Benchmarking Variational Autoencoder with MNIST Dataset

Professor Bryon Aragam

March 2024

Contents

1 Theory **2**
1.1 About Variational Autoencoders (VAE) 2
1.1.1 Encoders and Dimensionality Reduction 2
1.1.2 Autoencoders 2
1.1.3 Variational Autoencoders 2
1.2 Underlying Model 2
1.2.1 Autoencoder 2
1.2.2 Variational Autoencoder 3
1.3 Evaluation 4
1.3.1 Loss Function 4
1.4 Inception Score 5

2 Implementation **6**
2.1 First Attempt 6
2.2 Increasing Latent Space Dimension 9
2.2.1 Qualitative Assessment of KL Divergence 14
2.2.2 Qualitative Assessment of Reconstruction Loss 21

3 Potential for Further Exploration **22**

1 Theory

1.1 About Variational Autoencoders (VAE)

1.1.1 Encoders and Dimensionality Reduction

Encoders are a specialized class of algorithms that reduce the number of features describing an input. The encoder transforms input data into a reduced dimensional representation called the latent space. The decoder then attempts to reconstruct the original input faithfully.

More generally, **dimensionality reduction** is a statistical problem that aims to find the best encoder/decoder pair that maintains minimum information loss during the encoding/decoding process. In other words, let E, and D denote the set of all encoders and decoders, respectively, we aim to find $e \in E$ and $d \in D$ that satisfies the following

$$arg\,min\,\epsilon(x, d(e(x)))$$

1.1.2 Autoencoders

Autoencoders are encoders that use neural networks to compress the input data into the latent space. Autoencoders are comprised of two parts:

1. **Encoder**: A process that extracts features from the sample. In this example, we will use a convoluted neural network (CNN). The encoder collapses an input into a point in the latent space.
2. **Decoder**: The decoder takes the point in the latent space and tries to reconstruct the original image.

1.1.3 Variational Autoencoders

In addition to requiring minimal reconstruction loss, variational autoencoder also requires the latent space to follow a specified distribution.

More than dimensionality reduction, VAEs can also be used for generative modeling. By learning the underlying probability distribution of the input data, VAEs can generate new data samples that are similar to the original dataset.

1.2 Underlying Model

1.2.1 Autoencoder

Autoencoders leverage neural networks for dimensionality reduction. Both the encoder and the decoder are neural networks trained to learn the best encoding-decoding scheme. For each training iteration, we feed the autoencoder architecture with a sample and compare the encoded-decoded output with the original.

Using backpropagation, the weight parameters of the neural network are adjusted until the loss is minimized. In this implementation, the optimizer is the stochastic gradient descent with a learning rate of 0.0005.

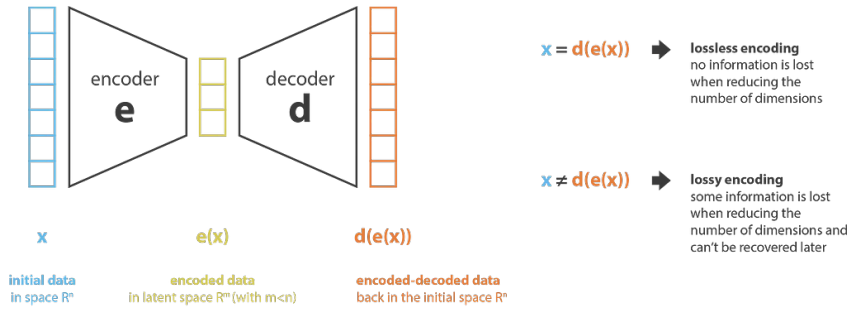


Figure 1: The model architecture of Variational Autoencoder

Encoders can be susceptible to overfitting. Given a sufficiently complex encoder and decoder, we can map all the training samples onto the real axis and decode them without reconstruction loss.

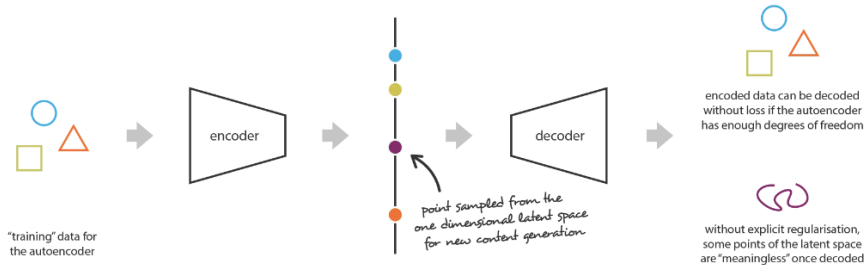


Figure 2: Potential for overfitting with traditional autoencoder

1.2.2 Variational Autoencoder

Variational autoencoder regularizes for overfitting by requiring a regular latent space. The requirement of a well-formed latent space also allows for the generation of novel images as the latent space is dense. The training consists of the following steps.

