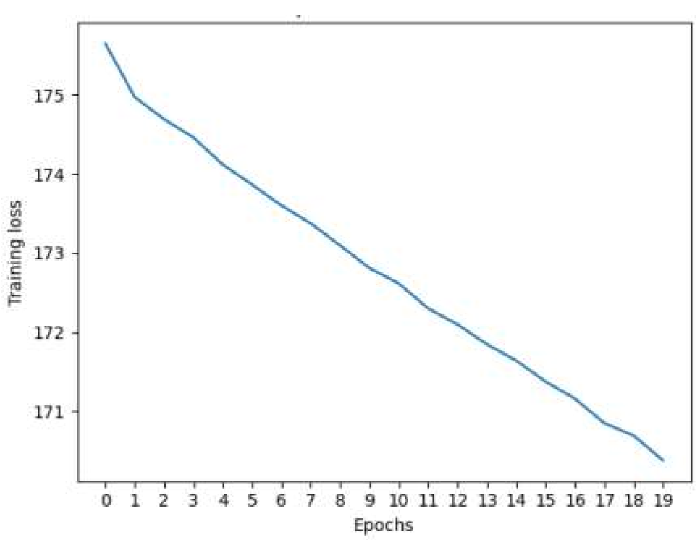
Plot for training loss over epochs Depth : 1 , Batch Size : 10

With accuracy : 87.2%



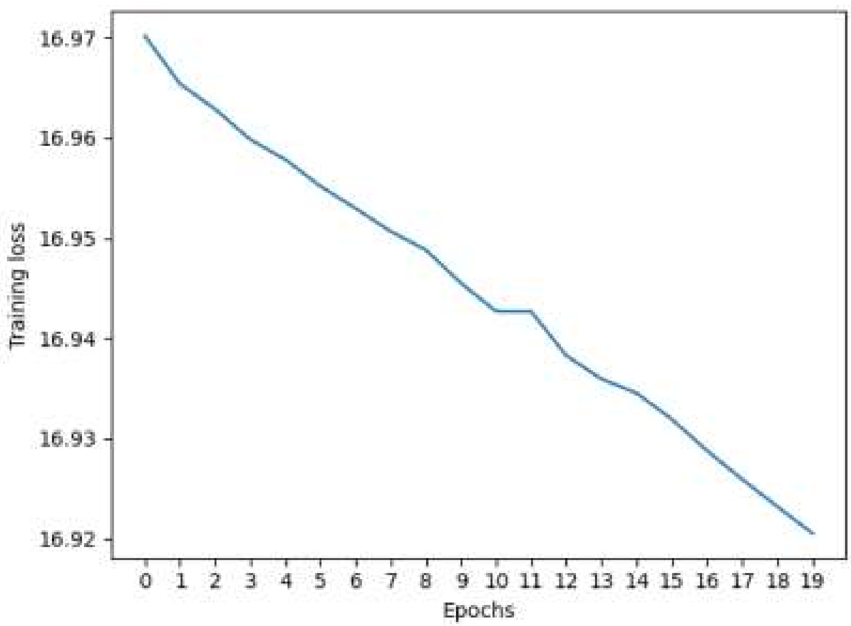
Plot for training loss over epochs Depth : 1 , Batch Size : 100

With accuracy : 87.5%



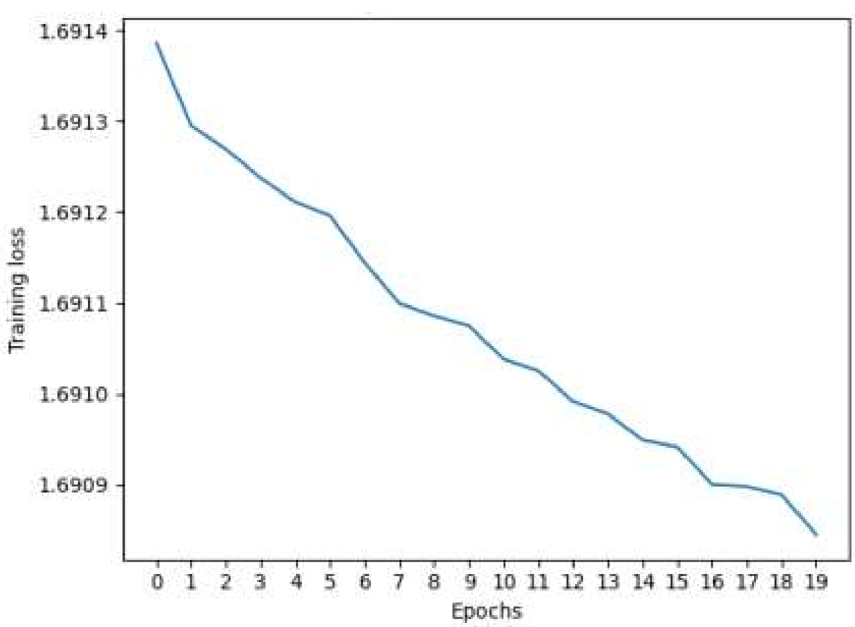
Plot for training loss over epochs Depth : 1 , Batch Size : 1000

