CS F437: Generative Artificial Intelligence
Assignment 1

Agenda: Image Reconstruction

In this assignment, we are expected to perform image reconstruction by employing the following Latent Variable Models (LVMs) covered in class:

- Principal Component Analysis (PCA)
- Probabilistic Principal Component Analysis (PPCA)
- Variational Autoencoders (VAEs)

Each of these models projects the input image down to the latent vector space, and tries to reconstruct it back to the original image while minimizing the loss of information.

Part A: Principal Component Analysis (PCA)

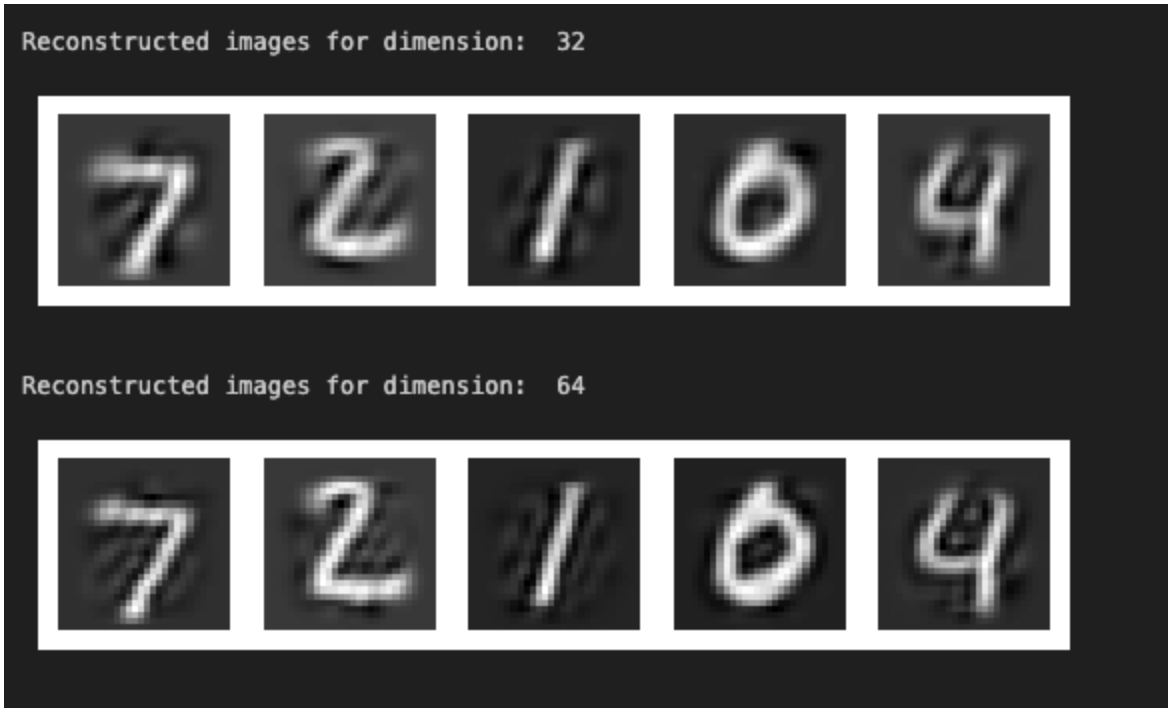
In PCA, we take the eigenvectors of the covariance matrix that capture the maximum variance. We first find the eigenvalues and sort it in descending order. We then pick the first M eigenvectors from D to represent the data in M dimensions instead of D. We then multiply the original data with the eigenvectors to transform it into that M feature space. For image reconstruction, we use this new low dimensional data and multiply the transpose to get the image in the original feature space D. We run this for all 2,4,8,16,32,64 dimensions.