1. First, the input is encoded as a distribution over the latent space.
2. Second, a point from the latent space is sampled from the distribution.

3. Third, the sample point is decoded and the reconstruction error can be computed
4. Finally, the reconstruction error is calculated and backpropagation is performed.

A simple autoencoder and a variational encoder is contrasted in the diagram below.

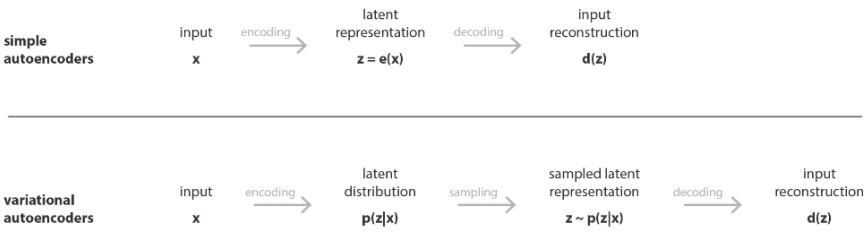


Figure 3: Simple autoencoder versus variational autoencoder

1.3 Evaluation

1.3.1 Loss Function

The loss function should thus capture the two objectives of variational autoencoders. First, we want the reconstruction loss to be minimized. In other words, the encoded-decoded reconstruction needs to be as similar to the original as possible. Furthermore, we require the distribution in the latent space to be as close to the chosen distribution as possible. In practice, the latent space is often assumed to be the multivariate Gaussian distribution.

In my implementation, the reconstruction loss is captured by the MSE loss, given by the average squared difference between the original pixel and the reconstructed pixel.

$$MSE = \frac{1}{2m}\sum_{i=1}^{m}(Y_i - \hat{Y}_i)^2$$

The Kullback-Leibler Divergence is used for the conform the latent space to the multivariate Gaussian. KL divergence measures how one probability distribution P is different from a second, reference probability distribution Q. The KL divergence is represented by $D_{KL}(P||Q)$.

For continuous probability distributions P and Q defined on the same sample space X, the relative entropy is defined to be the integral

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x)log(\frac{p(x)}{q(x)})dx$$

where p and q are probability densities functions of P and Q. In other words, it is the expectation of the logarithmic difference between the probabilities P and Q. The expectation is taken using the probabilities of P.

Plugging in multivariate Gaussian distribution for p(x) and q(x), we can arrive at the following closed form solution.[1]

$$KL = (N(x|\mu_1, \sigma_1)||N(x|\mu_2, \sigma_2)) = log\frac{\sigma_2}{\sigma_1} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2)}{2\sigma_2^2} - \frac{1}{2}$$

1.4 Inception Score

The **inception score** can also be calculated to quantitatively measure the quality of the VAE performance. The inception score uses a pre-trained classifier and seeks to capture two properties:

- Quality: Does the image look like a specific object?
- Diversity: Is a wide range of objects generated?

After sampling the latent space, the decoder generates a novel image. The classifier calculates the probability that the novel image belongs to each output class (label distribution). If the classifier is confident, the distribution is narrow (i.e. one peak)

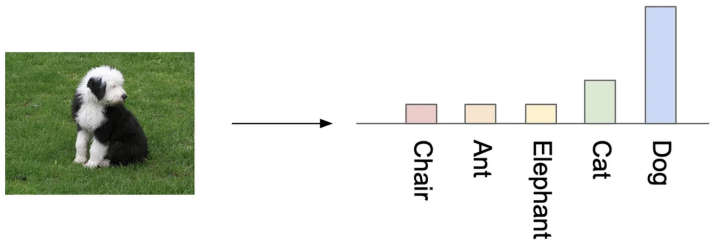


Figure 4: If the produced image is of high quality, the classifier will be confident in its result. Therefore the label distribution will be narrow.

By sampling from the latent space, we can generate many images and sum up its marginal distribution. The marginal distribution is indicative of the variety of our generator's output. If the diversity is high, the marginal distribution will be uniform (an equal number of images of each class generated).

[1] https://stats.stackexchange.com/questions/318748/deriving-the-kl-divergence-loss-for-vaes

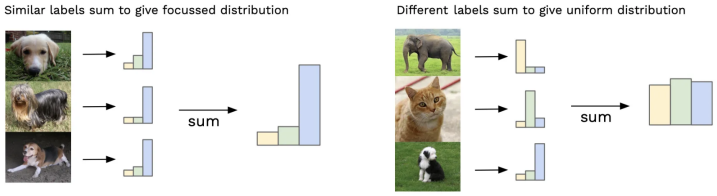


Figure 5: If the produced image is of high diversity, the marginal distribution will appear uniform.