With accuracy : 87.61%



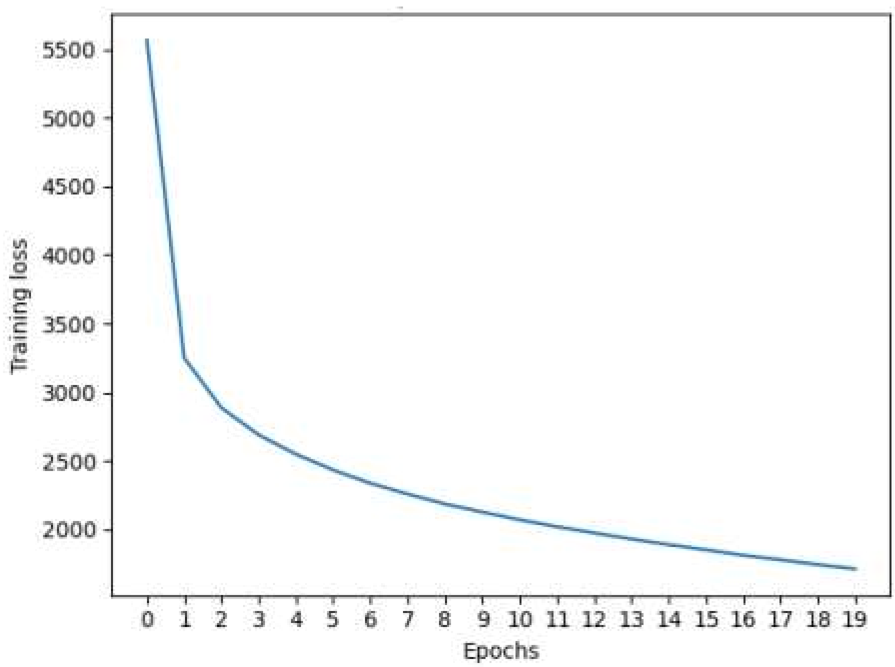
Plot for training loss over epochs Depth : 1 , Batch Size : 10000

Accuracy is : 87.55%



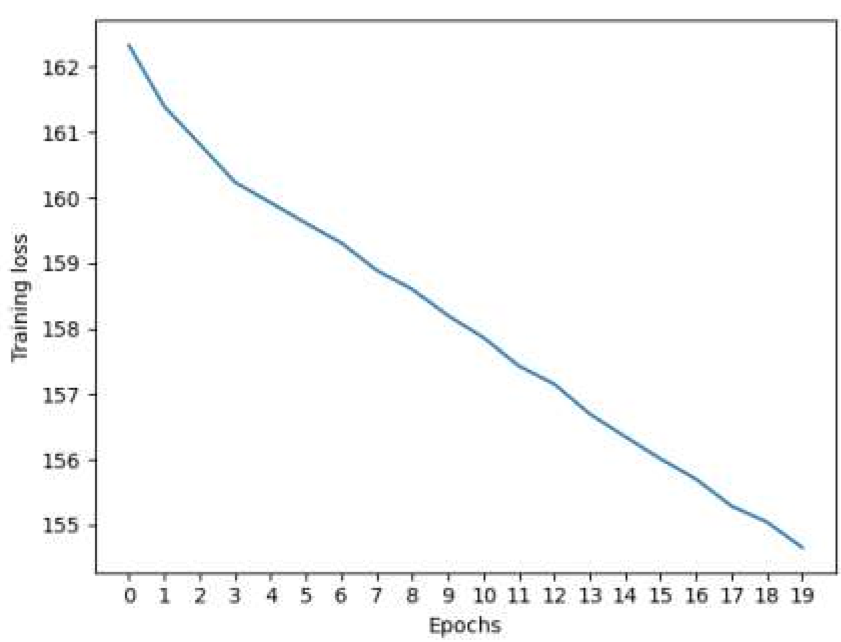
Plot for training loss over epochs Depth : 2 , Batch Size : 10