Reconstructed images for dimension: 2

Reconstructed images for dimension: 4

Reconstructed images for dimension: 8

Reconstructed images for dimension: 16



Part B: Probabilistic Principal Component Analysis (PPCA)

In PPCA, we try to find the probability distribution of the latent space given a sample point in X. Then we sample a point from that distribution and try to find a probability distribution for X. We then sample a point from this distribution and use that as the reconstructed image.

$$p(\mathbf{x}) = \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\,\mathrm{d}\mathbf{z}.$$

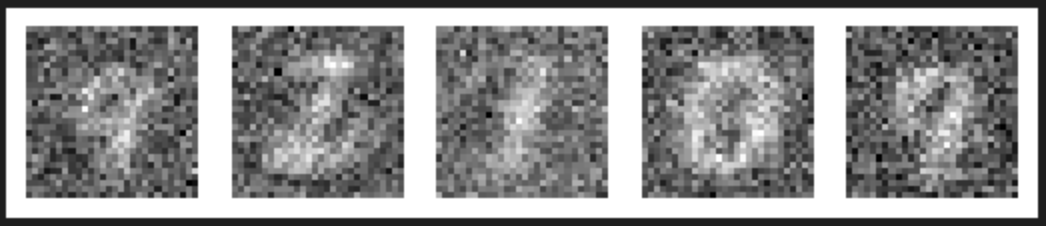
$$p(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}|\mathbf{M}^{-1}\mathbf{W}^{\mathrm{T}}(\mathbf{x}-\boldsymbol{\mu}), \sigma^{-2}\mathbf{M}\right).$$

$$\mathbf{M} = \mathbf{W}^{\mathrm{T}}\mathbf{W} + \sigma^2\mathbf{I}.$$

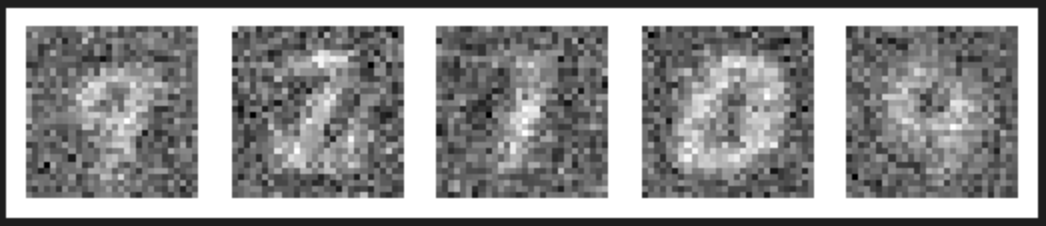
$$\mathbf{W}_{\mathrm{ML}} = \mathbf{U}_M(\mathbf{L}_M - \sigma^2\mathbf{I})^{1/2}\mathbf{R}$$

