Variational Auto-Encoder

Ojas Barve
Electrical and Computer Engineering
North Carolina State University
Raleigh, USA
ovbarve@ncsu.edu

Priya Diwakar
Electrical and Computer Engineering
North Carolina State University
Raleigh, USA
psdiwaka@ncsu.edu

Monil Mehta
Electrical and Computer Engineering
North Carolina State University
Raleigh, USA
mmehta@ncsu.edu

***Abstract*—We implement a Variational Auto-Encoder where the generative model (encoder) and variational approximation (decoder) have Gaussian outputs. We train the model on MNIST dataset using Keras.**

***Index Terms*—variational, auto-encoder, MNIST, gaussian distribution, keras**

I. Introduction

A Variational auto-encoder provides a probabilistic manner for describing an observation in the latent space. The encoder is formulated such that it describes a probability distribution for the latent variable z.

In this project, we let the prior over the latent variables be multivariate Gaussian $p_\theta(z) = \mathcal{N}(0, I)$ .We let $p_\theta(x|z)$ be a multivariate Gaussian whose distribution parameters are computed from z with a neural network. We assume that the true (but intractable) posterior $p_\theta(z|x)$ takes on a approximate multivariate Gaussian distribution and so we let the variational approximate posterior $q_\phi(z|x^{(i)})$ be a multivariate Gaussian $\mathcal{N}(\mu^{(i)}, \sigma^{2(i)}I)$.

The idea behind the adversarial training is to maximize the Lower bound, which is given by

$$L(\theta,\phi;x) = 0.5 * \sum_{j=1}^{J}(1 + log((\sigma_j)^2) - ((\mu_j)^2) - ((\sigma_j)^2)) + \frac{1}{L}\sum_{l=1}^{L}(logp_\theta(x(i)|z(i,l))) \tag{1}$$

II. Network Structure

We implemented two different network structures.

- MultiLayer Perceptron
- Convolutional layers and MultiLayer Perceptron

A. *MultiLayer Perceptron*

1) Encoder:
Fig. 1 shows the encoder structure. The input to the encoder are the image pixels. The hidden layer has 512 hidden units. The encoder has two output layers, one each for the mean and variance.

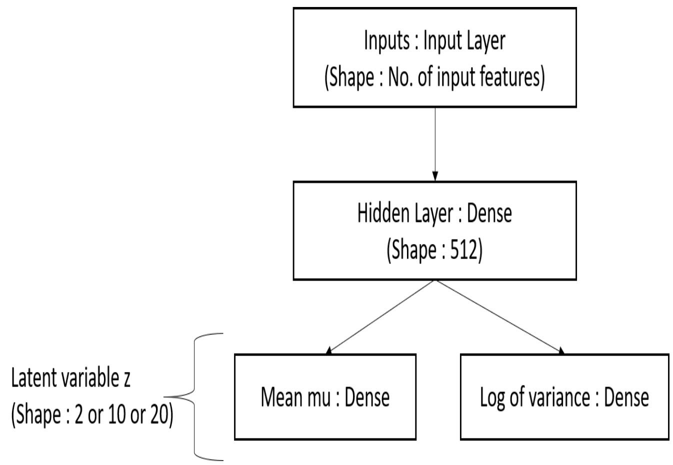

Fig. 1. Encoder

2) Decoder:
Fig. 2 shows the decoder structure. The latent variable z is the input for the decoder. The hidden layer has 512 hidden units. The decoder has one output layer, whose output is the reconstructed image.

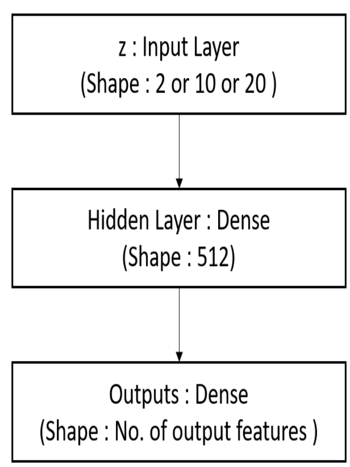

Fig. 2. Decoder

B. *Convolutional layers and MultiLayer Perceptron*

1) Encoder:
Fig. 3 shows the encoder structure. The image is the input

for the encoder. The image is passed through 4 Convolutional 2D layers having 32,64,64,64 feature maps and each with a kernel size of 3 X 3. The output of the convolutional layers is flattened to a vector and passed to a hidden layer with 32 hidden units. The encoder has two output layers, one each for the mean and variance.

