Center for Computational Engineering and Networking

M.Tech Data Science

Academic Year 2020 - 2022

Applications of Variational Autoencoders

H Theivaprakasham CB.EN.P2CEN20026

A report submitted to **Dr. Gopalakrishnan EA** in partial fulfilment of the requirements of the course 20DS602 – Probabilistic Graphical Models and Deep learning.

Amrita Vishwa Vidyapeetham
Center for Computational Engineering and Networking
Coimbatore
Tamil Nadu
641112
India

Abstract

Increasing awareness and availability of data has impacted the field of generative algorithms positively. The concept of auto-encoders have proved to be very successful in many application. But the deterministic auto-encoders aren't still successful to use them as generative models. In this study, we explore the probabilistic graphical model variant of auto-encoders named Variational Auto-encoders and also few of their applications namely image generation, image denoiser and in anomaly detection.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Problem

One major division in machine learning is generative versus discriminative modeling. While in discriminative modeling one aims to learn a predictor given the observations, in generative modeling one aims to solve the more general problem of learning a joint distribution over all the variables. A generative model simulates how the data is generated in the real world.

Generative modeling can be useful more generally. One can think of it as an auxiliary task. For instance, predicting the immediate future may help us build useful abstractions of the world that can be used for multiple prediction tasks downstream. This quest for disentangled,semantically meaningful, statistically independent and causal factors of variation in data is generally known as unsupervised representation learning, and the variational autoencoder (VAE)[4] has been extensively employed for that purpose. Alternatively, one may view this as animplicit form of regularization: by forcing the representations to be meaningful for data generation, we bias the inverse of that process, which maps from input to representation, into a certain mould. The auxiliary task of predicting the world is used to better understand the world a tan abstract level and thus to better make downstream predictions. The VAE can be viewed as two coupled, but independently parameterized models: the encoder or recognition model, and the decoder or generative model. These two models support each other. The recognition model delivers to the generative model an approximation to its posterior over latent random variables, which it needs to update its parameters inside an iteration of "expectation maximization" learning. Reversely, the generative model is a scaffolding of sorts for the recognition model to learn meaningful representations of the data, including possibly class-labels. The recognition model is the approximate inverse of the generative model according to Bayes rule.One advantage of the VAE framework, relative to ordinary Variational Inference (VI), is that the recognition model (also called inference model) is now a (stochastic) function of the input variables. This in contrast to VI where each datacase has a separate variational distribution, which is inefficient for large data-sets. The recognition model uses one set of parameters to model the relation between input and latent variables and as such is called "amortized inference". This recognition model can be arbitrary complex but is still reasonably fast because by construction it can be done using a single feed forward pass from input to latent variables. However the price we pay is that this sampling induces sampling noise in the gradients

required for learning. Perhaps the greatest contribution of the VAE framework is the realization that we can counteract this variance by using what is now known as the "reparameterization trick", a simple procedure to reorganize our gradient computation that reduces variance in the gradients [3].

1.2 Research Objectives

In this study, we
(i) Explore the mathematical development of Variational Auto Encoders
(ii) Implement various applications of Variational Auto Encoders namely Image Generation, Image Denoising and Image anomaly detection.

Chapter 2

Background

2.1 Related works

Burda et al [2] proposed the importance weighted autoencoder (IWAE), a generative model with the same architecture as the VAE, but which uses a strictly tighter log-likelihood lower bound derived from importance weighting. In the IWAE, the recognition network uses multiple samples to approximate the posterior, giving it increased flexibility to model complex posteriors which do not fit the VAE modeling assumptions. [1] introduced the Introspective Adversarial Network, a novel hybridization of the VAE and GAN. which efficiently captures long-range dependencies through use of a computational block based on weight-shared dilated convolutions, and improves generalization performance with Orthogonal Regularization, a novel weight regularization method. They also validated the contributions on CelebA, SVHN, and CIFAR-100, and produced samples and reconstructions with high visual fidelity.

