Variational Autoencoders

Introduction

In this laboratory we have experimented on how variational autoencoders work using MNIST dataset. The general pipeline we followed starts by implementing individual components(methods) given in the reference script files. Afterwards we trained the network with different epochs to see how the results respond. In this report we have presented a general method we have used in the methodology section and we also presented results of our experiment in the results and analysis section. Finally, we conclude by giving general emphasis on our understanding on how Variational inference works.

Methodology

For the implementation of the specified task we started by understanding the general problem and we find out that the general idea of autoencoders consists in setting an encoder and a decoder as neural networks and to learn the best encoding-decoding scheme using an iterative optimization process. the general generative model we used in this practice is describe in equation 1. The goal is to learn parameters of the model that describes the observed data well. This could be done using by assuming the problem as an optimization problem given the data. But due to the strong nonlinearity of the conditional probability *p(x|z)* the probability of *p(x)* is intractable. We used varational inferenece to better approximate this problem in more simple distribution. An optimization problem is formed interms of the new distributions and we impelemented ADAM as a stocahstic optimization algorithm.

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}\,, \tag{1}$$

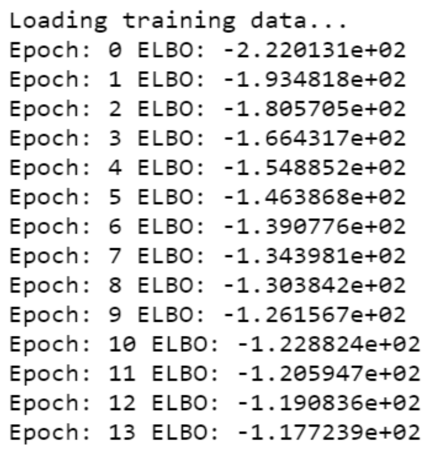
After set of derivations the objective function will have the form described in equation 11.

$$O(\phi,\theta) = \sum_{i=1}^{N} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)}\left[\log p_\theta(\mathbf{x}_i|\mathbf{z})\right] - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x}_i)|p(\mathbf{z}))\,, \tag{11}$$

For the whole process to work we implemented the micros code fragments given in the script and our results are described below.

Results and analysis

Before testing the model, obviously we did the training of the whole network based on the stochastic optimization and we trained the model for different epochs to see the effect of learning more enhances the model. Some results of our training for the evidence of lower bound are shown in figure



```
Loading training data...
Epoch: 0 ELBO: -2.220131e+02
Epoch: 1 ELBO: -1.934818e+02
Epoch: 2 ELBO: -1.805705e+02
Epoch: 3 ELBO: -1.664317e+02
Epoch: 4 ELBO: -1.548852e+02
Epoch: 5 ELBO: -1.463868e+02
Epoch: 6 ELBO: -1.390776e+02
Epoch: 7 ELBO: -1.343981e+02
Epoch: 8 ELBO: -1.303842e+02
Epoch: 9 ELBO: -1.261567e+02
Epoch: 10 ELBO: -1.228824e+02
Epoch: 11 ELBO: -1.205947e+02
Epoch: 12 ELBO: -1.190836e+02
Epoch: 13 ELBO: -1.177239e+02
```

Figure 1 : ELOB of the training

In the first task we generated 25 images using the generative model by first sampling the latent variable from a normal distribution prior and then feeding this sample to the generative model. One can observe that the numbers are ambiguous in some cases (second row first column). This is because we sampled our latent variable from a normal distribution with out any knowledge of the actual data.

For different optimizations we find out slightly different results. From this we understand that the more our model learns from the data, the generative model performs good as illustrated in figure 2.

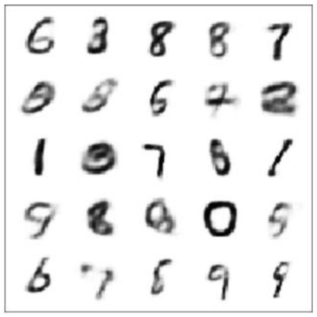


Figure 2 : #epochs = 30, z from uniform distribution

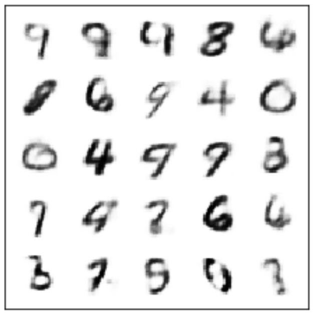


Figure 3 : #epochs = 20, z from uniform distribution

If we look closely on the reconstructed and generated data, we would notice that some of the data are ambiguous That's because our latent variable space is a continuous distribution (i.e. N(0,1)), hence there bound to be some smooth transition on the edge of the clusters. And, the cluster of digits are close to each other if they are somewhat similar. That is why in the latent space, 5 is close to 3