$$\sigma^2_{\mathrm{ML}} = \frac{1}{D-M}\sum_{i=M+1}^{D}\lambda_i$$

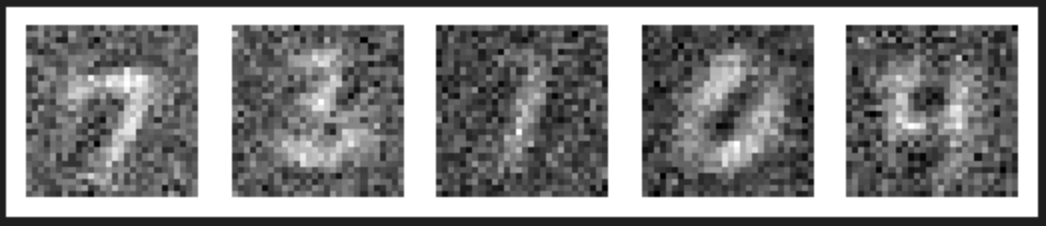
Reconstructed images for dimension: 2



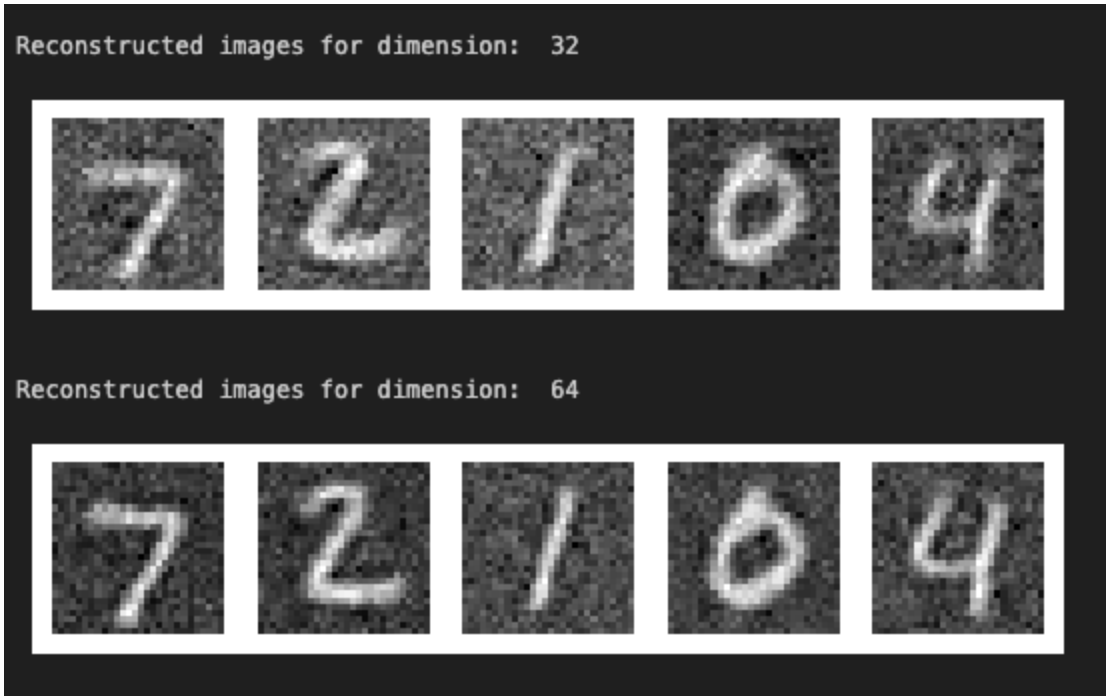
Reconstructed images for dimension: 4



Reconstructed images for dimension: 8

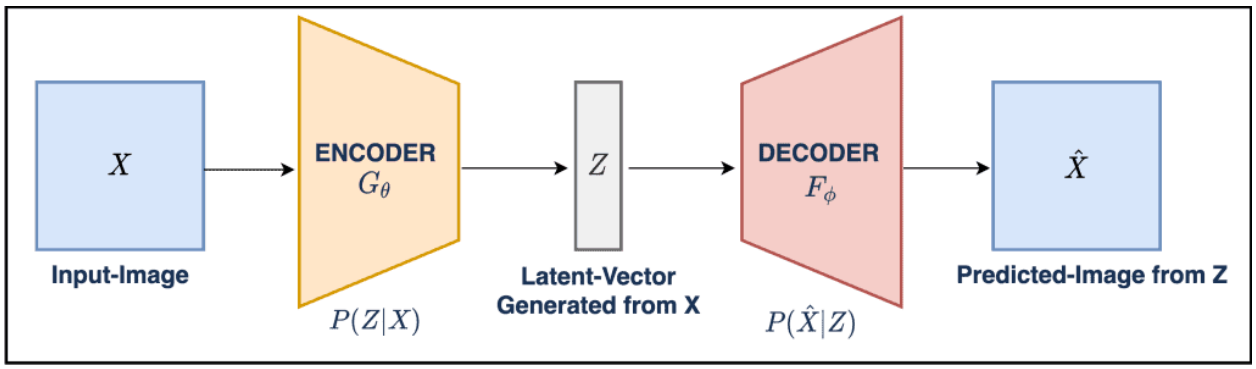


Reconstructed images for dimension: 16



Part C: Variational Autoencoder

A variational autoencoder (VAE) is a generative AI algorithm that uses deep learning to generate new content, detect anomalies and remove noise.
The encoder architecture can be represented by the following:



The encoder and decoder here are both neural networks, responsible for scaling the information to and from the latent variable vector Z.

The dataset used for image classification is the MNIST dataset, and Neural Networks (NNs) have been used as the encoder/decoder with “ReLU” as the

activation function for all layers except the decoder output layer, which uses the sigmoid function. *Mean Square Error* (*MSE*) and the *Adam optimizer* are the loss function and optimizing algorithm of choice, respectively.