Hence, the ideal label distribution is narrow but the ideal marginal distribution is uniform. Using KL divergence, we can compare the label distribution and the marginal distribution. Better-performing VAEs will have larger KL divergences.

To get the final score, we take the exponential of the result and then take the average for all the images. This will return the Inception score.

In mathematical terms, if the label distribution is p(y—x), where y is the set of labels and x is the image, the marginal distribution is p(y).

2 Implementation

In the implementation section below, I will try different encoding and decoding networks to see which version yields the best result.

2.1 First Attempt

In the first attempt, the encoder contains convolutional layers for feature extractions, followed by leaky ReLU activation for non-linearity.

Encoder

1. First convolution layer takes a single-channel MNIST image (grayscale) and produces 32 feature maps with a kernel size of 3x3 and padding of 1.
2. The Second convolution layer doubles the number of feature maps to 64, and reduces the size of feature maps by roughly half by setting stride to 2.
3. The Third convolution layer keeps the number of feature maps at 64, further reducing the spatial dimension by half.
4. The Fourth convolution layer maintains the number of feature maps at 64. Keeping the earlier feature map size.
5. The output of the convolutional layer is flattened for the fully connected layers.

6. Two fully connected layers produce the mean and log variance of the latent space. The latent dimension is set to 2. After each convolution layer, the leakyReLU activation function is applied.

Decoder
The decoder mirrors the architecture of the encoder but works in the opposite direction.

First, a fully connected layer maps the latent vector to a higher dimensional space, preparing it for reshaping into a convolutional feature map. Transpose of convolutional layers then try to reconstruct the original image. The output image is trimmed and the sigmoid activation ensures the output values are between 0 and 1.

The key statistics of the first attempt are shown below.

- Number of Epochs : 10
- Learning Rate : 0.0005
- Batch Size: 256
- Training Time: 16.99 minutes

The KL divergence error is plotted below.

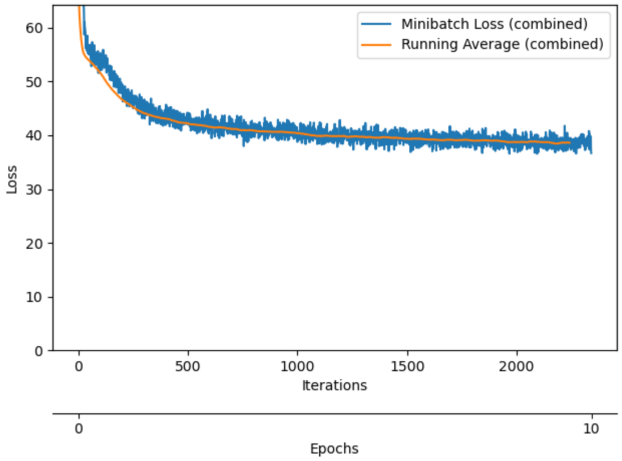


Figure 6: The total loss decreases, as expected.

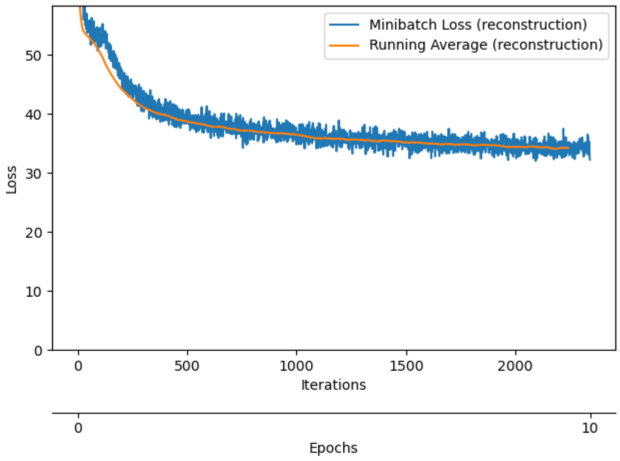


Figure 7: The reconstruction loss decreases, indicating that the reconstructed images are closer to the original.

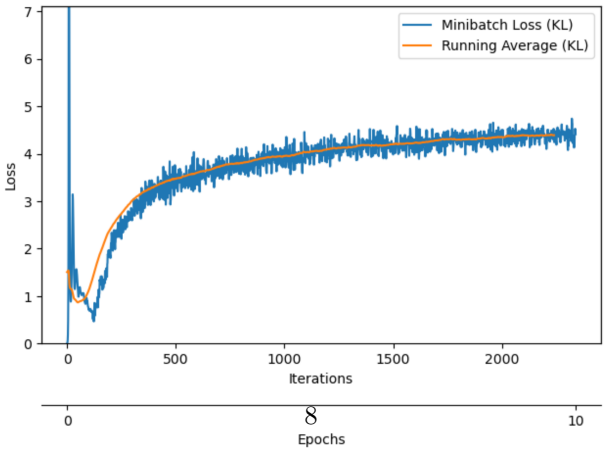


Figure 8: The KL loss increases, showing that the decrease in reconstruction loss comes with the cost of higher deviation from the Gaussian distribution.

