Image Reconstruction

The image reconstruction task involves projecting an image into a latent vector space and reconstructing the image back from this vector space, minimizing the loss of information. The objective of this assignment is to build image reconstruction models using the following three algorithms and analyze the differences in their reconstruction abilities:

- Principal Component Analysis
- Probabilistic Principal Component Analysis
- Variational Autoencoders

Dataset

The dataset for this assignment is the 'MNIST' dataset which consists of images of handwritten digits from 0 to 9. The dataset consists of 60,000 training images and 10,000 testing images.

- This dataset is available on the internet, or you can also use any library that has the MNIST dataset to load it.
- Do not load the labels associated with the images. We will only be working with the images and not their labels.

Part A: Principal Component Analysis

The objective is to build image reconstruction models using Principal Component Analysis. You will be required to build the models with the following dimensions of latent variables: 2, 4, 8, 16, 32, 64, thus giving 6 PCA models in total.

What needs to be documented?

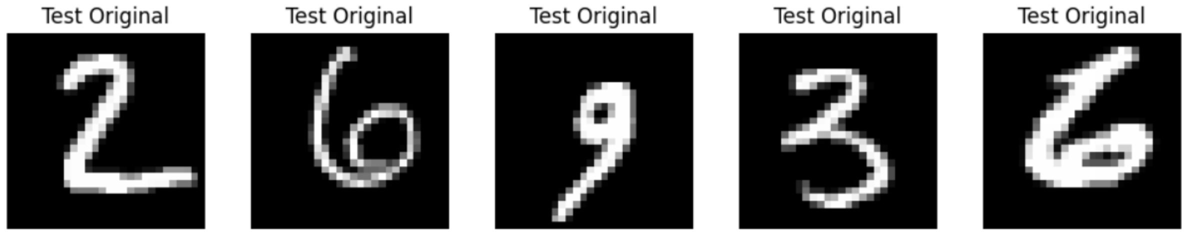
1. In the report, put 5 input images selected at random from the test dataset and for all 6 models, put the corresponding reconstructed images given by them.

NOTE:

1. The selected images need not be the same as the images given in the below example.
2. The outputs given in the examples are only for sample. In all cases, the actual outputs may not exactly match the outputs in the examples.

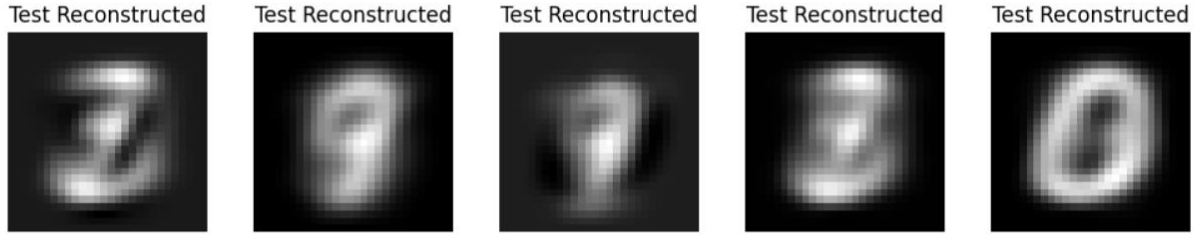
Eg: -

Input images:

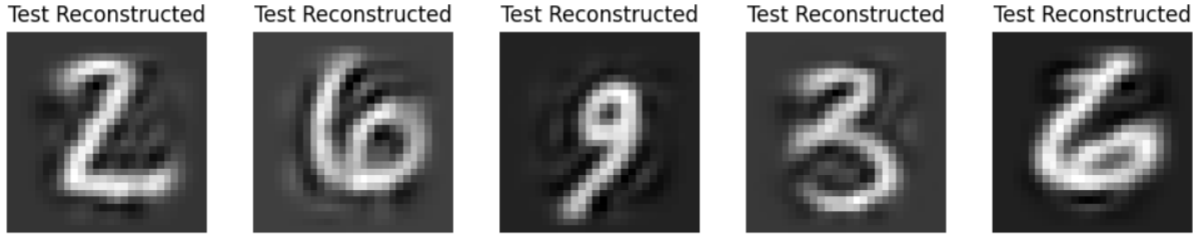


Outputs:

Latent variable dimension 2:



Latent variable dimension 64:



Similarly, put the output for all latent variables in the report.

2. Calculate the mean squared error (MSE) on the test dataset for all 6 models and note down the values in the report.

Formula:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

y_i -> True image vector/matrix

$\hat{y}_i$ -> Predicted image vector/matrix

$(y_i - \hat{y}_i)^2$ -> Mean of the squared error of each component of the vector/matrix

N -> No. of test images

Implementation Requirements:

1. Scikit-learn can be used for PCA implementation.
2. Apart from the standard libraries, no other library should be used for calculating the MSE metric value.

Part B: Probabilistic Principal Component Analysis

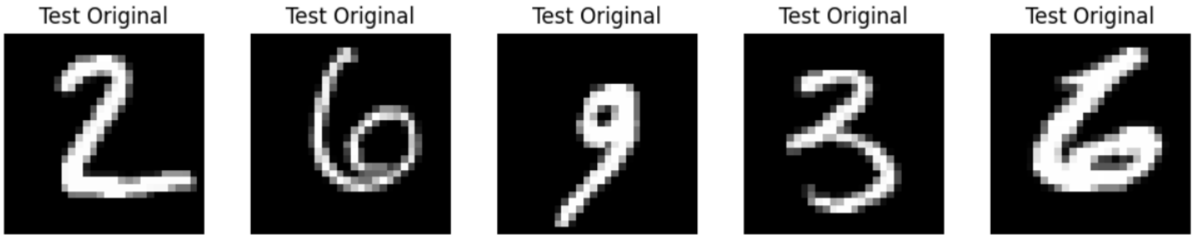
The goal is to build image reconstruction models using the Probabilistic Principal Component Analysis method taught in class. You will be required to build the models with the following dimensions of latent variables: 2, 4, 8, 16, 32, 64, thus giving 6 Probabilistic PCA models in total.