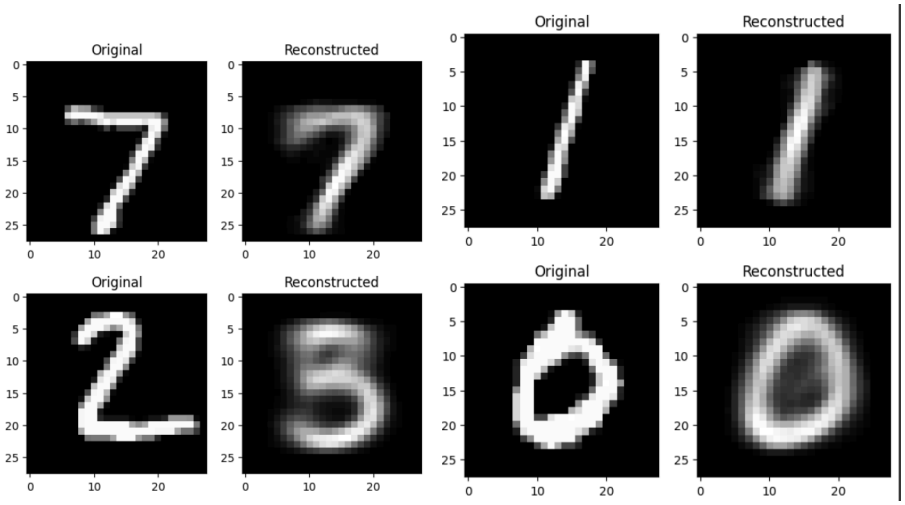
Firstly, since every (*28x28*) image sample of the dataset is represented by a linear vector (*length 784*) of its pixel values ranging between *0 and 255,*
We preprocessed this dataset by reshaping every image into a 28,28 array of its pixels, with each pixel value standardized to a value between 0 and 1, by dividing it by 255.

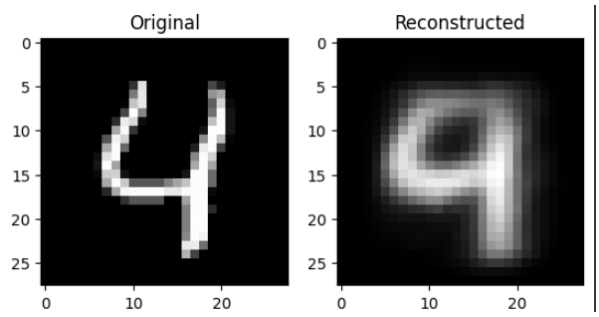
Next, we use *Tensorflow* and *Keras* libraries to define the NNs composing the encoder and decoder networks. Each of them has an input layer, 4 hidden layers and an output layer for the transformation of the variables.

We create 6 models of VAE, train them for 10 epochs, and analyse their behaviour by comparing their MSEs as reported below:

1. **Model 1 (2 dimensional latent variable)**

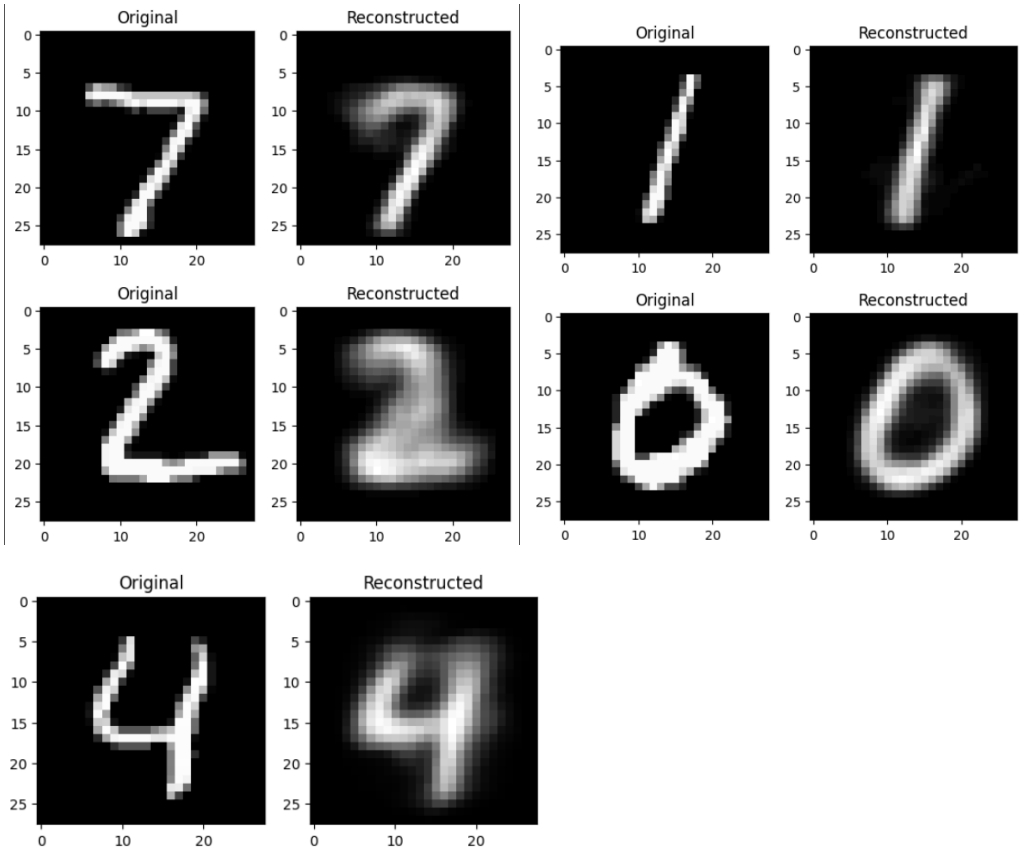
   As the title suggests, this model is defined to have its latent vector Z of length 2. After training, we test its generating power by inputting the *first 5 samples of the testing set* and plotting the results for a visual representation-