Lin et al [5] proposed DR-A (Dimensionality Reduction with Adversarial variational autoencoder), a data-driven approach to fulfill the task of dimensionality reduction. DR-A leverages a novel adversarial variational autoencoder-based framework, a variant of generative adversarial networks. DR-A is well-suited for unsupervised learning tasks for the scRNA-seq data, where labels for cell types are costly and often impossible to acquire. Compared with existing methods, DR-A is able to provide a more accurate low dimensional representation of the scRNA-seq data. Shao et al [6] proposed a modification to VAE named ControlVAE which was used to evaluate tasks on image and language. The results show that ControlVAE can achieve much better reconstruction quality than the other methods for comparable disentanglement. On the language modeling task, ControlVAE can avoid posterior collapse (KL vanishing) and improve the diversity of generated text. Wan et al [7] proposed a novel triplet domain translation network by leveraging real photos along with massive synthetic image pairs by training two variational autoencoders (VAEs) to transform old photos and clean photos into two latent spaces. And the translation between these two latent spaces is learned with synthetic paired data. The proposed pipeline demonstrated superior performance over state-of-the-art methods as well as existing commercial tools in terms of visual quality for old photos restoration.

Chapter 3

Methodology

The workflow of the experiments consisted of five steps. (1) Dataset and preparation (2) Variational Auto-encoder (3) VAE training and (4) VAE performance assessment.

3.1 Dataset

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. The black and white images from NIST were normalized to fit into a 28x28 pixel bounding box and anti-aliased, which introduced grayscale levels. The MNIST database contains 60,000 training images and 10,000 testing images. Half of the training set and half of the test set were taken from NIST's training dataset, while the other half of the training set and the other half of the test set were taken from NIST's testing dataset 3.1.

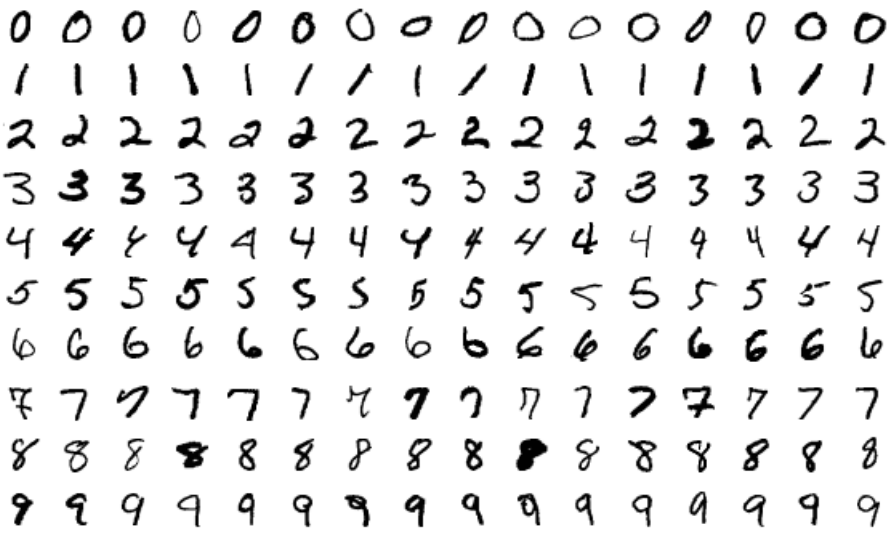

Figure 3.1: Few random sample images from MNIST dataset

3.2 Variational Auto-encoder

A variational auto-encoder (VAE) provides a probabilistic manner for describing an observation in latent space. Thus, rather than building an encoder which outputs a single value to describe each latent state attribute, we'll formulate our encoder to describe a probability distribution for each latent attribute. With this approach, we'll now represent each latent attribute for a given input as a probability distribution. When decoding from the latent state, we'll randomly sample from each latent state distribution to generate a vector as input for our decoder model. By constructing our encoder model to output a range of possible values (a statistical distribution) from which we'll randomly sample to feed into our decoder model, we're essentially enforcing a continuous, smooth latent space representation. For any sampling of the latent distributions, we're expecting our decoder model to be able to accurately reconstruct the input. Thus, values which are nearby to one another in latent space should correspond with very similar reconstructions.

Suppose that there exists some hidden variable z which generates an observation x.

We can only see x, but we would like to infer the characteristics of z. In other words, we'd like to compute $p\left(z|x\right)$.

$$p\left(z|x\right) = \frac{p\left(x|z\right)p\left(z\right)}{p\left(x\right)} \tag{3.1}$$

Unfortunately, computing $p\left(x\right)$ is quite difficult.

$$p\left(x\right) = \int p\left(x|z\right)p\left(z\right)dz \tag{3.2}$$

This usually turns out to be an intractable distribution. However, we can apply varitational inference to estimate this value.

