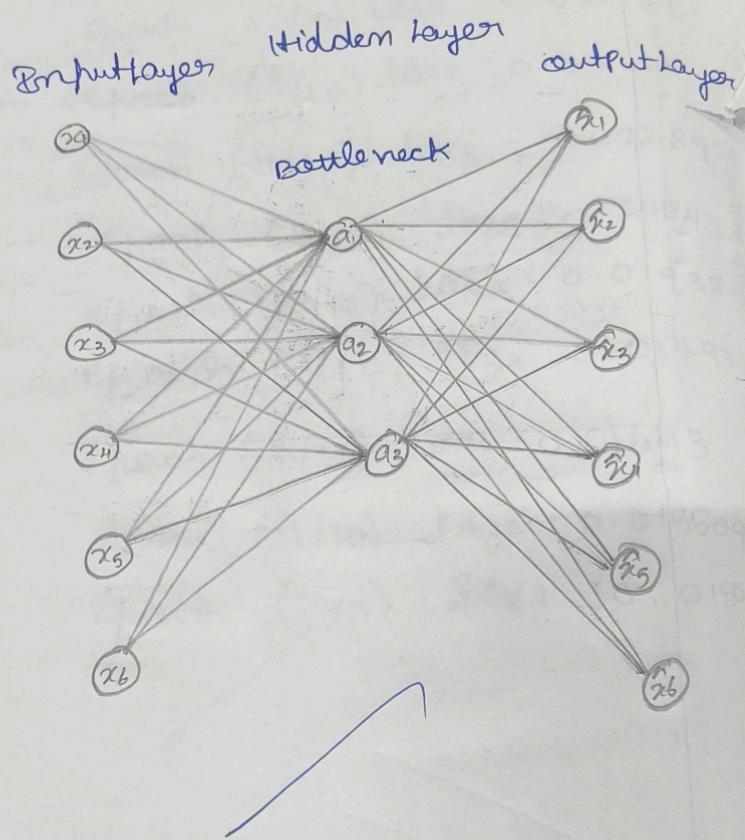


## Architecture :-



thus

more information in bottleneck  
better performance

## INVENT

### II. Experiments Using Variational Autoencoders

#### Aim:-

To implement and analyze a variational Autoencoder (VAE)

#### Objective:

- \* To understand the concept and working of variational Autoencoders (VAEs)
- \* To perform unsupervised feature learnings using probabilistic latent space representation
- \* To train a VAE model that learns both encoding and decoding
- \* To visualize the latent space and generate new images by sampling from it