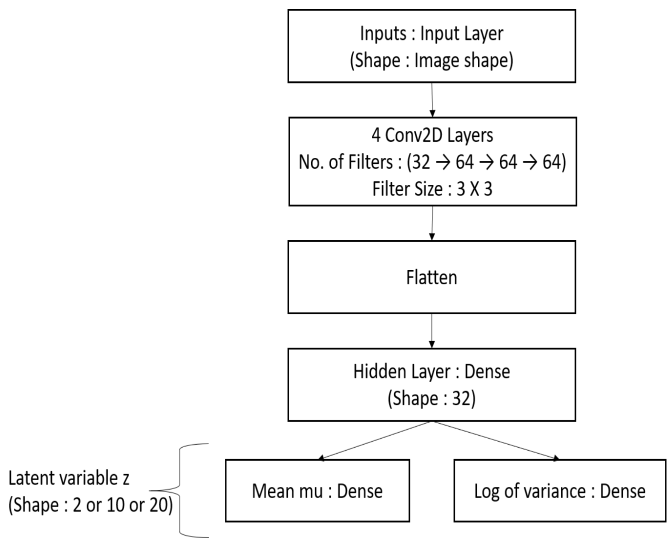

Fig. 3. Encoder

2) Decoder:
Fig. 4 shows the decoder structure. The latent variable z is the input for the decoder. The hidden layer has hidden units equal to the number of image pixels and its output is reshaped to the input image size. This reshaped output is passed to the deconvolutional layer with 32 filters with a kernel size of 3 X 3. The last layer is a convolutional layer with 1 filter with a kernel size of 3 X 3. The output is the reconstructed image.

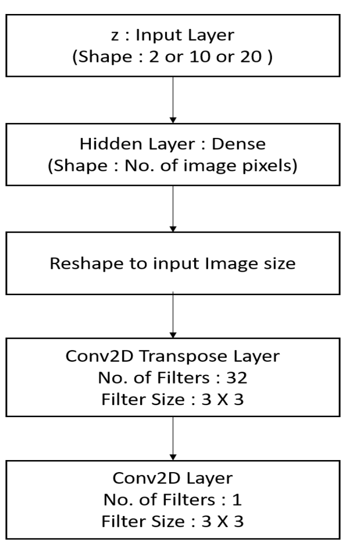

Fig. 4. Decoder

III. Results

A. MultiLayer Perceptron

Results obtained from the MultiLayer Perceptron network are shown in Figures 5, 6, 7, 8 and 9.

1) 2D Latent Variable:
In Fig. 5, we visualize the representation of digits in latent space colored according to their digit labels. In Fig. 6, we observe the Visualization of 2D manifold of MNIST digit. Fig. 7 shows the randomly generated images using the trained VAE model.