With accuracy : 87.4%



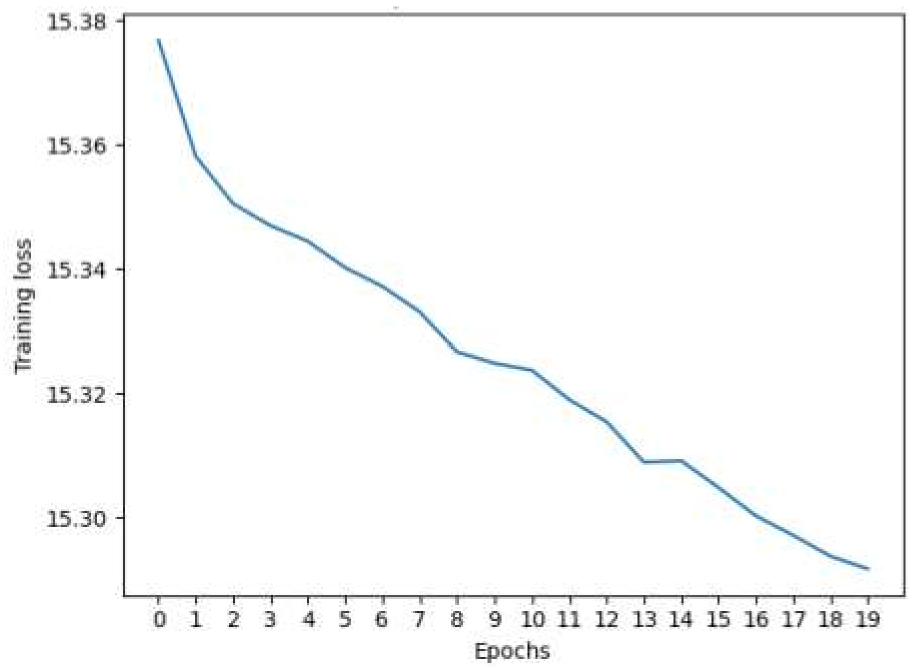
Plot for training loss over epochs Depth : 2 , Batch Size : 100

With accuracy : 88.01%



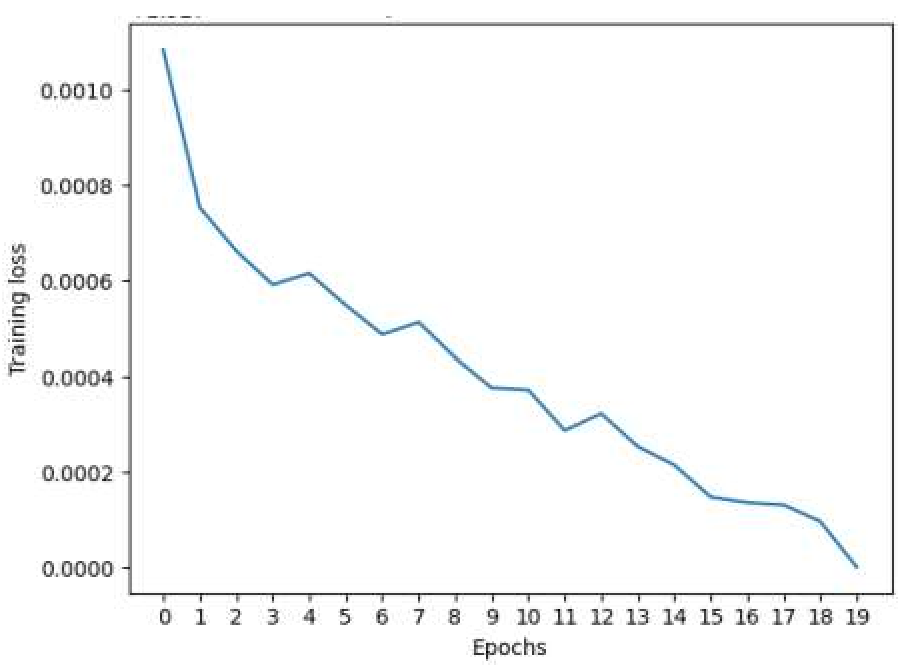
Plot for training loss over epochs Depth : 2 , Batch Size : 1000