Let's approximate $p\left(z|x\right)$ by another distribution $q\left(z|x\right)$ which we'll define such that it has a tractable distribution. If we can define the parameters of $q\left(z|x\right)$ such that it is very similar to $p\left(z|x\right)$, we can use it to perform approximate inference of the intractable distribution.

Recall that the KL divergence is a measure of difference between two probability distributions. Thus, if we wanted to ensure that $q\left(z|x\right)$ was similar to $p\left(z|x\right)$, we could minimize the KL divergence between the two distributions.

$$\min KL\left(q\left(z|x\right)||p\left(z|x\right)\right) \tag{3.3}$$

When we minimize the KL divergence, our q will be an estimator of p.

This is the definition for the KL divergence between two distributions P, Q:

$$D_{KL}(Q||P) = \sum Q\log\frac{Q}{P} \tag{3.4}$$

We'll replace Q, P with our distributions

$$P = p(z|x) and Q = q(z|x) \tag{3.5}$$

$$D_{KL}(q(z|x)||p(z|x)) = \sum q(z|x)\log\frac{q(z|x)}{p(z|x)} \tag{3.6}$$

$\mathrm{D}_{KL}(q(z|x)||p(z|x)) = \sum q(z|x) \log \frac{q(z|x)}{p(z|x)}$

$= \text{-}\sum q(z|x) \log \frac{p(z|x)}{q(z|x)}$

$= \text{-}\sum q(z|x) \left[\log p(z|x) - \log q(z|x)\right]$

$= \text{-}\sum q(z|x) \left[\log \frac{p(x|z)p(z)}{p(x)} - \log q(z|x)\right]$

$= \text{-}\sum q(z|x) \left[\log p(x|z) + \log p(z) - \log p(x) - \log q(z|x)\right]$

$= \sum q(z|x) \left[-\log p(x|z) - \log p(z) + \log p(x) + \log q(z|x)\right]$

$\mathrm{D}_{KL}(q(z|x)||p(z|x)) - \sum q(z|x) \left[-\log p(x|z) - \log p(z) + \log q(z|x)\right] = \log p(x)$

$\text{-}\sum q(z|x) \left[-\log p(x|z) + \log \frac{q(z|x)}{p(z)}\right] =$

$\text{-}\mathrm{E}_{z\sim q(z|x)} \left[-\log p(x|z) + \log \frac{q(z|x)}{p(z)}\right] =$

$\text{-}\mathrm{E}_q \left[-\log p(x|z) + \log \frac{q(z|x)}{p(z)}\right] =$

$\mathrm{E}_q \left[\log p(x|z) - \log \frac{q(z|x)}{p(z)}\right] =$

$\mathrm{D}_{KL}(q(z|x)||p(z|x)) + \left[E_q \log p(x|z) - E_q \log \frac{q(z|x)}{p(z)}\right] = \log p(x)$

To maximize the ELBO, we can minimize the negative of the ELBO.

$$\max E_q \log p(x|z) - E_q \log \frac{q(z|x)}{p(z)} = \min -[E_q \log p(x|z) - E_q \log \frac{q(z|x)}{p(z)}]$$

$\text{-}\mathrm{E}_q \log p(x|z) + E_q[\log q(z|x) - \log p(z)] = E_q[\log q(z|x) - \log p(z)] - E_q \log p(x|z) =$

The end result is:

$$D_{KL}(q(z|x)||p(z|x)) + \left[E_q \log p(x|z) - E_q \log \frac{q(z|x)}{p(z)}\right] = \log p(x) \tag{3.7}$$

The first term is the original D_{KL} we want to minimize, $(D_{KL}(q(z|x)||p(z|x)))$. The second term is called the variational lower bound (ELBO).

$$\max \mathcal{L} = E_q \log p(x|z) - E_q \log \frac{q(z|x)}{p(z)} \tag{3.8}$$

Since $\log p(x)$ is a constant, and we want to minimize the D_{KL} term, we can achieve the same by maximizing the ELBO.

$$\min KL\left(q\left(z|x\right)||p\left(z|x\right)\right) \tag{3.9}$$

So, the end goal is to minimize this function:

$$\min E_q[\log q(z|x) - \log p(z)] - E_q \log p(x|z) \tag{3.10}$$

3.3 VAE Training

3.3.1 Application 1: Image Generator

In the first application of image generator, we use the MNIST dataset. The input and output were the MNIST images. We made use of the Adaptive Moment Estimation (Adam) Optimizer function as they use the techniques of Momentum and Adaptive Learning Rates to converge the model faster. Adam is been widely used due to their success in various domains of deep learning namely image recognition, Natural language Processing and speech processing. We minimize KL Divergence Loss, Reconstruction loss as we derived from equation 3.9. We trained the model for 100 epoch.