What needs to be documented?

1. In the report, put the same 5 input images selected for Part A from the test dataset and for all 6 models, put the corresponding reconstructed images given by them.

Eg: -

Input images:



Outputs:

Latent variable dimension 2:

Test Reconstructed
Test Reconstructed
Test Reconstructed
Test Reconstructed
Test Reconstructed

Latent variable dimension 64:

Test Reconstructed
Test Reconstructed
Test Reconstructed
Test Reconstructed
Test Reconstructed

Similarly, put the output for all latent variables in the report.

2. Calculate the mean squared error (MSE) on the test dataset for all 6 models and note down the values in the report. Use the same formula mentioned for MSE in Part A.

Implementation Requirements:

1. Apart from the standard libraries, no other library should be used for implementing Probabilistic PCA.
2. Apart from the standard libraries, no other library should be used for calculating the MSE metric value.

Part C: Variational Autoencoders

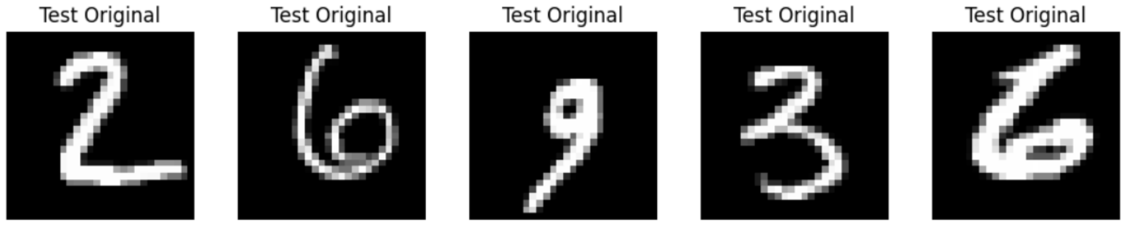
The goal is to build image reconstruction models using Variational Autoencoders. You will be required to build the models with the following dimensions of latent variables: 2, 4, 8, 16, 32, 64, thus giving 6 VAE models in total.

What needs to be documented?

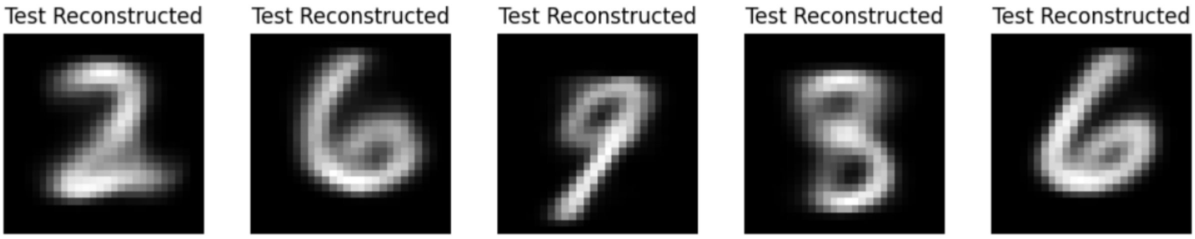
1. In the report, put the same 5 input images selected for Part-A from the test dataset and for all 6 models, put the corresponding reconstructed images given by the model.

Eg: -

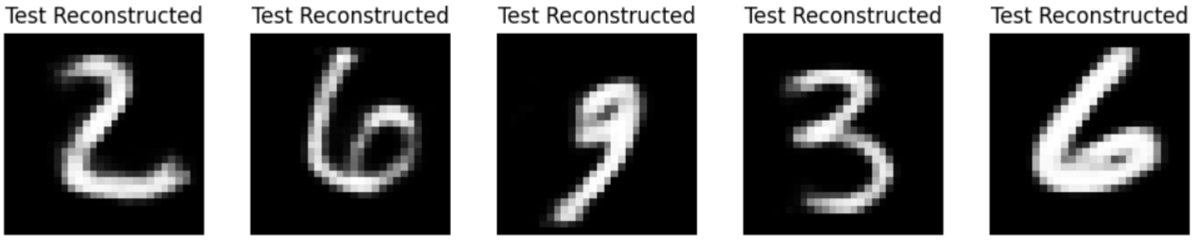
Input images:



Latent variable dimension 2:



Latent variable dimension 64:



Similarly, put the output for all latent variables in the report.

2. Calculate the mean squared error (MSE) on the test dataset for all 6 models and note down the values in the report. Use the same formula mentioned for MSE in Part A.

Implementation Requirements:

1. The following libraries are permitted: Scikitlearn, Tensorflow, Keras, and PyTorch -> These libraries should only be used to define the network layers and train the model. No pre-existing VAE implementation should be used.
2. The encoder and decoder network architecture is up to the student. Both Fully-connected and CNN layers can be used.
3. The activation function is from any of the following functions: tanh, sigmoid, ReLU.
4. The optimization algorithm is from any of the following: Stochastic Gradient Descent, Adam optimizer.
5. Apart from the standard libraries, no other library should be used for calculating the MSE metric value.

Part D: Graph Plotting

Plot a line graph using Matplotlib with the following axes:

- X-axis: latent variable dimension
- Y-axis: Mean Squared Error