Implementation of Variational Auto-Encoders on MNIST Dataset

Anushka Gupta[1], Aravind Reddy [1], Nathan Starliper[1]
[1]Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC
Email: {agupta35, rkarnam, nstarli}@ncsu.edu,

***Abstract*—Variational auto-encoders us design generative models of data and fit them to large data-sets, and can also be used for image generation and reinforcement learning. In this report, we train VAE models varying the dimensionality of latent code (2D, 10D, 20D) and then randomly generate images using these three trained models.**

***Index Terms*—Variational Auto-encoder, MNIST, latent code.**

I. Introduction

Variational Autoencoder is an encoder-decoder neural network that can encode the image into a vector in the latent space of z real numbers [1]. The vector is a random sample drawn from a z-dimensional normal distribution. The Decoder network decodes the vector and obtains the original image. There are many possible choices of encoders and decoders, depending on the type of data and model. In our example we used relatively simple neural networks, namely multi-layered perceptrons (MLPs).For the encoder and decoder we used MLPs with Gaussian output.

II. Dataset

The dataset used is the MNIST dataset. It is a database of handwritten digits with each image a gray scale image of dimension 28*28. We used 60000 images in the training set and 10000 images in the test set.

III. Architecture

A. Structure of the network

The input to the encoder is a flattened image with 784 input nodes, followed by an intermediate layer of 512 nodes and then the latent dimensions (2D, 10D, 20D) in the three cases. The decoder takes the one of the samples drawn from these latent dimensions as input, which is then fed to a hidden layer with 512 nodes and gives original flattened image as output. The structure of the encoder and the decoder network for latent dimensions 2 is as shown in the Fig. 1 and Fig. 2 respectively.

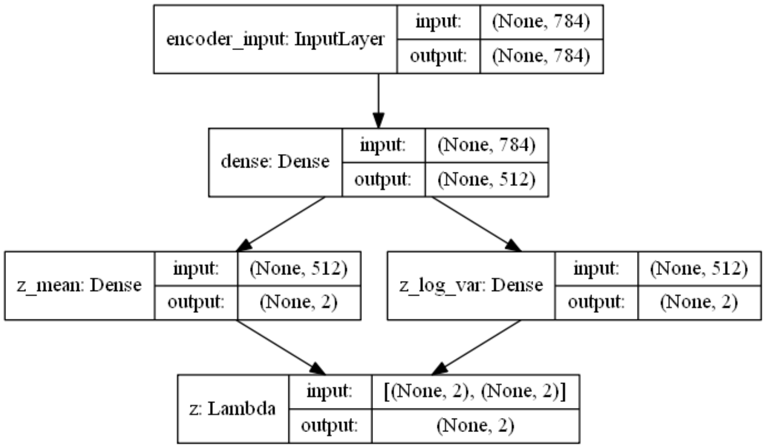

Fig. 1: Encoder Network Architecture

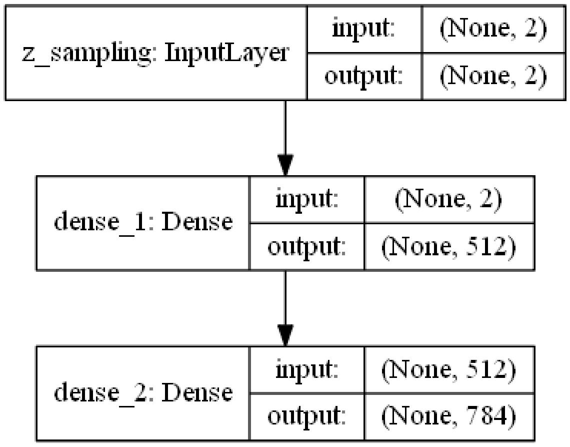

Fig. 2: Decoder Network Architecture

B. Loss Function

The loss for the VAE model consists of reconstruction and KL-divergence terms:

$$\text{VAE Loss} = \text{Reconstruction loss} + \text{K-L Divergence}$$

or, more explicitly,

$$l_i(\theta, \phi) = -E_{z \ q_\theta(z|x_i}[\log p_\phi(x_i|z)] + KL(q_{theta}(z|x_i)||p(z))$$

The first term encourages correct reconstruction of the data (clustering different input classes together in the latent space), while the second term, the KL divergence, functions as a regularizer of sorts, promoting uniform distribution of *all* data over the latent space by enforcing the q distribution to be as similar as possible to the ideal distribution p. Together, these terms teach a latent-space distribution with nearest-neighbor similarity (via clustering) while globally being densely packed around the origin.

IV. Implementation

The code is implemented using python with keras as deep learning framework. The code is executed for latent spaces 2, 10 and 20 and the results are as shown in the next section. The number of epochs are chosen to be 50 and the batch size chosen is 128. The visualizations created a similar to the experiments carried out by Mahkzani et. al. [2].

V. Results

The following are the results obtained for two latent spaces: Visualization of learned data manifold for generative models with two-dimensional latent space is shown in the Fig. 3. Since the prior of the latent space is Gaussian, linearly spaced coordinates on the unit square were transformed through the inverse CDF of the Gaussian to produce values of the latent variables z. For each of these values z, we plotted the corresponding generative p(x—z) with the learned parameters .