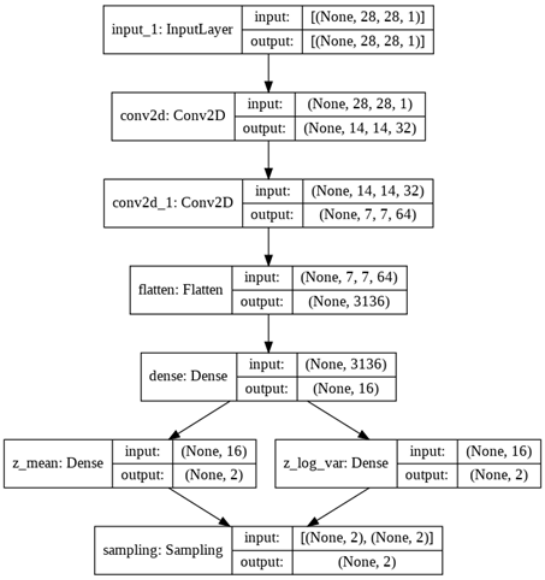

Figure 3.2: VAE Encoder model

The encoder and decoder model are as shown in figure 3.2 and figure 3.3.

3.3.2 Application 2: Image Denoising

In the second application of image denoising, we use the MNIST dataset. The input and output were the MNIST images. We made use of the Adaptive Moment Estimation (Adam) Optimizer function as they use the techniques of Momentum and Adaptive Learning Rates to converge the model faster. Adam is been widely used due to their success in various domains of deep learning namely image recognition, Natural language Processing and speech processing. We minimize KL Divergence Loss, Reconstruction loss as we derived from equation 3.9. We trained the model for 100 epoch. We tested the reconstruction image by additive random noise to the input images. We use the same VAE model as use in figure 3.2 and figure 3.3.

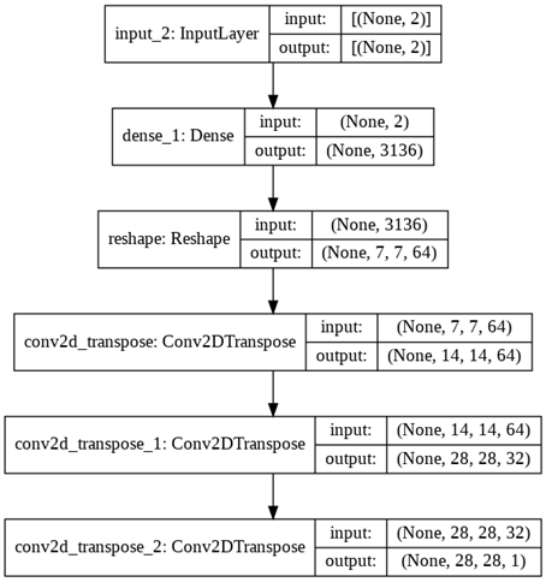

Figure 3.3: VAE Decoder model

3.3.3 Application 3: Image Anomaly detection

In the third application of Image Anomaly detection, we use the MNIST dataset digits 0 to 8 as training set while use only digit 9 as th testing set to detect the anomaly. The input and output were the MNIST images. We made use of the Adaptive Moment Estimation (Adam) Optimizer function as they use the techniques of Momentum and Adaptive Learning Rates to converge the model faster. Adam is been widely used due to their success in various domains of deep learning namely image recognition, Natural language Processing and speech processing. We minimize KL Divergence Loss, Reconstruction loss as we derived from equation 3.9. We trained the model for 100 epoch. We use the same VAE model as use in figure 3.2 and figure 3.3.