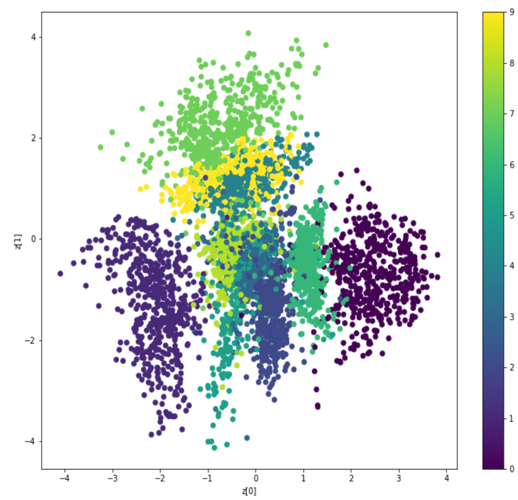

Fig. 5. Latent Variable Representation for 2D space

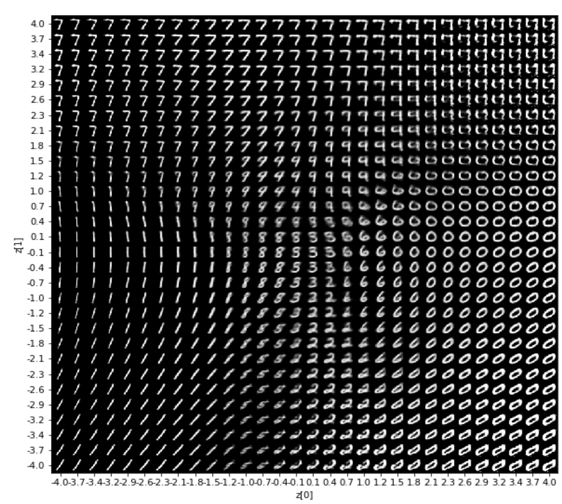

Fig. 6. 2D Latent Space Manifold

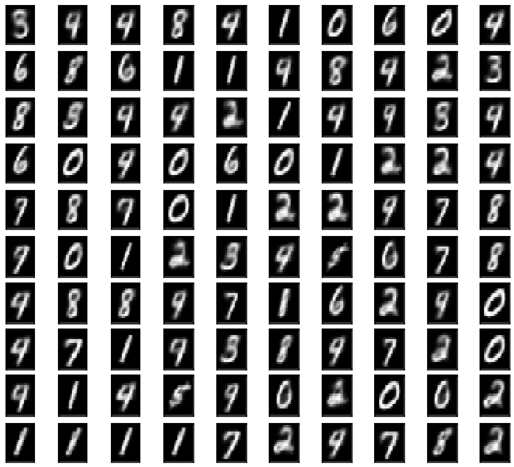

Fig. 7. Reconstructed Images for 2D latent space

2) 10D Latent Variable:
Fig. 8 shows the randomly generated images using the trained VAE model for 10D latent space.

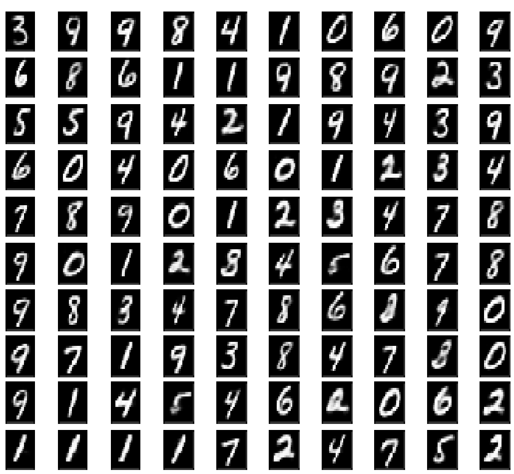

Fig. 8. Reconstructed Images for 10D latent space

3) 20D Latent Variable:
Fig. 9 shows the randomly generated images using the trained VAE model for 20D latent space.

B. Convolutional layers and MultiLayer Perceptron

Results obtained from the Convolutional layers and MultiLayer Perceptron network are shown in Figures 10, 11, 12, 13 and 14.

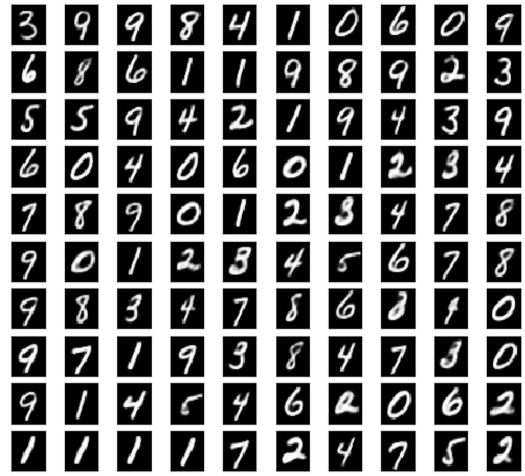

Fig. 9. Reconstructed Images for 20D latent space

1) 2D Latent Variable:
In Fig. 10, we visualize the representation of digits in latent space colored according to their digit labels. Compared to the previous network, we observe a much better visualization of the clusters of the classes. In Fig. 11, we observe the Visualization of 2D manifold of MNIST digit. Fig. 12 shows the randomly generated images using the trained VAE model.