#### PSEUDOCODE:-

BEGIN

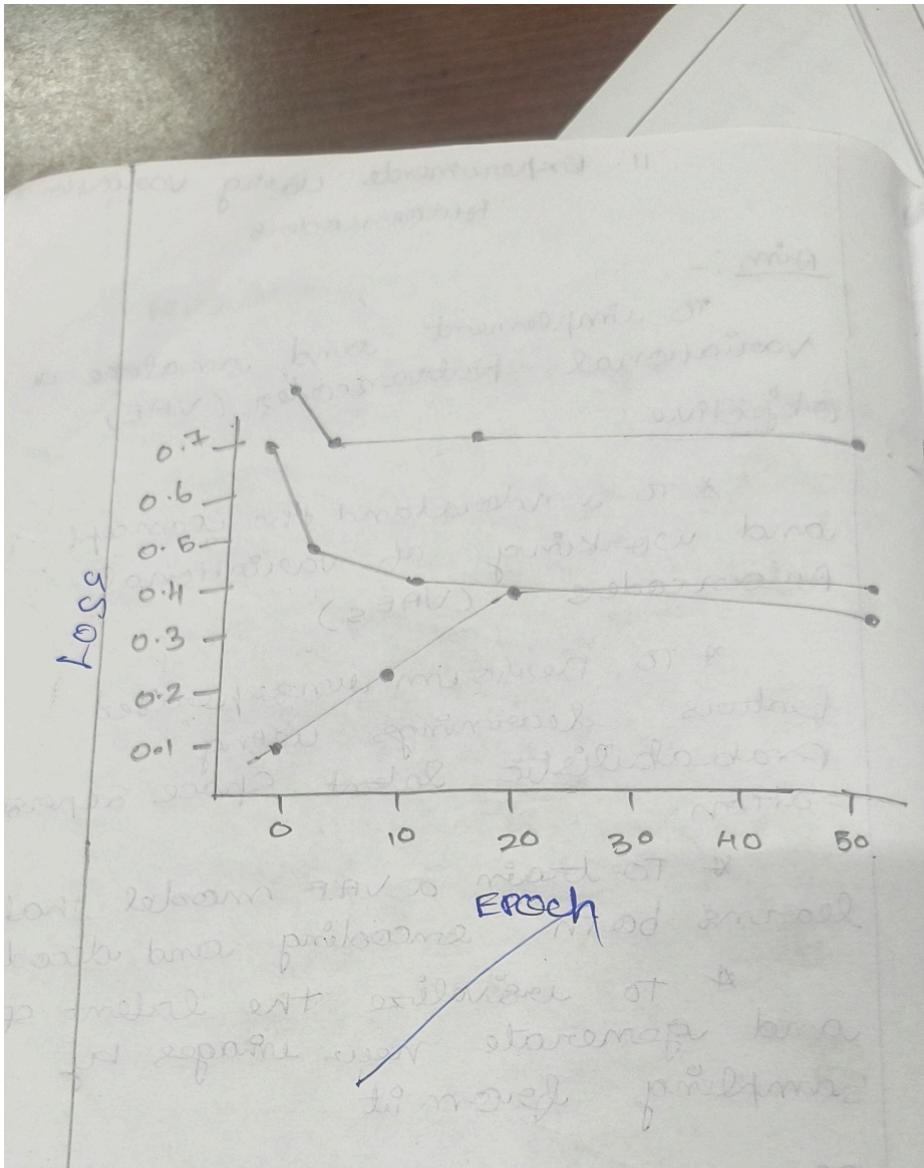
Import torch , torchvision  
load MNIST dataset and Normalize  
images to  $[0, 1]$

Define VAE model.

Encoder:  $784 \rightarrow 400 \rightarrow$   
(mean  $\mu$ : log-variance  $\sigma^2$ )

Reparameterize :  $z = \mu + \sigma * \epsilon$  ( $\epsilon \sim N(0, 1)$ )

Decoder :  $z - 400 \rightarrow 784$  (Sigmoid &  $\epsilon$ )



RESULTS

### RESULTS

initialized, saved trained

different type features TRAIN base  
2000 at epoch

below are images

← only HGT recovered

(↑ can see pale + main)

→ or HGT, watermark kept

→ biomed) HGT ← OOH → ↓ : recovered

## VENT

refine loss:

total loss = reconstruction  
loss + KL Divergence use Adam  
optimizer for training

for each epoch:

For each batch:  
forward pass → encode sample,  
decode.  
compute loss → back propagate →  
update weights

After trainings

reconstruct test images and  
generate new samples from latent  
space

visualize original, reconstructed  
and generated images

END

## OBSERVATION

\* The VAE learned smooth  
latent space allowing continuous  
interpolation between different  
digits

\* Generated samples  
resemble realistic handwritten  
digits, though slightly blurry  
due to probabilistic sampling

\* The KL Divergence term  
regularized the latent space,  
ensuring meaningful representation

\* Reconstruction loss decreased  
gradually, indicating successful learning

MENT

Kuberman

FLA  
04/04/2010

mark: 'AI'  
'A'

convolutional - Generative models  
complex color images

To implement a variational autoencoder  
loss to generate complex  
color images

Approach

• build a generator to  
create new images from  
random noise

• build a discriminator  
to distinguish between  
real images

• train the generator and  
discriminator alternately

• Generate and evaluate  
realistic results

Result:-

Q. Therefore the implementation  
and analyse of variational  
Autoencoder is successfully  
completed.