All the experiments were conducted using Python as Programming Language and Pytorch v1.7 as Deep learning Libraries and Google Colaboratory as the training environment with a single Nvidia Tesla P100 GPU (16GB RAM), Intel(R) Xeon(R) CPU @ 2.30GHz with 2 cores and 25 GB of DDR4 RAM.

Chapter 4

Results

4.1 Application 1: Image Generator

The output results are obtained by varying the Varying values (Mean and Variance) in Latent Vector between -1 to 1. Manifold Plot depicted in Figure 4.1. The custom digits were also obtained by Varying values (Mean and Variance) in Latent Vector. The custom generated digits are illustrated in Figure 4.2.

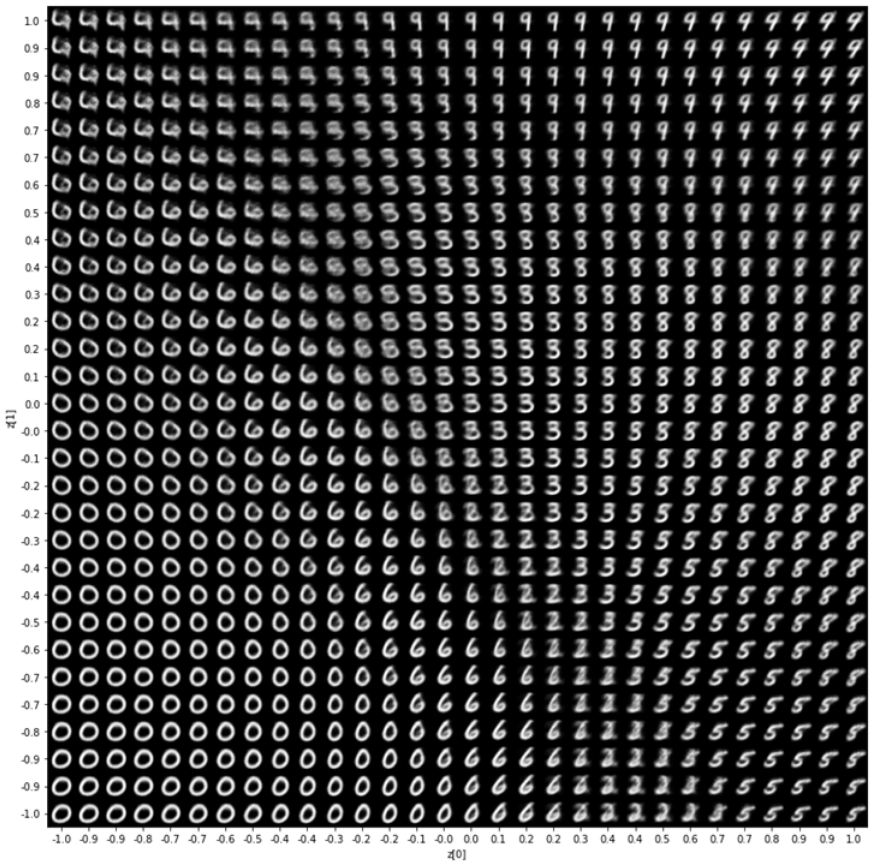

Figure 4.1: Manifold Plot obtained by varying mean and variance

Figure 4.2: Digits generated from VAE by varying the mean and variance of the latent vector

4.1.1 Limitations

We noted that VAE works well at lower resolution and lower dimensional images but produces blurry Images. VAE also needs more training time and dataset for high dimensional images.

4.2 Application 2: Image Denoising

The results on inputting the the noisy image is depicted in the figure 4.3. We observe that the denoised image is very similar to the input image.