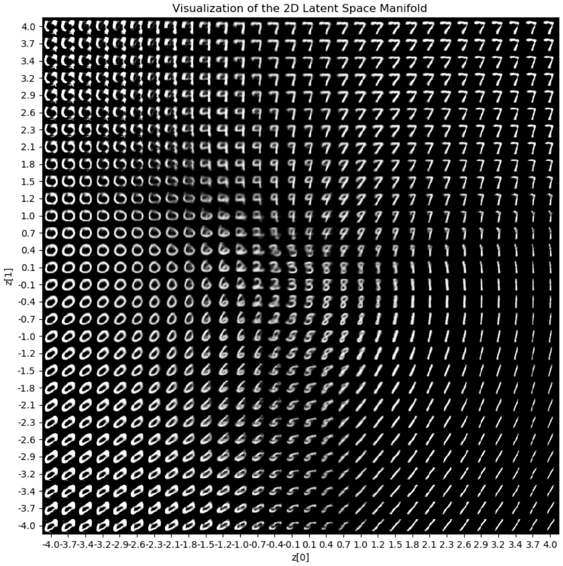

Fig. 3: Digits over latent space for 2 dimensions

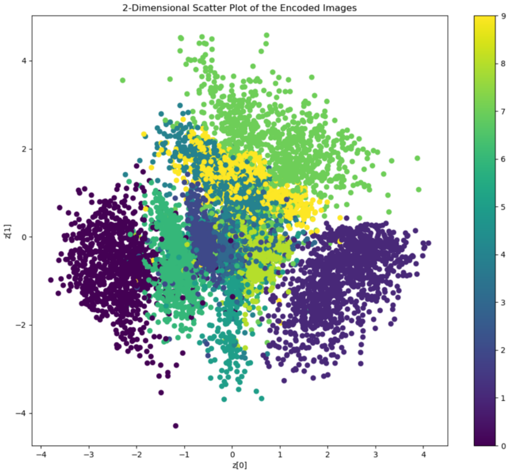

Fig. 4: Visualization of latent space for 2 dimensions

The Fig. 4 shows the hidden code z of the hold-out images for an auto-encoder fit to a 2-D Gaussian. The samples from the learned generative models for 2D, 10D and 20D are as shown in figures 5, 6 and 7.

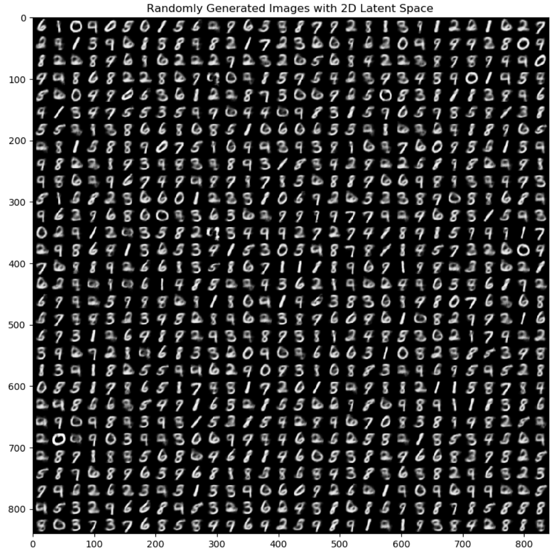

Fig. 5: The random samples from learned generative models of MNIST for 2 latent space dimensions.

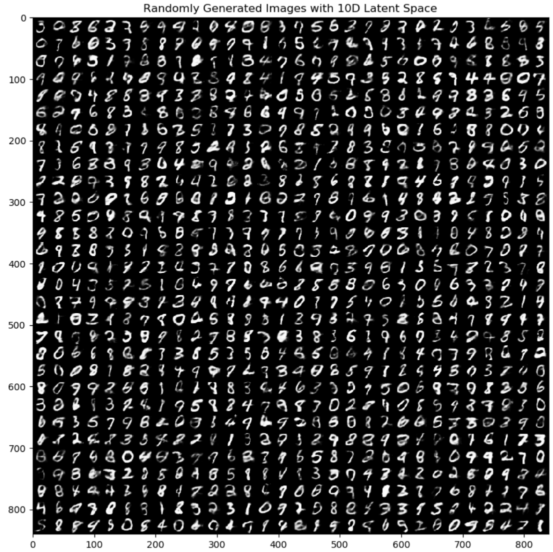

Fig. 6: The random samples of MNIST for 10 latent space dimensions.

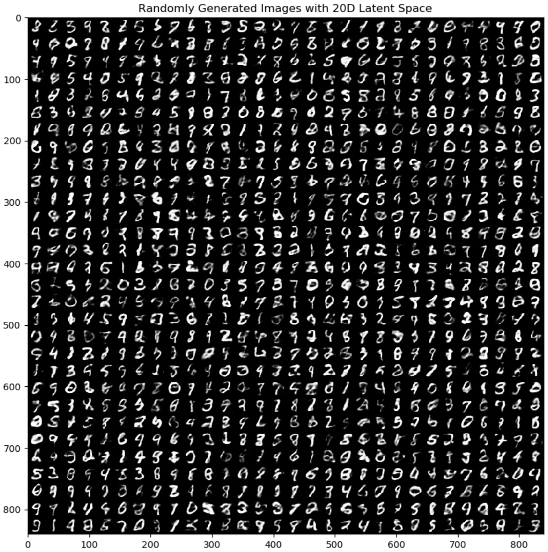

Fig. 7: The random samples of MNIST for 20 latent space dimensions.

REFERENCES

[1] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” *CoRR*, vol. abs/1312.6114, 2013.

[2] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, “Adversarial autoencoders,” *CoRR*, vol. abs/1511.05644, 2015.