Furthermore, we can visualize the latent space.

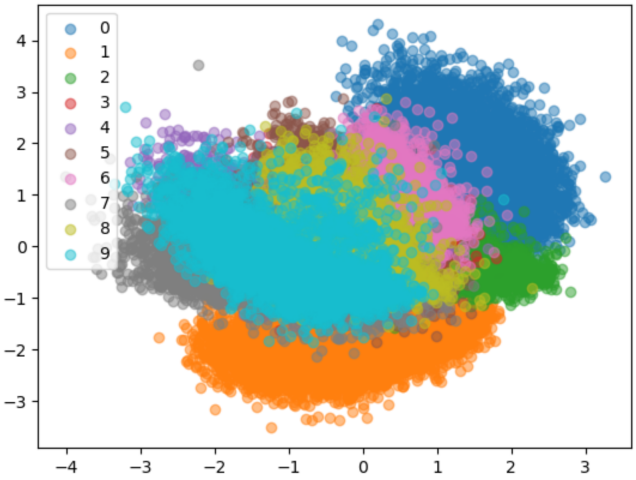


Figure 9: Labeled latent space of the 2D VAE model

In the diagram below, the original MNIST digits and their reconstructed pairs are compared.

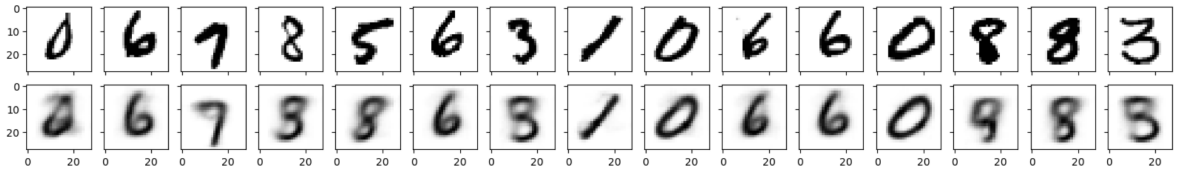


Figure 10: Selected MNIST inputs and its reconstructed pairs

The latent space plot suggests that the latent space is very crowded. There is significant overlap between clusters of different MNIST digits. This leads to ambiguous reconstruction which increases the reconstruction loss. To remedy this problem, we can increase the dimension of the latent space.

2.2 Increasing Latent Space Dimension

To remedy the crowding in the latent space, the second attempt increases the dimension of the latent space to three. The key statistics of the training process are shown below:

- Number of Epochs : 10
- Batch Size: 256

- Learning Rate : 0.0005
- Training Time: 16.31 minutes
- KL Loss (Per Batch): 0.000483
- Reconstruction Loss (Per Batch): 176.2263
- Final Loss (Per Batch): 176.2268

Observe that the reconstruction loss greatly exceeds KL loss. We will discuss this in detail later. The new latent space is visualized in a 3D scatterplot.