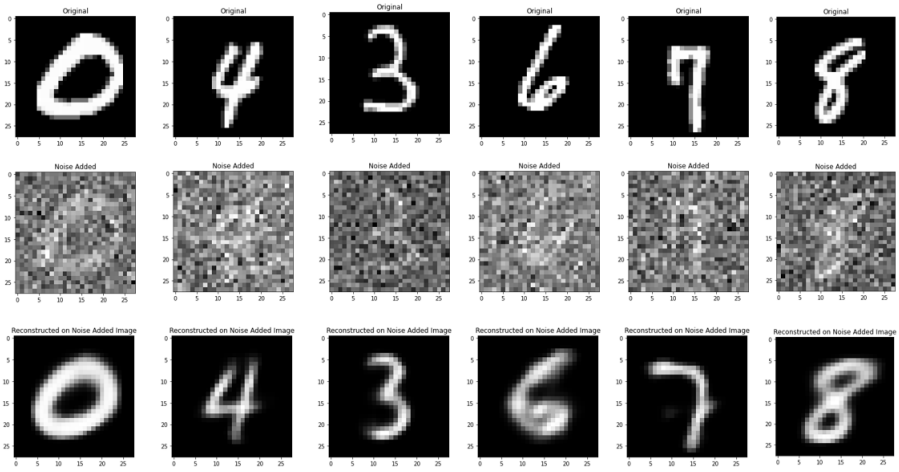

Figure 4.3: Digits denoised by VAE when input with noisy image

4.2.1 Limitations

We observed that the reconstruction was perfect only for digits that we have trained. Though reconstruction is possible for the even with moderate-high noise, our VAE works well at lower resolution and lower dimensional images. VAE is gain observed to produces blurry Images. We also noted that the reconstruction capabilities becomes low when the noise was extreme.

4.3 Application 3: Image Anomaly detection

We tested our VAE model with both images of trained data and anomaly data. We observed that the VAE was able to reconstruct back the fed in input image from training set but failed to reconstruct image when inputted anomaly digit. This is illustrated in figure 4.4.

The figure 4.5 depicts the difference between the original digit and the anomaly reconstructed digit.

The figure 4.6 shows the T-SNE representation of both original digit and the anomaly reconstructed digit. The figure 4.7 shows the reconstruction loss of both original digit and the anomaly reconstructed digit which helps to visually understand the anomaly of the input data. The figure 4.8 shows anomaly prediction performance by calculating the Area Under Convergence value in Region of Convergence plot.

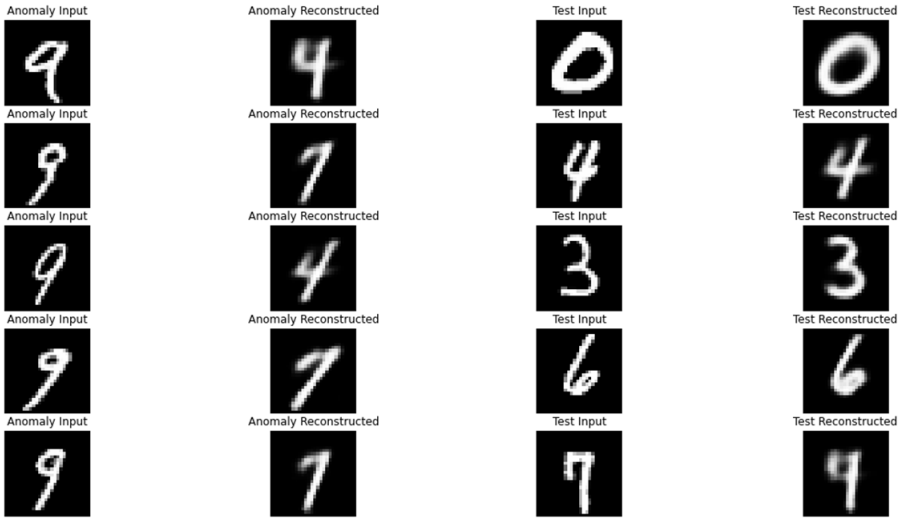

Figure 4.4: Reconstructed image of anomaly and trained digits

4.3.1 Limitations

Though the performance of the anomaly detection is satisfactory, the VAE fails when the input training data has very resembling anomaly data. This is seen from the figure 4.8 where AUC value is just 63%.

