LAB REPORT 1 [LAB1]

SANTIAGO VILLARREAL VILLARRAGA
8043768
VILLARRS@MYUMANITOBA.CA
01-03-2025

Contents

I, Santiago Villarreal attest that the work I am submitting is my own work and that it has not been copied/plagiarized from online or other sources. Any sourced material used for completing this work has been properly cited. Santiago Villarreal Villarraga

1. Problems

P1-1 What additional steps are required to use the VAE to generate a sample?

We can assume a random normal distribution at the latent space of de decoder so the only need is to generate the values for the Z

```
def generate_single_image(decoder):
    random_latent_vector = np.random.normal(size=(1, latent_dim))
    generated_image = decoder.predict(random_latent_vector)
    plt.imshow(generated_image[0].reshape(28, 28), cmap="gray")
    plt.axis("off")
    plt.show()

generate_single_image(decoder)
```

But for a better implementation it would need to save the last Zlog and Zmean form the encoder, but in this case the random normal distribution works well.

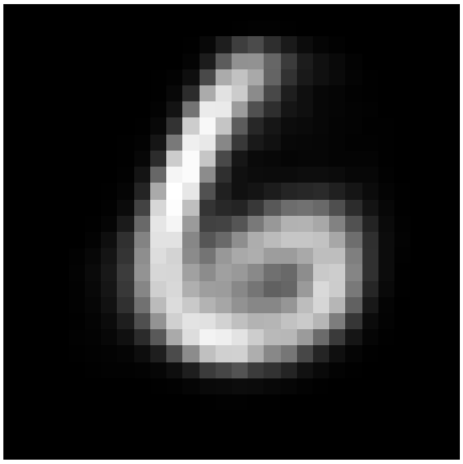

FIGURE 1. Single generated image

P1-2 Augment the code to generate and display 10 random samples from a trained VAE.

```
def generate_images(decoder, n):
    plt.figure(figsize=(20, 4))
    for i in range(n):
        random_latent_vector = np.random.normal(size=(1, latent_dim))
        generated_image = decoder.predict(random_latent_vector)
        ax = plt.subplot(2, n, i + 1)
        plt.imshow(generated_image[0].reshape(28, 28), cmap="gray")
        plt.axis("off")
    plt.show()

generate_images(decoder, 10)
```

FIGURE 2. 10 generated images

P1-3 What conclusions can you draw about the generative quality of the VAE when it is used in a "truly generative" fashion (i.e., sampling the normal distribution output of the model)? Is it common practice to use a VAE in this manner? Or does one usually just use the mean? Support your answer with references.

First of all, the sampling introduces randomness, the generated samples can sometimes show blurriness or less precise details thanks to the effect of binary cross-entropy loss (Foster 2023). VAEs are commonly used in a "truly generative" mode by sampling z - N(0,I) for tasks like synthetic data generation or creative applications (Prince, 2023 Foster, 2023), While the mean of the encoder's posterior distribution q(z x) is typically reserved for deterministic tasks.

P1-4 During training, what is the behaviour of the two loss terms? Does one dominate? Should it? Why or why not?

During early training, the reconstruction loss dominates as the model prioritizes learning to reproduce inputs, establishing latent representations (Foster, 2023), but the idea is that neither dominate, since we want a good generalization with quality, if one dominates during the hole training, for example KL, it can lead to posterior collapse, where latent variables become uninformative, degrading the model's ability to capture data variability (Murphy, 2023; Prince, 2023). but, if reconstruction loss remains dominant, the latent space becomes poorly regularized, leading to overfitting and incoherent samples during generative sampling (Foster, 2023).

P1-5 Increase the latent space from 2 to 3 and re-train the VAE. Visualize the results by showing appropriately selected slices through the latent space. Do you think there is any advantage to a larger latent space for this particular problem? Why or why not?

Yes, the 3d latent space allowed the model to decrease more the KL loss therefore not having a strong dominant in any loss, additional it wasn't harder to train and the images can be sharper than the 2d.