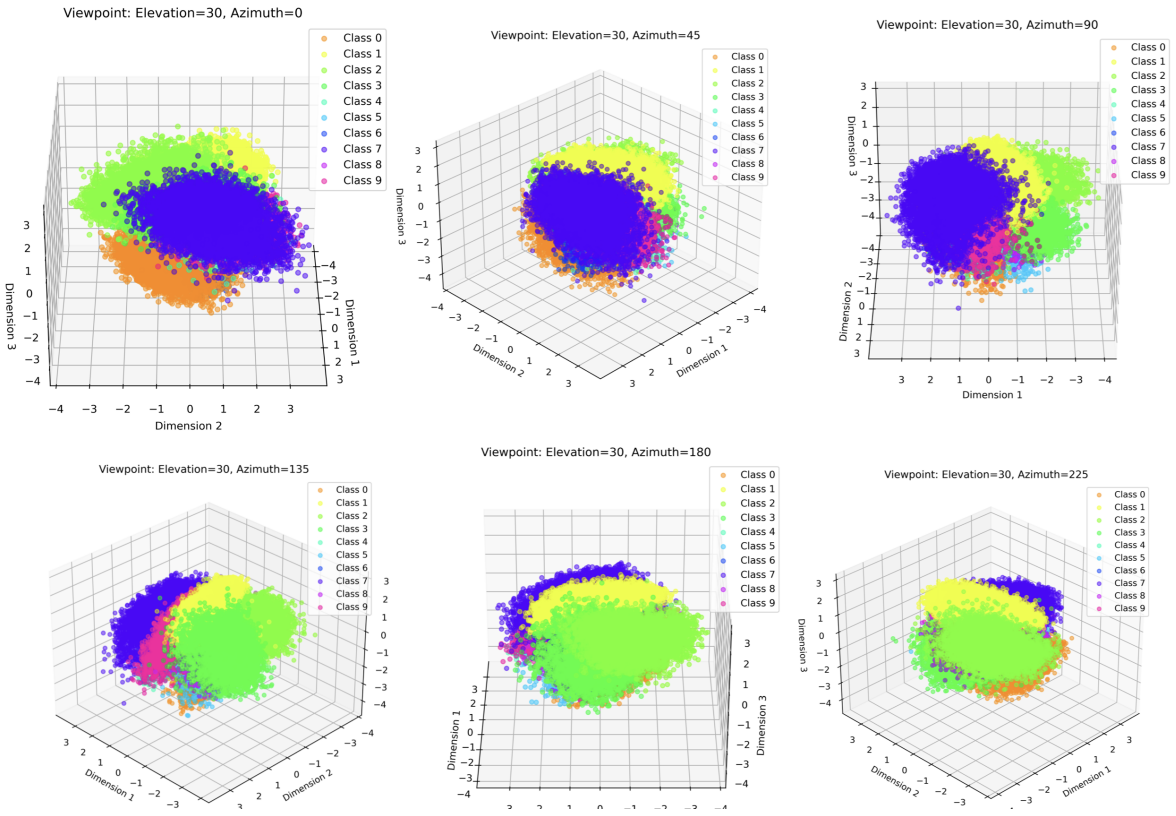


Figure 11: Labeled 3D Latent Space. The overlapping of different MNIST digits have been somewhat mitigated.

Though the overlapping of different MNIST digits in the latent space has been partially mitigated, there is still a significant overlap between digits 7 and 9. This is reflected in the ease of confusion between the two digits in the reconstruction step. The figure below shows some of the original MNIST digits and

their reconstruction counterpart. In this batch, two sevens get confused as nines.

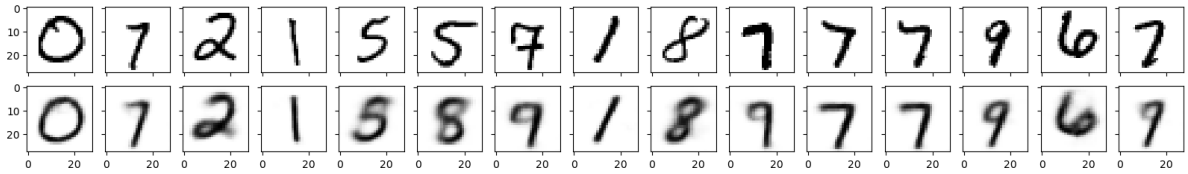


Figure 12: Selected MNIST inputs and its reconstruction pair (Latent space dimension: 3)

The exact degree of overlap can be quantified in further studies. Passing the generated image through a trained classifier, we can also construct a confusion matrix that can help identify shortcomings of the model.

The logical follow-up question is what latent space dimension minimizes the total loss function? With this objective in mind, we can treat the latent space dimension as a hyperparameter that can be tuned.