Figure 4.5: Anomaly difference plot of the the reconstructed image

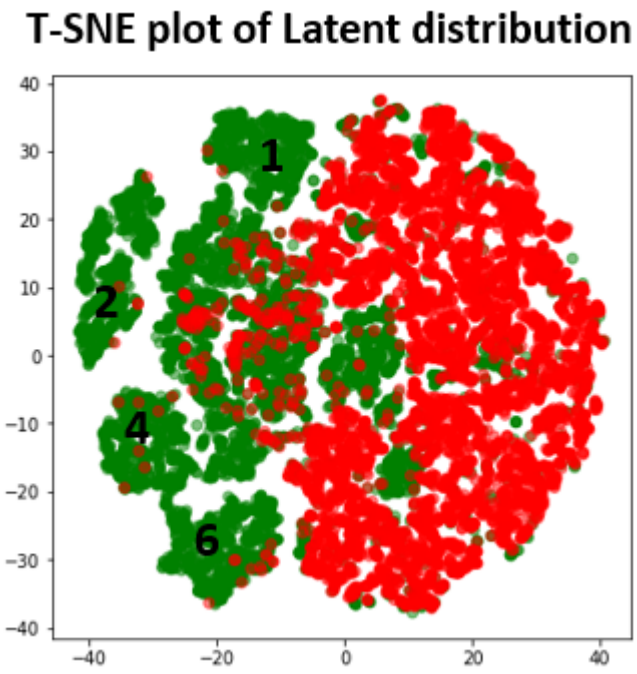

Figure 4.6: T-SNE plot of both trained and anomaly digits

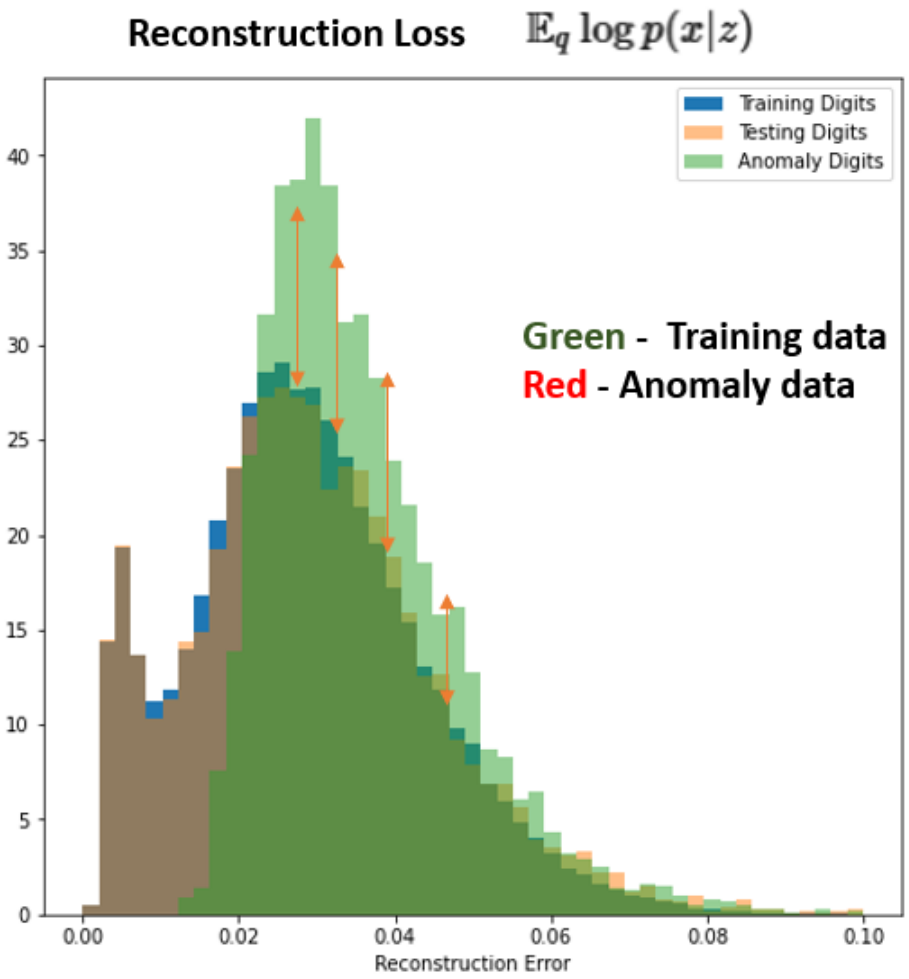

Figure 4.7: Reconstruction loss of trained digits and anomaly digits

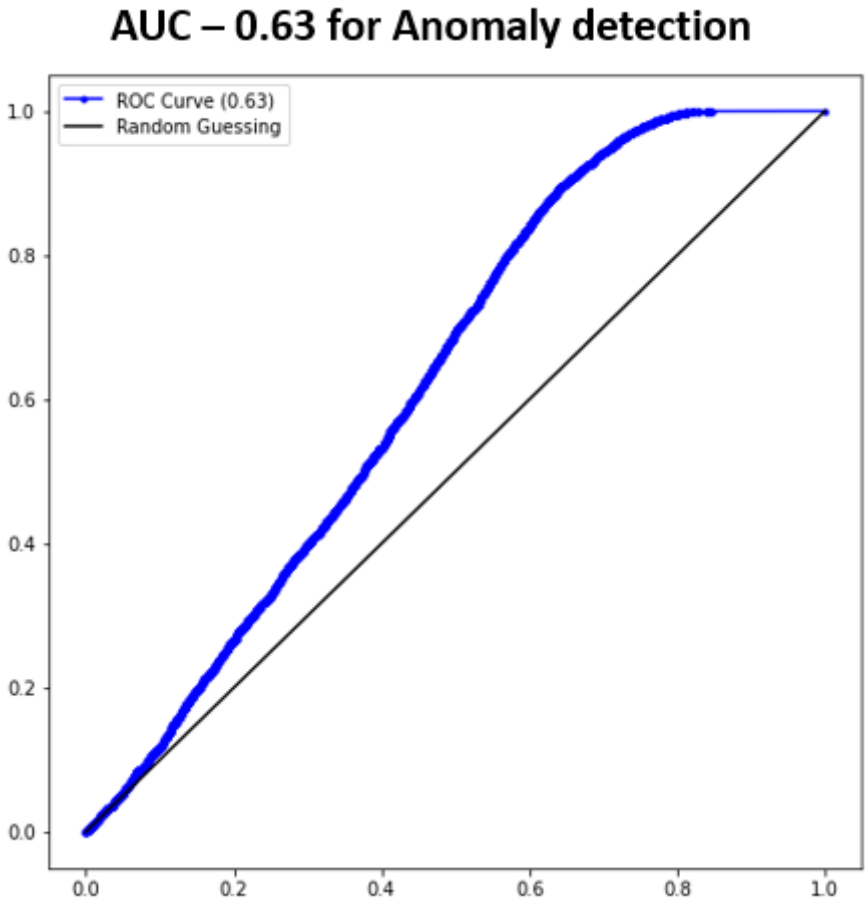

Figure 4.8: Region of convergence graph showing the performance of anomaly detection

Chapter 5

Discussion

We observe that in all of the three applications VAE performed better and indicated that it can be used in image generation, image denoising and image anomaly detection process. One common observation is that the output reconstructed image is slightly blurry in all of the the above applications which can be mainly due to the assumed Gaussian distribution which brings out Gaussian blur. Though the blur is helpful in many of the application, they are not very suitable for applications where we need high resolution features in the output reconstructed images.

Chapter 6

Conclusion

Increasing awareness and availability of data has impacted the field of generative algorithms positively. The concept of auto-encoders have proved to be very successful in many application. But the deterministic auto-encoders aren't still successful to use them as generative models. In this study, we explore the probabilistic graphical model