With accuracy : 88.04%



Plot for training loss over epochs Depth : 2 , Batch Size : 10000

With accuracy : 88.12%

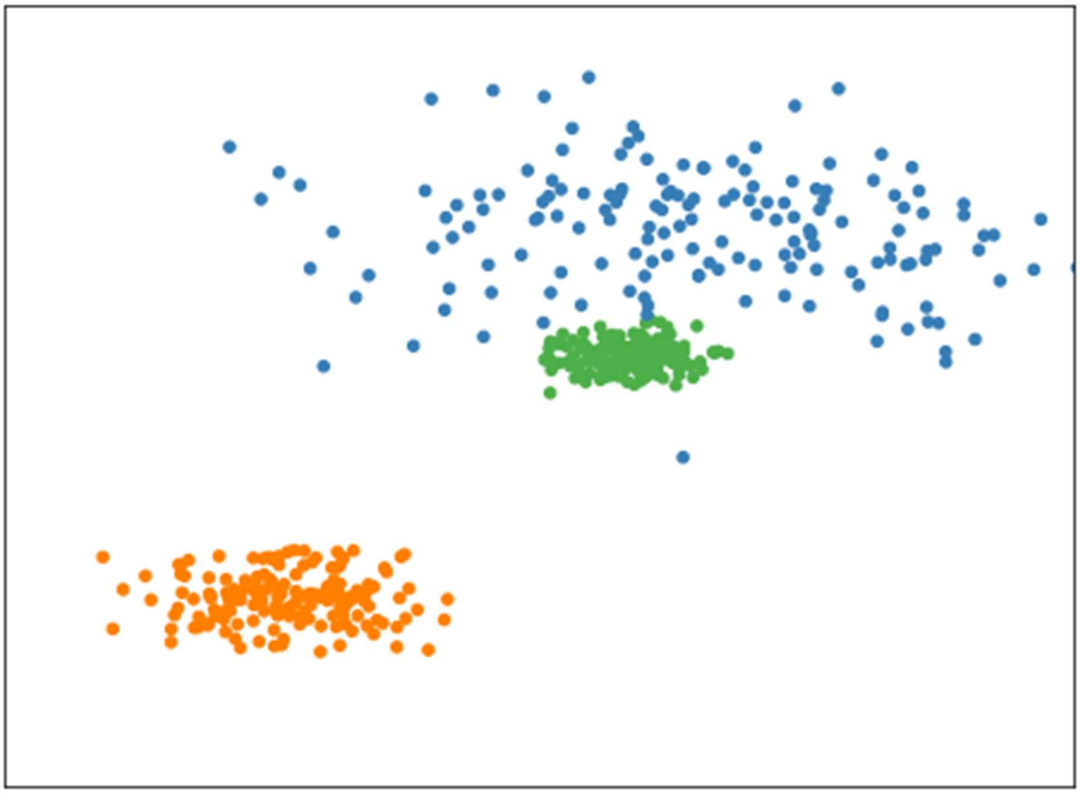


SVM :

```
Trained SVM: sigma = 0.1, C = 0.01: accuracy = 0.10302734375
Trained SVM: sigma = 0.1, C = 0.1: accuracy = 0.10302734375
Trained SVM: sigma = 0.1, C = 1: accuracy = 0.10302734375
Trained SVM: sigma = 0.1, C = 10: accuracy = 0.10302734375
Trained SVM: sigma = 0.1, C = 100: accuracy = 0.10302734375
Trained SVM: sigma = 1, C = 0.01: accuracy = 0.10302734375
Trained SVM: sigma = 1, C = 0.1: accuracy = 0.10302734375
Trained SVM: sigma = 1, C = 1: accuracy = 0.10302734375
Trained SVM: sigma = 1, C = 10: accuracy = 0.10302734375
Trained SVM: sigma = 1, C = 100: accuracy = 0.10302734375
Trained SVM: sigma = 10, C = 0.01: accuracy = 0.10302734375
Trained SVM: sigma = 10, C = 0.1: accuracy = 0.19921875
Trained SVM: sigma = 10, C = 1: accuracy = 0.8125
Trained SVM: sigma = 10, C = 10: accuracy = 0.82421875
Trained SVM: sigma = 10, C = 100: accuracy = 0.82421875
Trained SVM: sigma = 33.24893569946289, C = 0.01: accuracy = 0.395751953
125
Trained SVM: sigma = 33.24893569946289, C = 0.1: accuracy = 0.9086914062
5
Trained SVM: sigma = 33.24893569946289, C = 1: accuracy = 0.9365234375
Trained SVM: sigma = 33.24893569946289, C = 10: accuracy = 0.94555664062
5
Trained SVM: sigma = 33.24893569946289, C = 100: accuracy = 0.9448242187
```

Q . 6.3

Gaussian
Mixture



Here 0,1,2 are clusters

True parameters :

Mean of 0 : [-1.174,-1.288]

Mean of 1 : [0.755 , 1.039]

Mean of 2 : [0.409 , 0.2511]

Variance of 0 :

[[0.1019269 -0.00029818] [-0.00029818 0.02493694]]

```
Variance of 1 :
[[ 0.73527736 -0.04720549] [-0.04720549  0.15766104]]
Variance of 2 :
[[0.03158649 0.00019285] [0.00019285 0.00730099]]


MLE Parameters :
Mean of 0 : [-1.174,-1.288]
Mean of 1 : [0.755 , 1.041]
Mean of 2 : [0.422 , 0.2540]

Variance of 0 :

[[ 0.1019279  -0.00029818][-0.00029818  0.02493794]]

Variance of 1 :
[[ 0.74108496 -0.04699152][-0.04699152  0.15890399]]
Variance of 2 :
[[ 0.03092742  0.00041713] [ 0.00041713  0.00753385]]]

The Obtained parameters and the true parameters are almost same
```