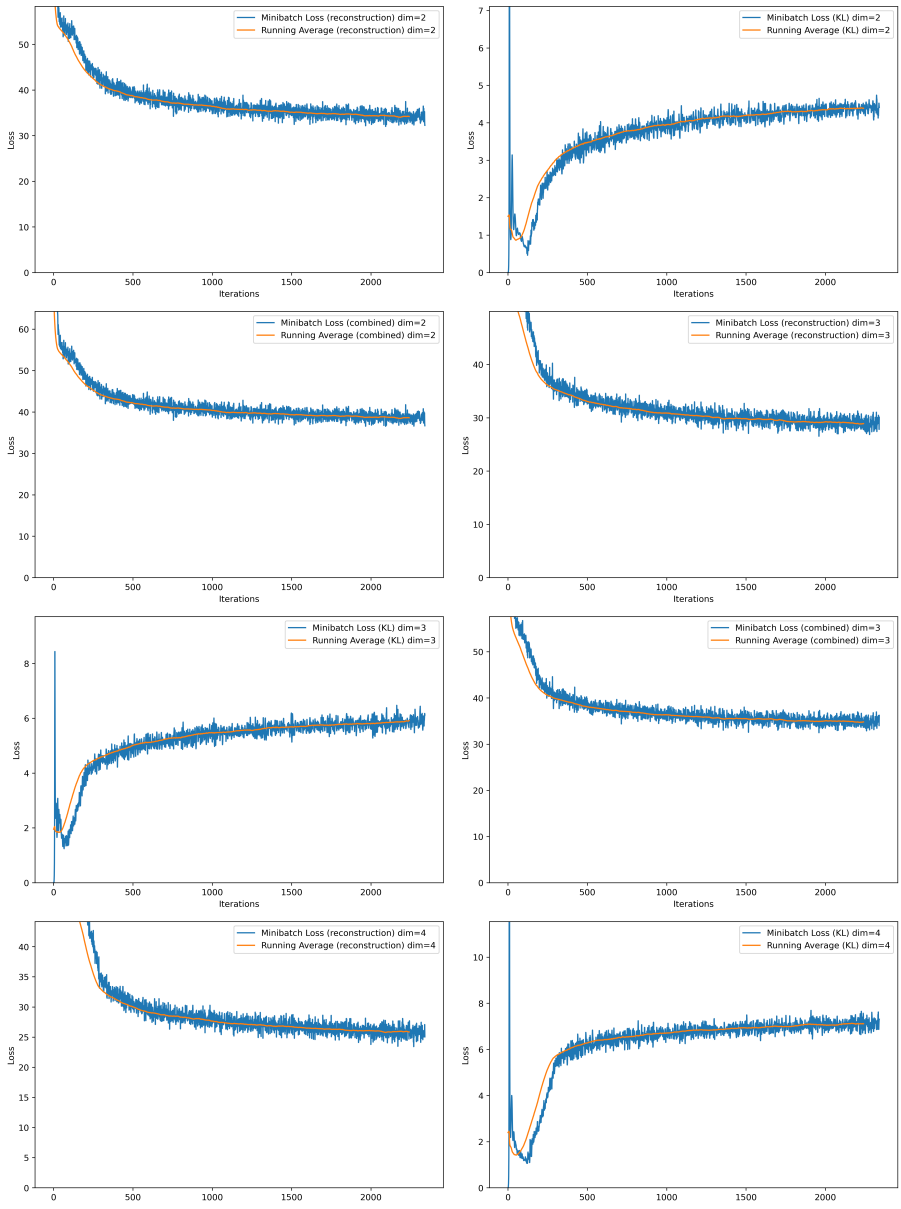
To determine the most optimal dimensionality of the latent space, VAE of latent space dimensions 2,3,4,5,10,30 is trained and the results are evaluated. Higher dimensional latent space cannot be visually inspected, therefore I evaluate the performance by the following criteria.

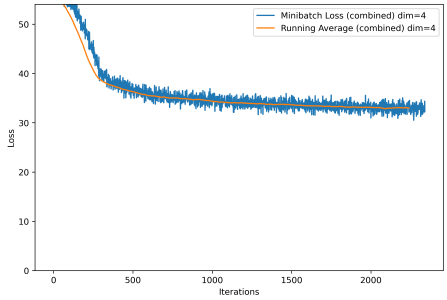
- Reconstruction Loss
- KL Loss
- Total Loss
- Reconstruction visualization
- Interpolation in latent space

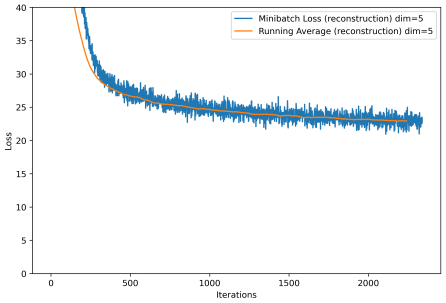
The quantitative results are summarized in the table below.

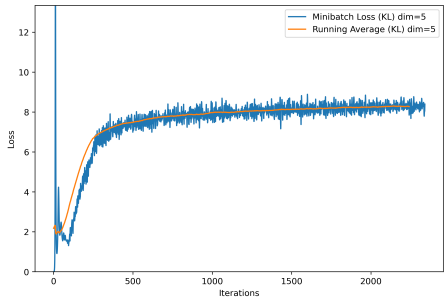
Dimension	KL Loss	Reconstruction Loss	Combined Loss
2	4.4122	32.26999	36.68219
3	6.12124	30.11598	36.23721
4	7.30888	25.05045	32.35933
5	8.39849	22.89452	31.29301
10	11.90844	17.10307	29.01152
30	12.58757	15.39129	27.97886

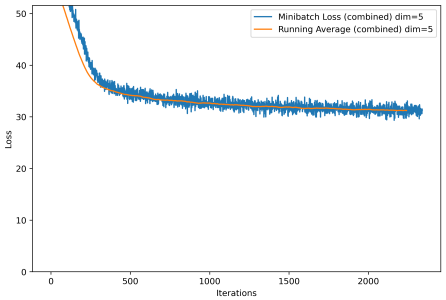
The evolution of the loss function over training is plotted below.

