# Report:
# Computer Vision

The implementation of the Variational Autoencoder consists of:
Encoder - Linear layer, ReLU, 2 separate linear layers for the mean and log variance.
Sampling - Latent space is sampled using the normal distribution with the learnt mu and logvar.
Decoder - Linear, ReLU, Linear, Sigmoid.

The total loss used for the training consists of 2 parts: the reconstruction loss and the KL divergence loss. The optimizer used is Adam with default parameters.
I imported the MNIST dataset, converted toTensor, then defined the Trainer class, which fits the model to the given dataset as well as generates a sample from the latent space for qualitative testing.
The model specifications include input_size 786 (number of pixels), hidden_size 392, latent_size 100. The model is trained for 100 epochs for every run, and 5 images are generated from sampling the latent space per experiment. I used the Binary CrossEntropy loss for optimizing the model.
I have uploaded 5 images for each of the following runs for reference.
Qualitative evaluation of generation of samples depends on factors such as clarity, diversity and correspondence to the actual dataset.
After sampling from the standard normal distribution, the latent space is able to capture most of the handwritten character data, but there is a good amount of noise present in the generated image, making it blurry.
On the other hand, the Gaussian(1,2) distribution makes the generation noisier in the sense that some of the generated images do not correspond to any of the actual digits (realism is slightly lost). Both these techniques offer ample diversity in generation of individual digits.

I tried implementing the beta distribution for sampling. For values of alpha and beta, I took a reference value 0.25 for both alpha and beta, giving me a mean of 0.5 and variance 0.2. The reason behind picking such a value is that the mean and variance of the beta distribution are restricted to [0,1] and the highest practical value of variance for the distribution is 0.25. Thus, I felt that the choice of the parameters was appropriate as there exists no standard distribution.
I experimented with the loss function and the forward method of the model to improve the performance of the model.
The model with the Binary CrossEntropy Loss function provides an unclear, noisy generation of digits. Also, correspondence to the digits in the dataset is worse compared to the previous runs.
The model with the Mean Square Error Loss function provides a sharper digit compared to the BCE. However, the issue of realism with the BCE Beta is present in the MSE model as well.
I then tried to change the forward function of the model. The function now works as follows: it learns the mean and log variance for the training data, but instead of sampling the latent space with these values and using the vector for decoding, it samples and the mean and variance from

a normal distribution with means being the values to be learnt and variance 1. Then, the latent space is sampled with these sampled parameters and used for decoding. However, on implementing it, I noticed that the image generation becomes extremely unclear as noise is added to the forward function deliberately using the sampling technique.
All these methods lack in creating diversity in the generation of digits. I realized that the number 3 was being generated more often compared to the others.

I will be trying out the bonus section of the assignment later after the submission is done. This was a very good learning experience for me.

P.S. I talked to Aditya about my current implementation and have gotten an additional week to finish the task as I was not aware of the correction in the assignment.