MSE: 0.0443215

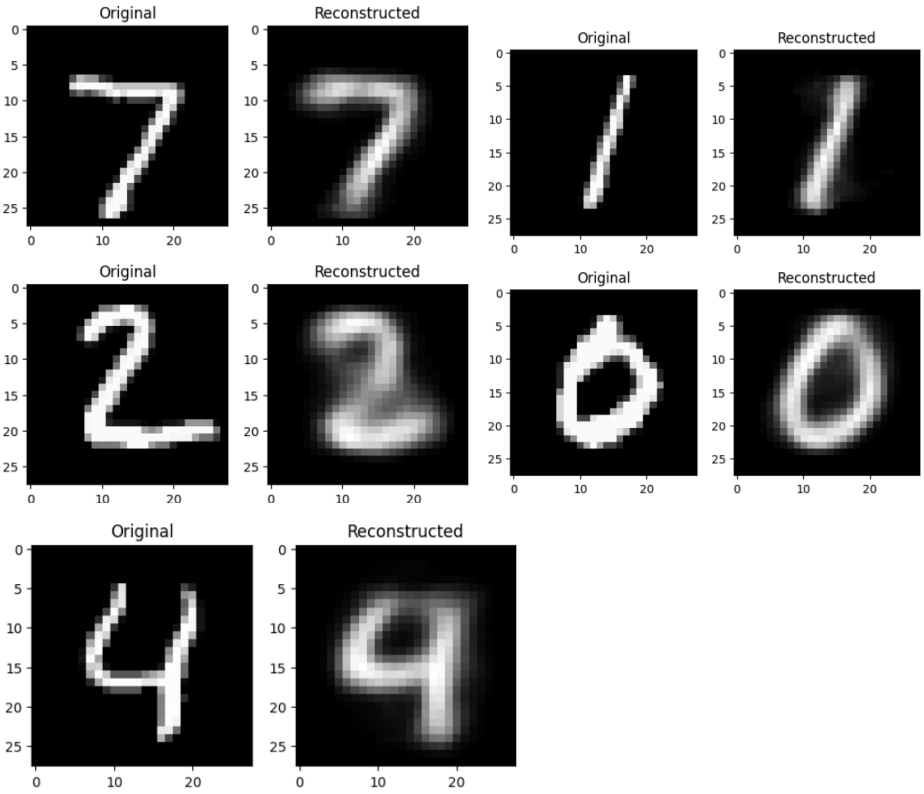
2. **Model 2 (4 dimensional latent variable)**
   This model is defined to have its latent vector Z of length 4. After training, we test its generating power by inputting the *first 5 samples of the testing set* and plotting the results for a visual representation-