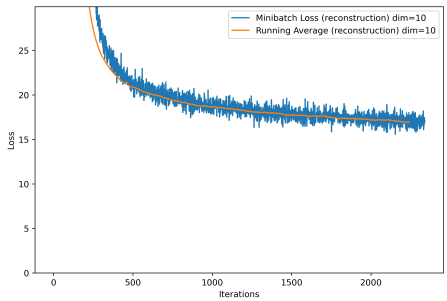

Minibatch Loss (reconstruction) dim=2
Running Average (reconstruction) dim=2
Minibatch Loss (KL) dim=2
Running Average (KL) dim=2
Minibatch Loss (combined) dim=2
Running Average (combined) dim=2
Minibatch Loss (reconstruction) dim=3
Running Average (reconstruction) dim=3
Minibatch Loss (KL) dim=3
Running Average (KL) dim=3
Minibatch Loss (combined) dim=3
Running Average (combined) dim=3
Minibatch Loss (reconstruction) dim=4
Running Average (reconstruction) dim=4
Minibatch Loss (KL) dim=4
Running Average (KL) dim=4
Loss
Iterations

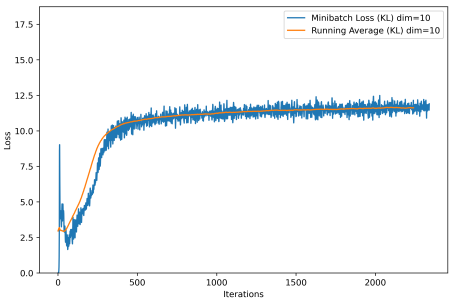

Minibatch Loss (combined) dim=4
Running Average (combined) dim=4
Loss
Iterations

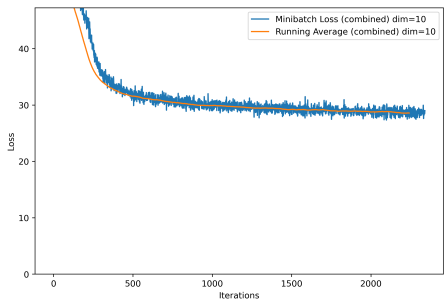

Minibatch Loss (reconstruction) dim=5
Running Average (reconstruction) dim=5
Loss
Iterations

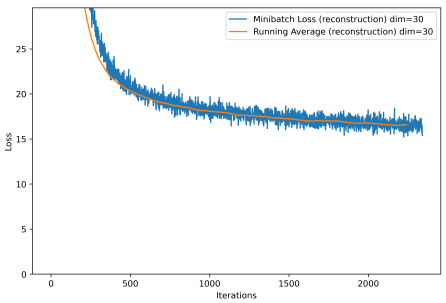

Minibatch Loss (KL) dim=5
Running Average (KL) dim=5
Loss
Iterations

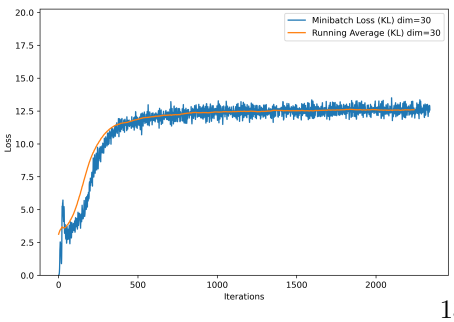

Minibatch Loss (combined) dim=5
Running Average (combined) dim=5
Loss
Iterations

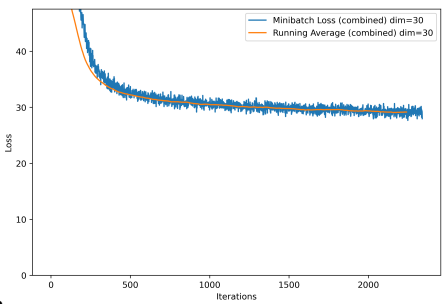

Minibatch Loss (reconstruction) dim=10
Running Average (reconstruction) dim=10
Loss
Iterations


Minibatch Loss (KL) dim=10
Running Average (KL) dim=10
Loss
Iterations


Minibatch Loss (combined) dim=10
Running Average (combined) dim=10
Loss
Iterations


Minibatch Loss (reconstruction) dim=30
Running Average (reconstruction) dim=30
Loss
Iterations


Minibatch Loss (KL) dim=30
Running Average (KL) dim=30
Loss
Iterations


Minibatch Loss (combined) dim=30
Running Average (combined) dim=30
Loss
Iterations

The following trends are observed. During training, the KL divergence term increases to compensate for decreases in the reconstruction error. With lower latent space dimensions, the reconstruction error dominates. In higher latent space dimensions, the KL term dominates. In the 30 dimension latent space, we see a balancing between the KL loss and the reconstruction loss. Due to constraints in computational power, only 10 epochs of training occurred. For all of the models, the total loss have not completely flattened, indicating more training epochs can further improve the performance of the VAE.