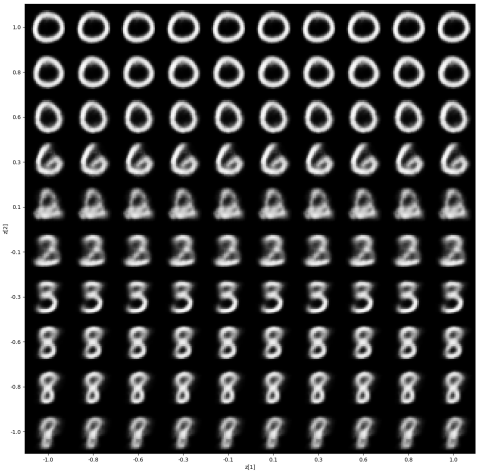

FIGURE 3. 2 vs 1

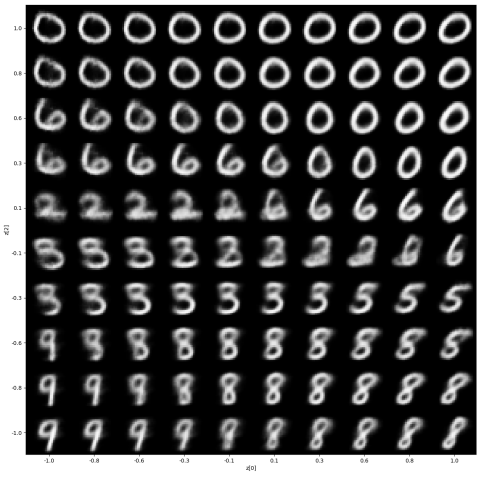

FIGURE 4. 2 vs 0

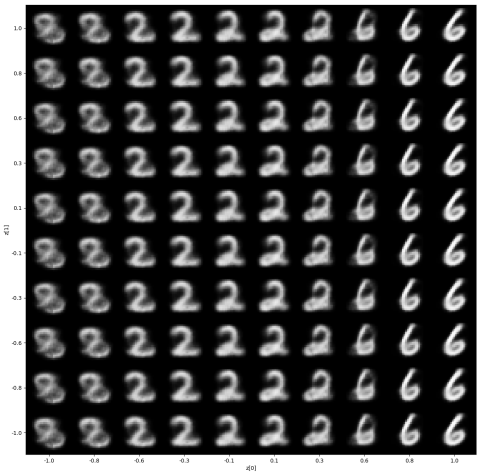

FIGURE 5. 0 vs 1

P2-1 Demonstrate latent space interpolation for a VAE of your choice (the given VAE is sufficient). Pay special attention to how you choose your two latent space points (some approaches are more interesting than others). Provide a summary of your implementation and how it performed.

For this point I had help from Copilot, I asked to code a function that takes 10 values from 2 ranom latent spaces in a VAE, wich hlp me having the following code:

```
def interpolate_points(p1, p2, n_steps=20):
    ratios = np.linspace(0, 1, num=n_steps)
    vectors = []
    for ratio in ratios:
        vector = p1 * (1 - ratio) + p2 * ratio
        vectors.append(vector)
    return np.array(vectors)
```

with this, I used the current VAE, and created the function plotlatentinterpolation, where it takes the vae and the data as input, generates to latent space Z1 and Z2, uses the interpolatepoints to create n points between those latent space. finally it generates n images, and plot it.

FIGURE 6. Interpolation

2. Additional Discussion Questions

- AEs and VAEs both use a latent representation, but VAEs provide a distributional interpretation of the output. Outside of generation, are there tasks that AEs are better at than VAEs? Vice versa? Support your answer with references.

  AEs focus on learning a compact code that accurately reconstructs the input, which makes them especially effective for tasks like denoising or compression where preserving fine details is key, while VAEs introduce a probabilistic twist by modeling uncertainty in the latent space, offering a smooth, continuous representation that is beneficial for tasks such as anomaly detection or interpolation between data points. So if the goal is to capture precise features without concern for variability. (Prince, 2023; Foster, 2023; Murphy, 2022; Murphy, 2023).

References

[1] Prince, S. (2023). Understanding Deep Learning, MIT Press. https://udlbook.github.io/udlbook/
[2] Foster, D. (2023). Generative deep learning, 2nd Edition. O'Reilly Media, Inc. ISBN:9781492041948
[3] Murphy, K. P. (2022). Probabilistic machine learning: An introduction. MIT Press. https://probml.github.io/pml-book/
[4] Murphy, K. P. (2023). Probabilistic machine learning: Advanced topics. MIT Press. https://probml.github.io/pml-book/