MSE: 0.037901234

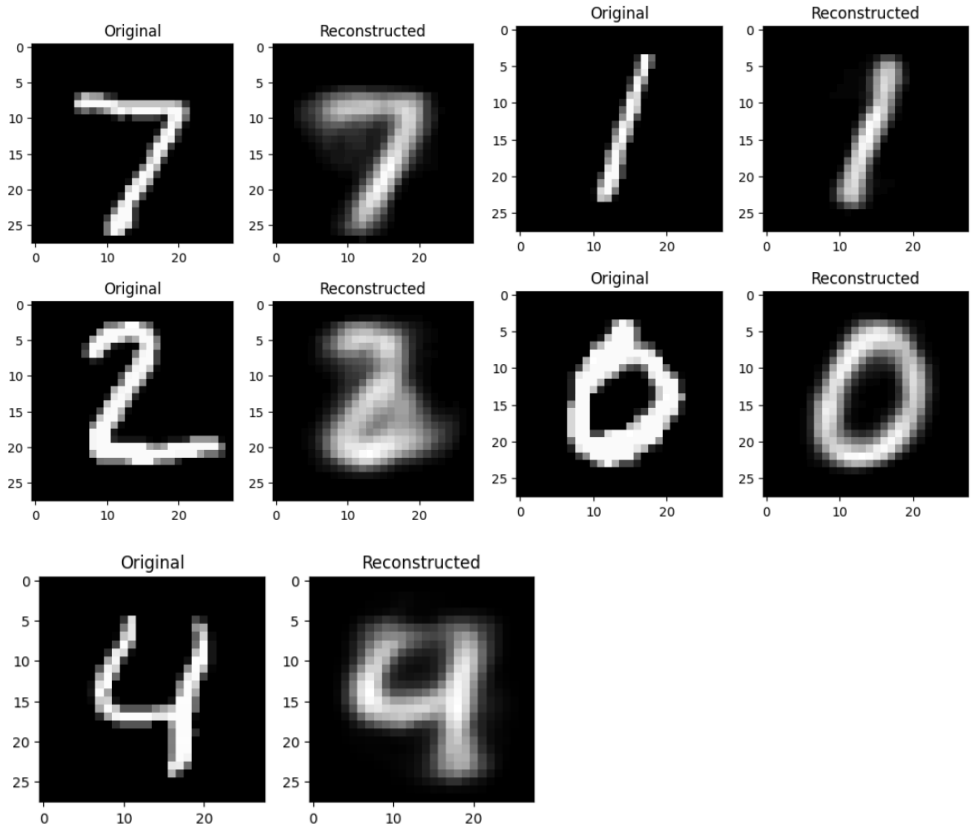
3. **Model 3 (8 dimensional latent variable)**
   This model is defined to have its latent vector Z of length 8. After

training, we test its generating power by inputting the *first 5 samples of the testing set* and plotting the results for a visual representation-



