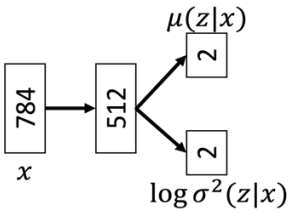
Assignment 1 – VAE

Due Date: Oct 14, 2022 Midnight

In this assignment we will implement a Variational Autoencoder (VAE) to generate data from the MNIST dataset. MNIST is a popular digit dataset that is available with most of the deep learning frameworks like PyTorch, TensorFlow, etc. Or, feel free to use the `digits.py` file which can download MNIST. It consists of 60,000 grayscale images, each 28x28 pixels. The dataset has 10 categories of digits, {0,1,…,9}. Normalize the pixel values to [0,1]. Use the entire dataset of 60,000 images to train the VAE. The architecture of the neural network is as follows:

1. The encoder network $q(z|x)$ is modeled by a branched 2 layer fully-connected neural network. One branch estimates $\mu(z|x)$ and the other branch estimates $\log \sigma^2(z|x)$. We estimate $\log \sigma^2(z|x)$ instead of $\sigma(z|x)$ because it is easier to work with $\log \sigma^2(z|x)$. The architecture of the encoder is:



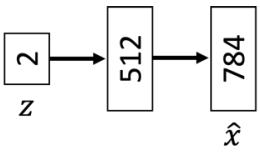
   where the latent space z is of 2 dimensions. 784 is the dimension of the input image x, which is the vectorized MNIST image (28x28 = 784). The hidden layer is of dimension 512 with ReLU activations and the dimension of the latent space z is 2.

2. $p(z)$ is a unit norm Gaussian distribution $N(0, I)$ of 2 dimensions. We will estimate the $KL(q(z|x)||p(z)) = KL(N(\mu(z|x), \sigma(z|x)) \,||\, N(0, I))$, which is given by

$$KL(q(z|x)||p(z)) = -\frac{1}{2}\sum_{k=1}^{K=2} 1 + \log \sigma_k^2(z|x) - \mu_k^2(z|x) - \sigma_k^2(z|x)$$

   We estimate $\log \sigma^2(z|x)$ instead of $\sigma(z|x)$ because it is used in the loss function and it is straightforward to estimate $\sigma^2(z|x)$ from $\log \sigma^2(z|x)$ as well.

3. The decoder neural network $p(x|z)$ has the following architecture:

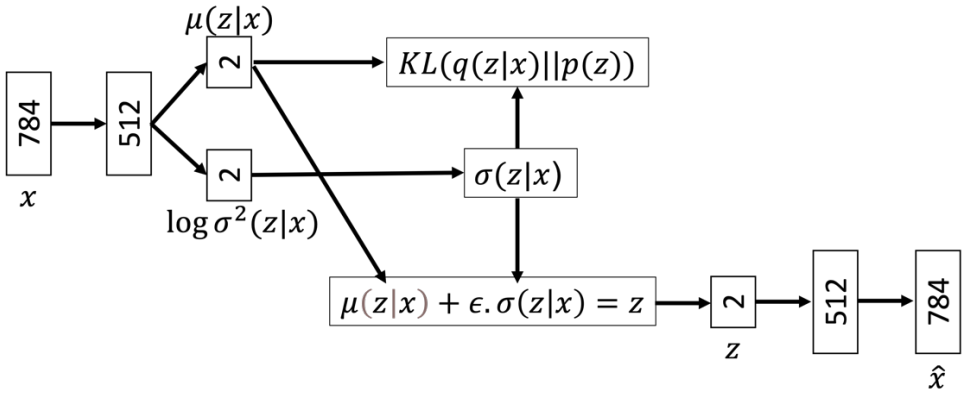


   The hidden layer has ReLU activation and the final layer has Sigmoid activation to normalize the decoded image between [0,1]. Use a binary cross-entropy loss to compare input x with the decoded image $\hat{x}$.

4. We will use the reparameterization trick to sample z as input to the decoder.

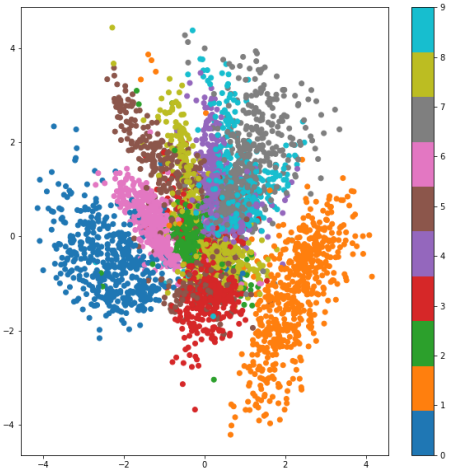
$$z = \mu(z|x) + \epsilon.\sigma(z|x)$$

Where, ϵ is a random sample from Gaussian $N(0, I)$. Putting it all together:

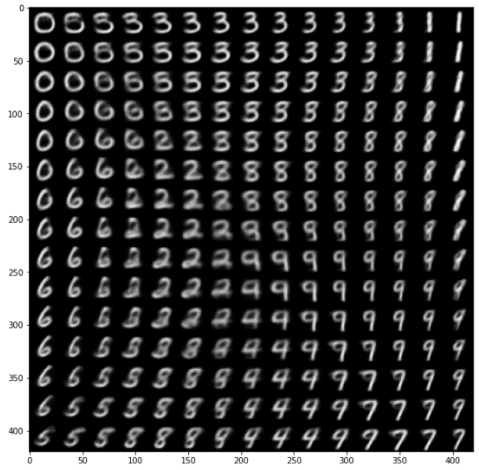


Submission Format:

1. Implement the assignment in Python and submit a Jupyter Notebook file with the following name format `Assignment1_FirstName_LastName.ipynb`
2. Convert your notebook to pdf after saving it with all the outputs and save the file as `Assignment1_FirstName_LastName.pdf`
3. When executed the notebook should output the following:
    a. Training curves (loss vs epochs) (25 pts)
    b. 2D-scatter plot of the z values corresponding to a random subset of 5000 MNIST input images (outputs of the encoder followed by reparameterization) (example below) (25 pts)



    c. A grid plot of 15x15 generated images which are the outputs of the decoder generated for the complete range of z values across the two components of z. (example below) (25 pts)



d. Create a grid of generated images where each row/column corresponds to varying images for one digit. Identify the z values that will generate the digit and vary the z to generate 5 samples for each digit. (Sample image below) (25 pts)

Your submissions will be executed to verify your solutions. The code and report will be evaluated for plagiarism. Ensure the notebook will execute on a generic Google Colab environment without the need for change/debugging. If it is not possible to execute your code, it is not possible to verify your results. The submission will also be evaluated for the quality of images in each of these outputs.