variant of auto-encoders named Variational Auto-encoders and also few of their applications namely image generation, image denoiser and in anomaly detection. The results indicate that the VAE technique can be effectively used for image generation, denoising and anomaly detection applications in future.

Bibliography

[1] Andrew Brock et al. *Neural Photo Editing with Introspective Adversarial Networks*. 2017. arXiv: 1609.07093 [cs.LG].

[2] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. *Importance Weighted Autoencoders*. 2016. arXiv: 1509.00519 [cs.LG].

[3] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2014. arXiv: 1312.6114 [stat.ML].

[4] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. ISSN: 1935-8245. DOI: 10.1561/2200000056. URL: http://dx.doi.org/10.1561/2200000056.

[5] Eugene Lin, Sudipto Mukherjee, and Sreeram Kannan. "A deep adversarial variational autoencoder model for dimensionality reduction in single-cell RNA sequencing analysis". In: *BMC Bioinformatics* 21.1 (2020), p. 64. ISSN: 1471-2105. DOI: 10.1186/s12859-020-3401-5. URL: https://doi.org/10.1186/s12859-020-3401-5.

[6] Huajie Shao et al. *ControlVAE: Tuning, Analytical Properties, and Performance Analysis*. 2020. arXiv: 2011.01754 [cs.LG].

[7] Ziyu Wan et al. *Old Photo Restoration via Deep Latent Space Translation*. 2020. arXiv: 2009.07047 [cs.CV].