1) a) Likelyhood function $L(\theta; x, z) = P(x, z|\theta)$

Now for a particular z_j

$P(x; z_i|\theta)$

$$= \prod_{i=1}^{n} P(x_i|z_j, \theta)\, P(z_j/\theta)$$

$$= \prod_{i=1}^{n} P(x_i|z_j)\, f_{z_j}(x_i/z_j)\, \tau_{z_j}$$

$$= \prod_{i=1}^{n} \lambda_{z_j} e^{-\lambda_{z_j} x_i}\, \tau_{z_j}$$

So, for $z = \{z_i\}_{i=1 \ldots K}$

$$P(x, z|\theta) = \prod_{j=1}^{K} \prod_{i=1}^{n} \lambda_{z_j} e^{-\lambda_{z_j} x_i}\, \tau_{z_j}$$

given $\theta^{(t)} = (\pi^{(t)}, \lambda^{(t)})$ as current estimate.

1b) $h_{ij}^{(t)} = P(z_i = j \mid x_i, \theta^{(t)})$

$$= \frac{P(z_i = j, x_i \mid \theta^{(t)})}{P(x_i \mid \theta^{(t)})}$$

$$= \frac{P(x_i \mid z_i = j, \theta^{(t)}) \, P(z_i = j \mid \theta^{(t)})}{\sum_{z_i} P(x_i, z_i \mid \theta^{(t)})}$$

$$= \frac{P(x_i \mid z_i = j, \theta^{(t)}) \, P(z_i = j \mid \theta^{(t)})}{\sum P(x_i \mid z_i = j, \theta^{(t)}) \, P(z_i = j \mid \theta^{(t)})}$$

$$= \frac{\lambda_j^t e^{-\lambda_j^t x_i} \pi_j^t}{\sum_{j=1}^{K} \lambda_j^t e^{-\lambda_j^t x_i} \pi_j^t}$$

(c) $Q(\theta, \theta^t) = E_{z|x,\theta^t}[\log L(\theta; x, z)]$

$$= \sum_{i=1}^{n} \sum_{j=1}^{K} p(z=j \mid x_i, \theta^t) \log(\tau_j \lambda_j e^{-\lambda_j x_i})$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{K} \{\log \tau_j + \log(\lambda_j e^{-\lambda_j x_i})\} p(z=j \mid x_i, \theta^t)$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{K} \{\log \tau_j + \log(\lambda_j e^{-\lambda_j x_i})\} \kappa_{ij}^{(t)} \quad [\text{putting value from 1b}]$$

(d) $L = Q(\theta, \theta^t) + \mu\left(1 - \sum_{j=1}^{K} \tau_j\right)$ [Lagrange multiplier used]

$\therefore \frac{\partial L}{\partial \tau_K} = 0$

$\Rightarrow \frac{1}{\tau_K} \sum_{i=1}^{n} p(z=K \mid x_i, \theta^t) + \mu \cdot (-1) = 0.$

$\Rightarrow \mu = \frac{1}{\tau_K} \sum_{i=1}^{n} p(z=K \mid x_i, \theta^t)$ ——— (1)

As we know $\sum_{K} \tau_K = 1$

$\therefore$ From equation 1

$$\tau_K = \frac{1}{\mu} \sum_{i=1}^{n} p(z=K \mid x_i, \theta^t)$$

$$\therefore \sum_{K=1}^{K} \tau_K = \frac{1}{\mu} \sum_{i=1}^{n} \sum_{K=1}^{K} p(z=K \mid x_i, \theta^t)$$

$$\Rightarrow 1 = \frac{1}{\mu} \sum_{i=1}^{n} 1.$$

$$\Rightarrow \mu = n$$

$\therefore$ From equation (2) putting $\mu = n$

$$\Rightarrow \tau_K^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p(z=K \mid x_i, \theta^t)$$

