Lab 8 Exercise - Exploring Latent Spaces

Amishapriya Singh
Student id: 33333904
as6u21@soton.ac.uk

1. Exploring the latent space of a VAE

1.1. Systematically sample a VAE

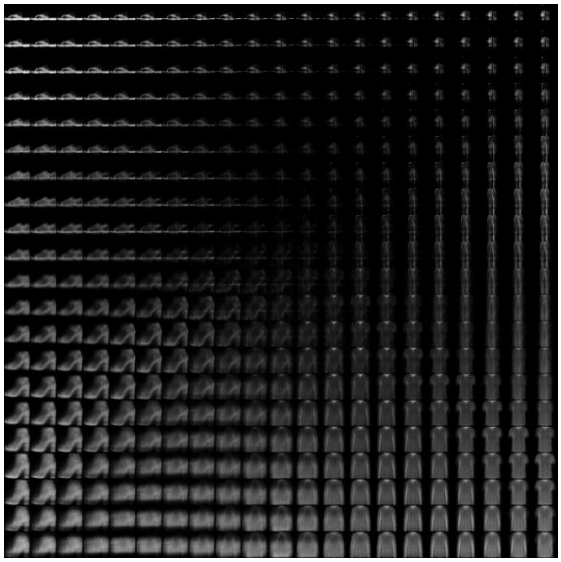

Figure 1. Systematic sampling from VAE latent space

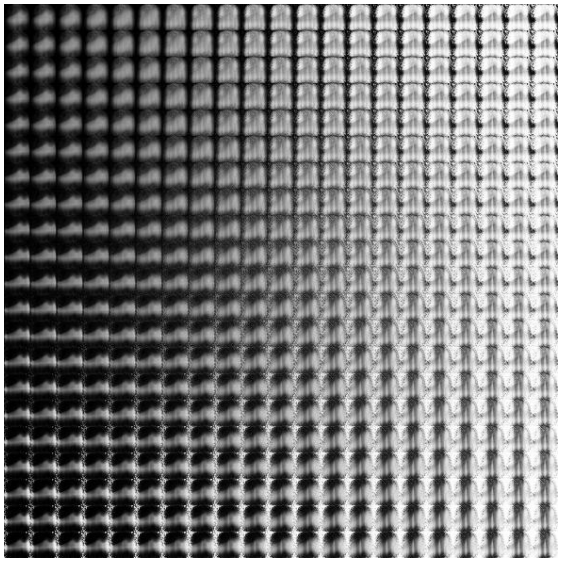

Figure 2. Systematic sampling from autoencoder code space

2. Exploring the code space of a standard auto-encoder

2.1. Systematically sample an Autoencoder

Following is the code for systematically sampling AE code space

```
figure = np.empty((588, 588))
grid_x = np.linspace(-4, 4, 21)
grid_y = np.linspace(-4, 4, 21)[::-1]
for i, yi in enumerate(grid_y):
    for j, xi in enumerate(grid_x):
        z_sample = torch.tensor([[xi, yi]], dtype=torch.float32).view(1, 2)
        output = dec(z_sample)
        img = output.view(28, 28).detach().numpy()
        figure[
            i * 28 : (i + 1) * 28,
            j * 28 : (j + 1) * 28,
        ] = img

plt.imshow(figure, cmap=plt.get_cmap('gray'))
```

2.2. Compare the latent spaces of the VAE and autoencoder

Autoencoders learn descriptive attributes of data in order to describe an observation in a compressed representation while a VAEs represent latent attributes as probability distributions.
The VAE is able to learn latent representation of data that looks simlar to our observed data, such as boots, flip-flops, t-shirts etc. On the contrary, the autoencoders simply learns an encoding which allows us to reproduce the input. We can see from Fig. 2 that the distribution of data within the latent space is uneven as there are areas in latent space that don't represent any of our observed data.