In the second experiment we visualized the effect of sampling the latent space(z) from the actual reconstructed image space and we observed that , the generative model was be able to generate very good approximations of the actual data , given the latent variable as and input. This tells us that variational inference can enable us to learn significant amount of information about a underling complex data using simple latent variables which acts as a high level representation of the actual underlying data.

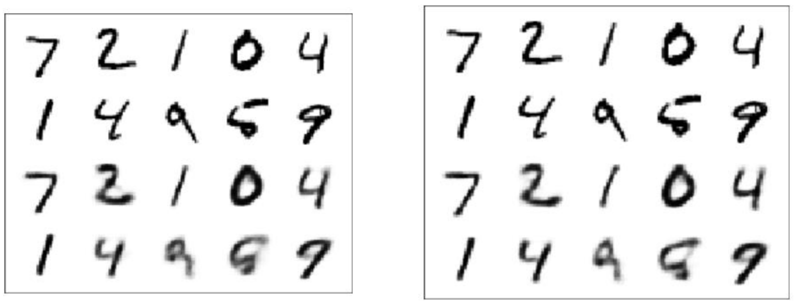


Figure 4 : In this case our generative model gives as similar results for different epochs

In this case our generative model gives us good reconstruction of the images as the latent space is sampled from the actual data. As we can observe from figure 4 as the latent variable is sampled from the actual data, the different between the two parameters

The third task involves interpolation of pair of images in the latent space from one image to another image. In this task we first represent the both pair of images into their latent space and afterwards we generated weighted linear combination of their latent vectors to reconstruct the interpolation of the two images by feeding the network.

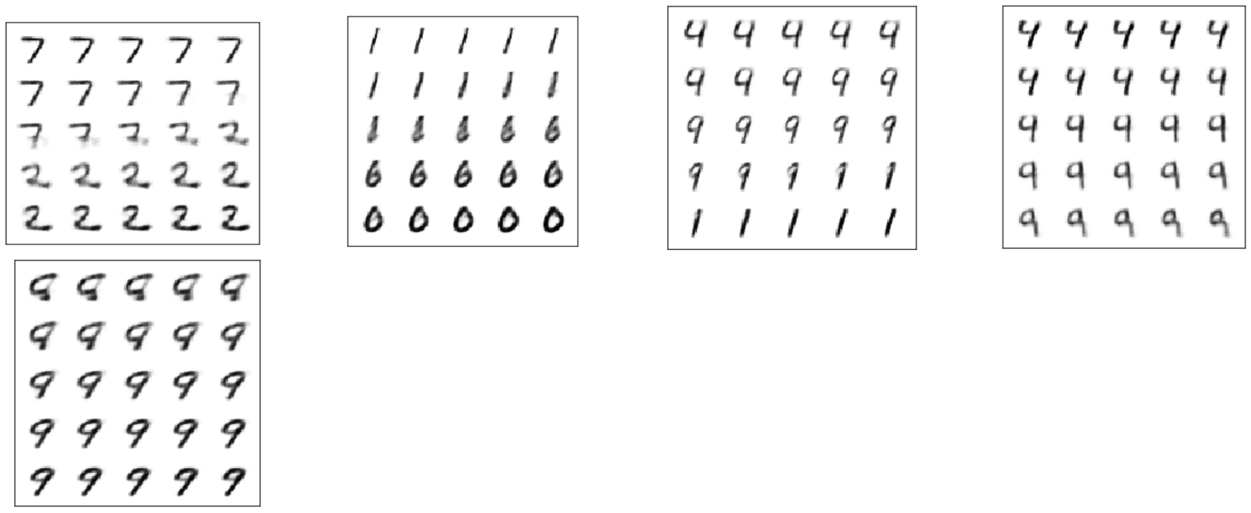


Figure 5 : Interpolation of pair of images

As one can clearly observe from the above images the interpolation of images seems smooth enough although we can observe that there are some ambiguous number in the middle of the translation in between the images.