2.2.1 Qualitative Assessment of KL Divergence

However, the KL/Reconstruction loss alone is not sufficient to gauge the performance of the model. Below I randomly sample points in the latent space and qualitatively evaluate the output of the VAE. The random sampler samples from the standard multivariate Gaussian distribution.

Figure 13: 2D Latent Space: Using a multivariate standard normal sampler, 64 points in the latent space are sampled, the image is constructed by the decoder.

Figure 14: 3D Latent Space: Using a multivariate standard normal sampler, 64 points in the latent space are sampled, the image is constructed by the decoder.

Figure 15: 4D Latent Space: Using a multivariate standard normal sampler, 64 points in the latent space are sampled, the image is constructed by the decoder.

Figure 16: 5D Latent Space: Using a multivariate standard normal sampler, 64 points in the latent space are sampled, the image is constructed by the decoder.

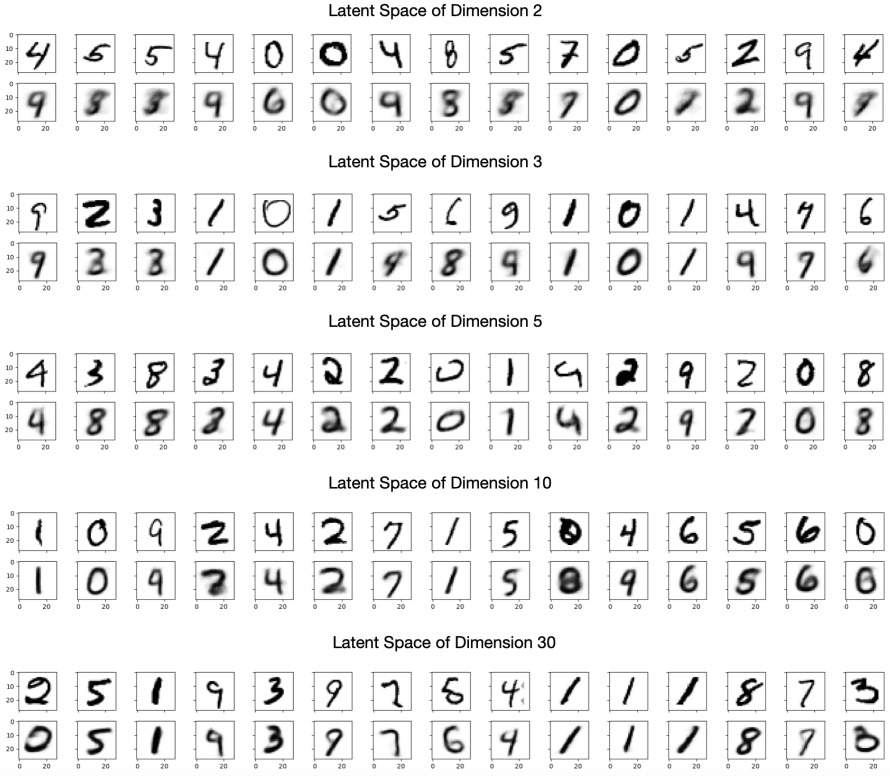
Figure 17: 10D Latent Space: Using a multivariate standard normal sampler, 64 points in the latent space are sampled, the image is constructed by the decoder.

Figure 18: 30D Latent Space: Using a multivariate standard normal sampler, 64 points in the latent space are sampled, and the image is constructed by the decoder.

In higher latent space dimensions, the KL divergence error is high. The latent space is more sparsely populated and gaps exist between different MNIST digit clusters. Random sampling is more likely to result in a point in the gap. When reconstructed, these images do not resemble any of the MNIST digits. This is especially evident in the latent space of dimension 30, where many digits are not recognizable.

2.2.2 Qualitative Assessment of Reconstruction Loss

Random samples of MNIST digits are chosen and compared with their reconstructed pair. As suspected, latent space of higher dimension is more capable of faithfully reconstructing the original image.



Based on visual inspections, the latent space of dimension 5 has the best performance. The encoder/decoder is able to faithfully reconstruct all 15 digits and produce legible images. When random points in the latent space are sampled, 60/64 points produce a digits-like shape that can be interpreted by humans.

Of course, these observations are subjective and can be further refined by a

more quantitative metric (such as inception score). Therefore, the latent space of dimension five is chosen as the model of minimal complexity.

3 Potential for Further Exploration

1. Implementation of inception score for quantitative evaluation of performance.
2. More computation time to further train VAEs.
3. Using confusion matrix to identify overlapping in latent space.
4. Investigate the performance of different neural network architectures (adding more convolution layers etc.)