∴ Now,

$$\frac{\partial L}{\partial \lambda_K} = \sum_{i=1}^{n} \frac{1}{\lambda_K e^{-\lambda_K x_i}} \left(e^{-\lambda_K x_i} + \lambda_K (-x_i) e^{-\lambda_K x_i} \right) \left(p(z=K|x_i, \theta^t) \right) = 0$$

$$\Rightarrow \sum_{i=1}^{n} e^{-\lambda_K x_i} = \sum_{i=1}^{n} x_i \lambda_K e^{-\lambda_K x_i}$$

$$\Rightarrow \sum_{i=1}^{n} e^{-\lambda_K x_i} = \lambda_K \sum_{i=1}^{n} x_i e^{-\lambda_K x_i}$$

$$\Rightarrow \lambda_K^{(t+1)} = \frac{\sum_{i=1}^{n} e^{-\lambda_K^t x_i}}{\sum_{i=1}^{n} x_i e^{-\lambda_K^t x_i}}$$

Hence the updated parameter is

$$(\lambda_K^{(t+1)}, \pi_K^{(t+1)}) = (\quad)$$

$$\lambda_K^{t+1} = \frac{\sum_{i=1}^{n} e^{-\lambda_K^{(t)} x_i}}{\sum_{i=1}^{n} x_i e^{-\lambda_K^{(t)} x_i}}$$

$$\pi_K^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} p(z=K|x_i, \theta^t)$$

Q.2a

Network architectures:

For Generator model:

(1) No of Dense Layer used-3 and activation function used is LeackyRelu at each layer. I have used LeackyRelu as activation function because it is free from vanishing gradient problem. Dimension of the used dense layers are 256, 512, 1024 respectively

(2) Then used batchnormalization to train my model in more stable distribution of input.

(3) Input shape used is (1,2,1) and latent dimension=6

(4) Then at the output layer I have used "tanh" as activation function.

For Discriminator Model:

(1) First layer is flattened input layer. Input shape is (1,2,1).

(2) No of Dense layer used -2 and activation function used is LeackyRelu.
Dimension of the dense layers are 512 , 256 respectively.

(3) Then at the output layer sigmoid function has been used as activation function..

For GAN Model:

Then Gan model has been created by combining both generator and discriminator model. Optimizer used to train the Gan model is "Adam" with parameter (0.0002, 0.5)

loss functions:

To calculate the loss, I have used "binary_crossentropy" as loss function because the discriminator just need to classify whether the image is real or fake i.e binary prediction. Discriminator loss obtained after training is 0.690711 and generator loss is 0.744336 and accuracy of the discriminator model is 49.22%

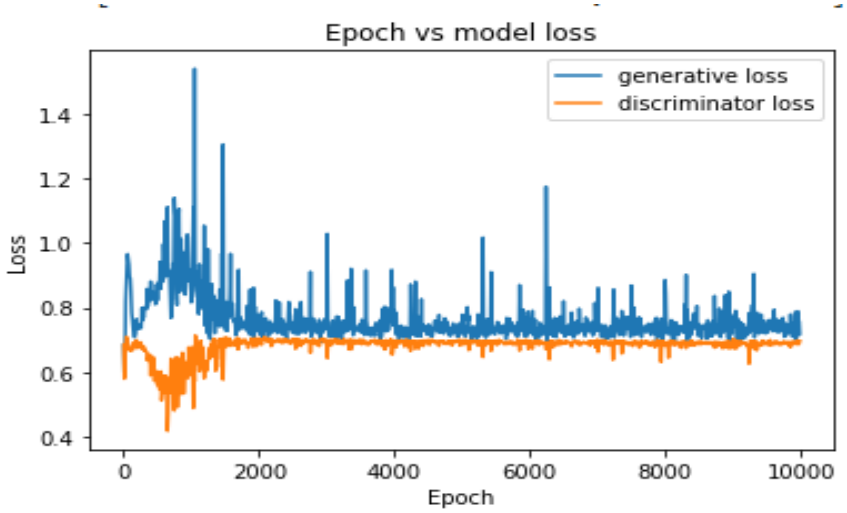
Hyperparameters:

To train the model I have used 10000 epochs , batch size = 128 and latent dimension 6