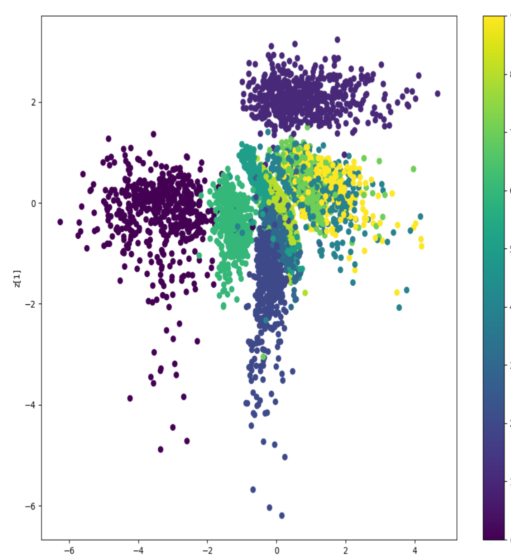

Fig. 10. Latent Variable Representation for 2D space

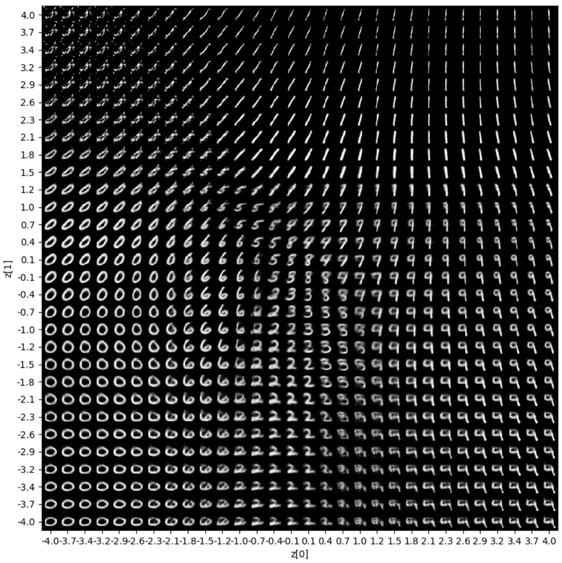

Fig. 11. 2D Latent Space Manifold

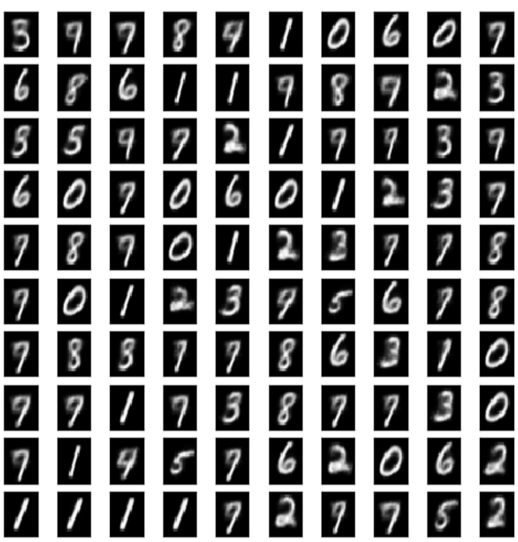

Fig. 12. Reconstructed Images for 2D latent space

2) 10D Latent Variable:

Fig. 13 shows the randomly generated images using the trained VAE model for 10D latent space.

3) 20D Latent Variable:

Fig. 14 shows the randomly generated images using the trained VAE model for 20D latent space.

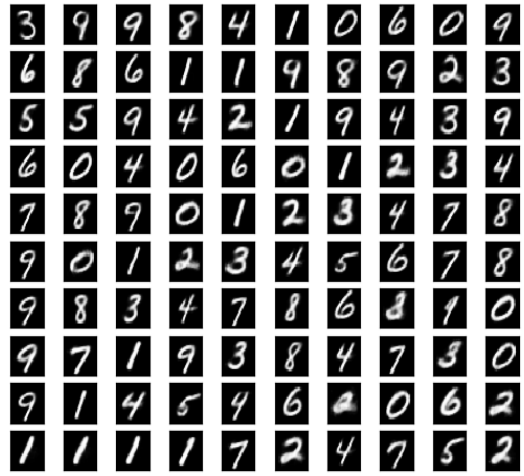

Fig. 13. Reconstructed Images for 10D latent space

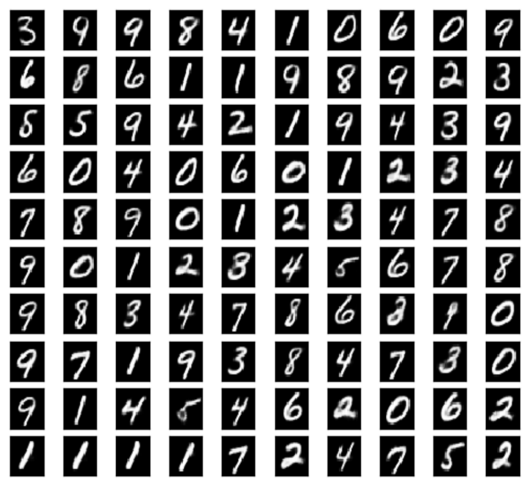

Fig. 14. Reconstructed Images for 20D latent space

REFERENCES

[1] Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." arXiv preprint arXiv:1312.6114 (2013).

[2] Makhzani, Alireza, et al. "Adversarial autoencoders." arXiv preprint arXiv:1511.05644 (2015).