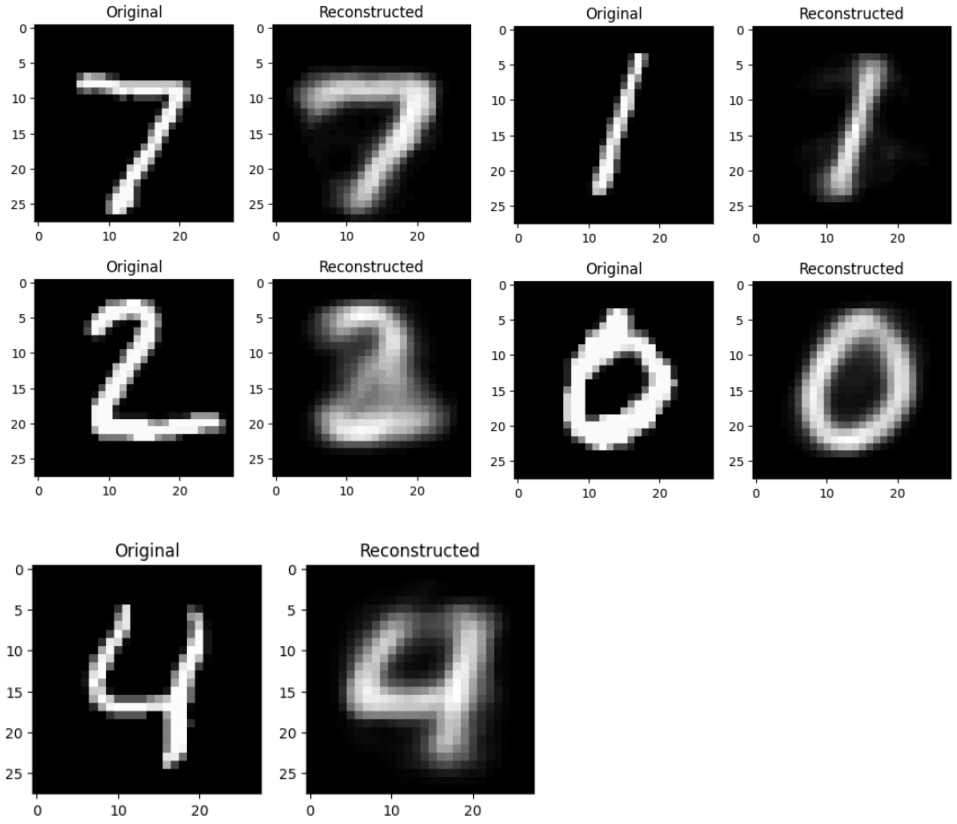
MSE: 0.0372155

4. **Model 4 (16 dimensional latent variable)**
   This model is defined to have its latent vector Z of length 16. After training, we test its generating power by inputting the *first 5 samples of the testing set* and plotting the results for a visual representation-



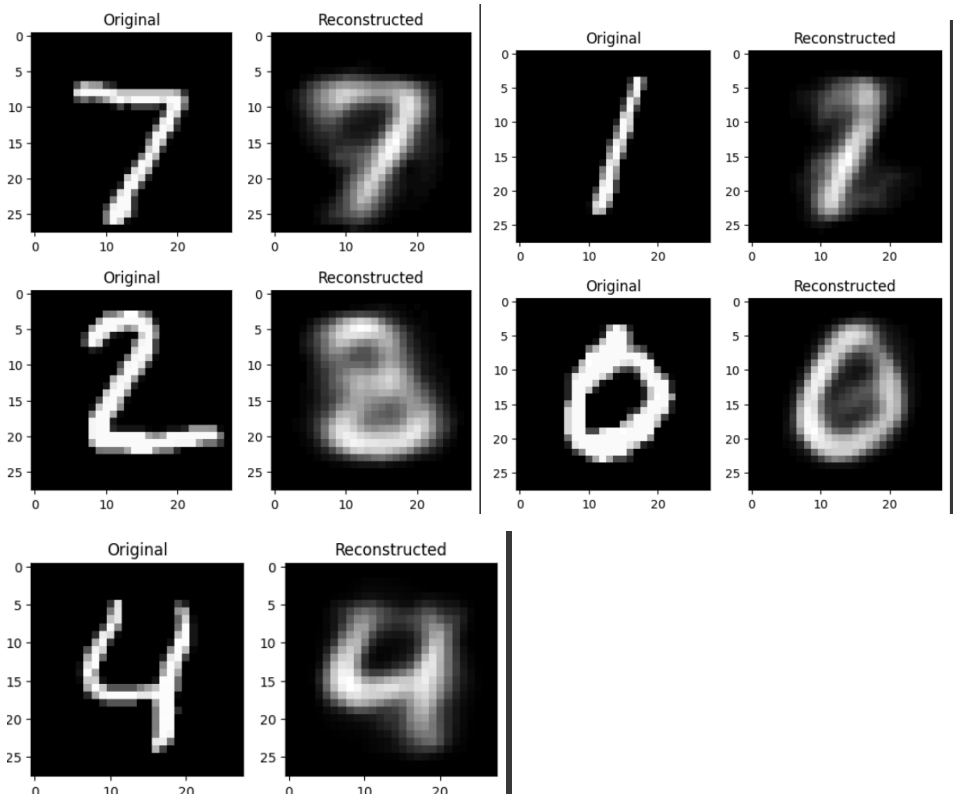
MSE: 0.03665639

5. **Model 5 (32 dimensional latent variable)**
   This model is defined to have its latent vector Z of length 32. After training, we test its generating power by inputting the *first 5 samples of the testing set* and plotting the results for a visual representation-



MSE: 0.03855335

6. **Model 6 (64 dimensional latent variable)**
   This model is defined to have its latent vector Z of length 64. After training, we test its generating power by inputting the *first 5 samples of the testing set* and plotting the results for a visual representation-



MSE: 0.041515283

Part D: Graph Plotting