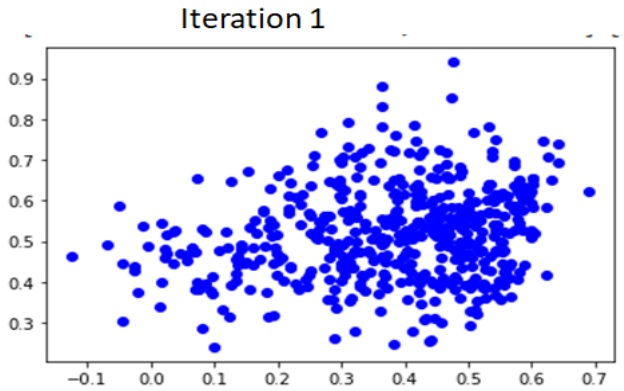
Preprocessing :

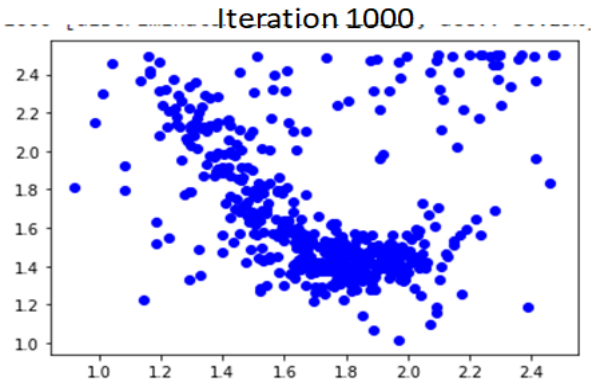
To normalize the input data , it has been divided by 3 .

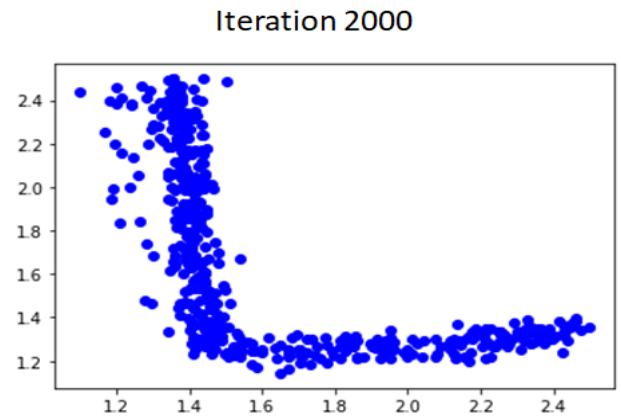
Plots depicting the generator and discriminator losses:

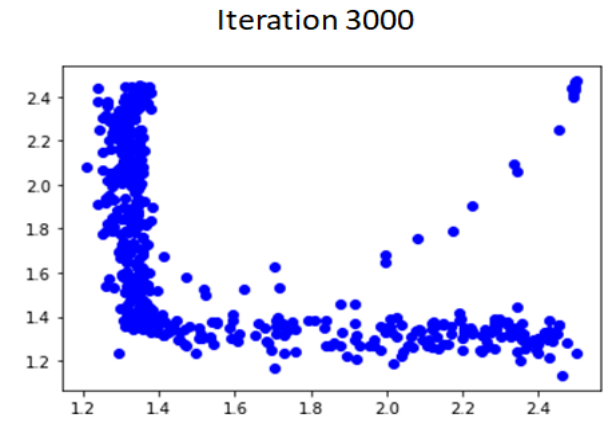


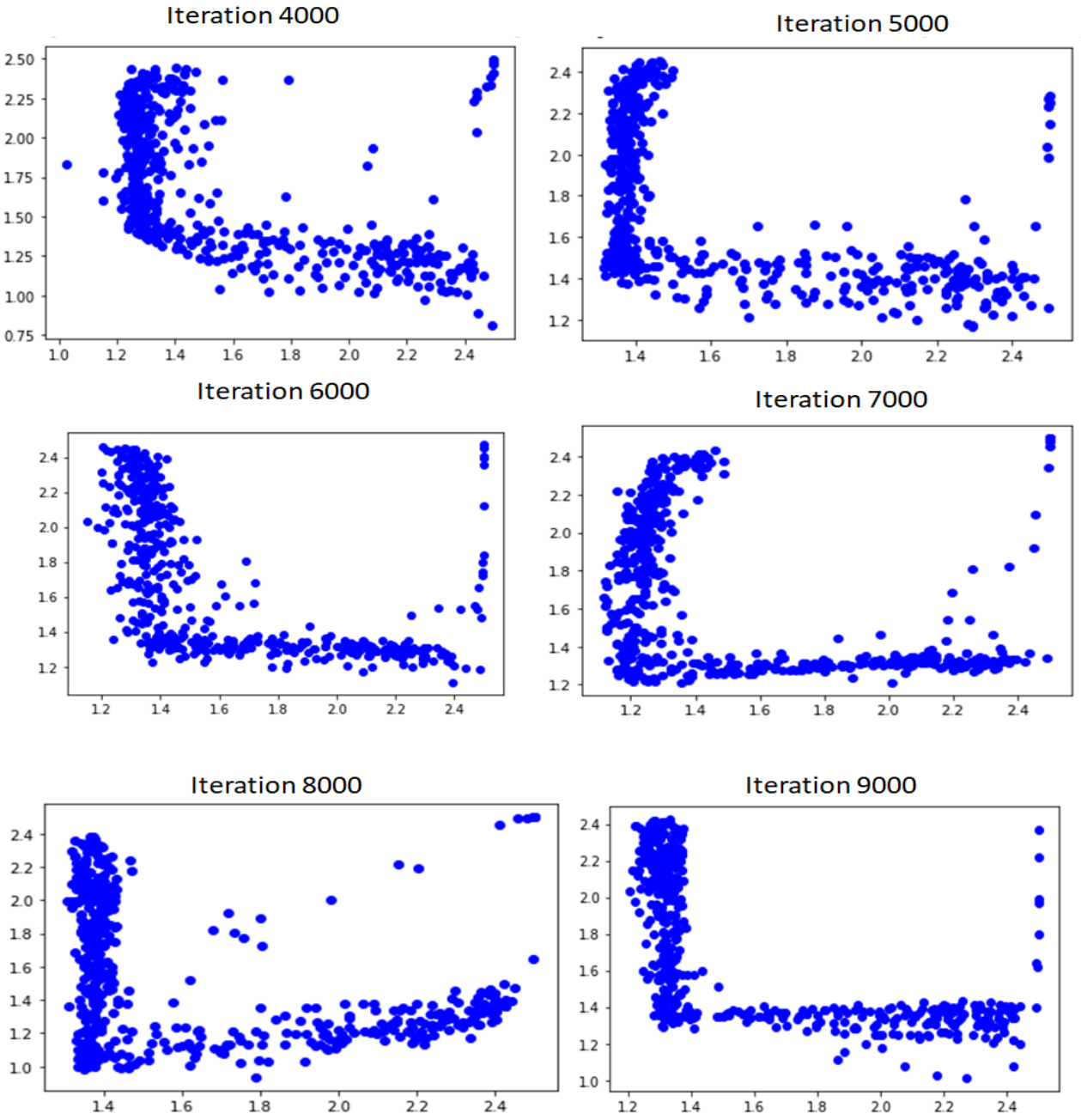
Ten visualizations generated at regular intervals:










Iteration 4000
Iteration 5000
Iteration 6000
Iteration 7000
Iteration 8000
Iteration 9000

Q.2b

Network architectures:

For Generator model:

(1) No of Dense Layer used-3 and activation function used is LeackyRelu at each layer. I have used LeackyRelu as activation function because it is free from vanishing gradient problem. Dimension of the used dense layers are 256, 512, 1024 respectively.

(2) Then used batchnormalization to train my model in more stable distribution of input.

(3) Input shape used is (28,28,1) and latent dimension=128

(4) Then at the output layer I have used "tanh" as activation function.

For Discriminator Model:

(1) First layer is flattened input layer. Input shape is (28,28,1).

(2) No of Dense layer used -2 and activation function used is LeackyRelu.
Dimension of the dense layers are 512 , 256 respectively.

(3) Then at the output layer sigmoid function has been used as activation function.

For GAN Model:

Then Gan model has been created by combining both generator and discriminator model. Optimizer used to train the Gan model is "Adam" with parameter (0.0002, 0.5)

Loss functions:

Loss function used is "binary_crossentropy". discriminator loss obtained after training is 0.669565 and generator loss is 0.846462 .

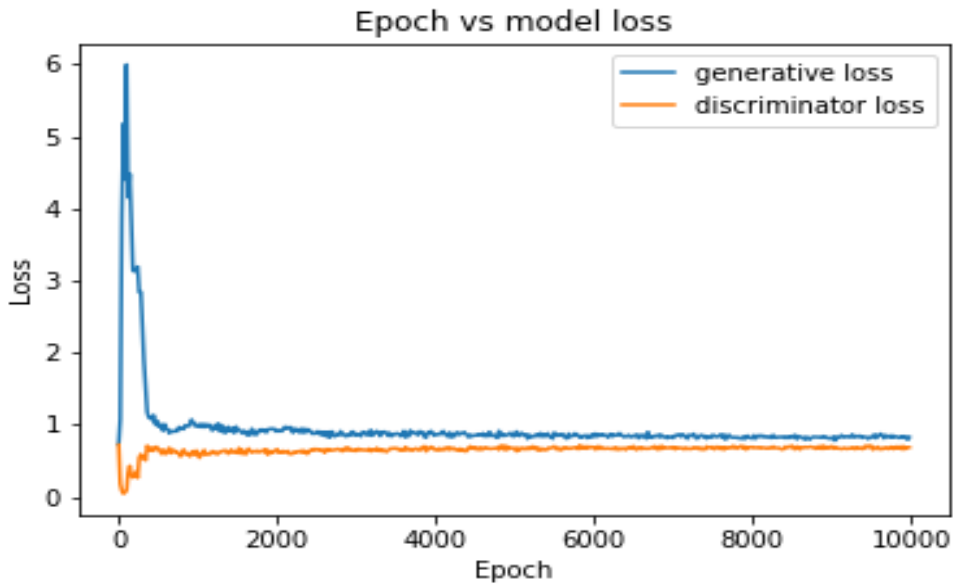
Hyperparameters:

To train the model I have used 10000 epochs , batch size = 128 and latent dimension 100

Data Preprocessing :

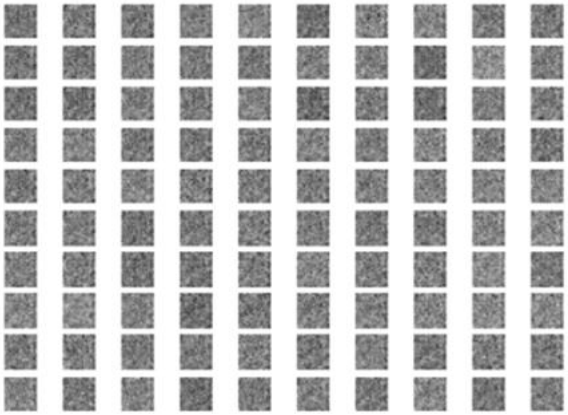
To normalize the input data ,it has been divided by 255 , so that the value is normalized between 0 to 1.

Plots depicting the generator and discriminator losses:



Ten visualizations generated at regular intervals:

Epoch 1



Epoch 1000

Epoch 2000

Epoch 3000

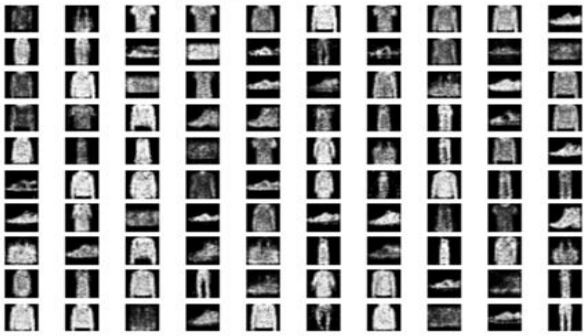
Epoch 4000

Epoch 5000

Epoch 6000

Epoch 7000

Epoch 8000



Epoch 9000