Lab 8 - Exploring Latent Spaces

Abhilash Pulickal Scaria
aps1n21@soton.ac.uk

1 Exploring the latent space of a VAE

1.1 Systematically sample a VAE

Listing 1: Code to generate latent image.

```
gen_img = np.empty((588, 588))

x_points = np.linspace(−4, 4, 21)
y_points = np.linspace(−4, 4, 21)
xx, yy = np.meshgrid(x_points, y_points)
for i in range(len(xx)):
    for j in range(len(yy)):
        z = torch.tensor([xx[j, i], yy[j,
            i]],
            dtype=torch.float32).view(1, 2)
        output = dec(z)
        img = output.view(28,
            28).detach().numpy()
        gen_img[j*28:j*28+28, i*28:i*28+28]
            = img
```

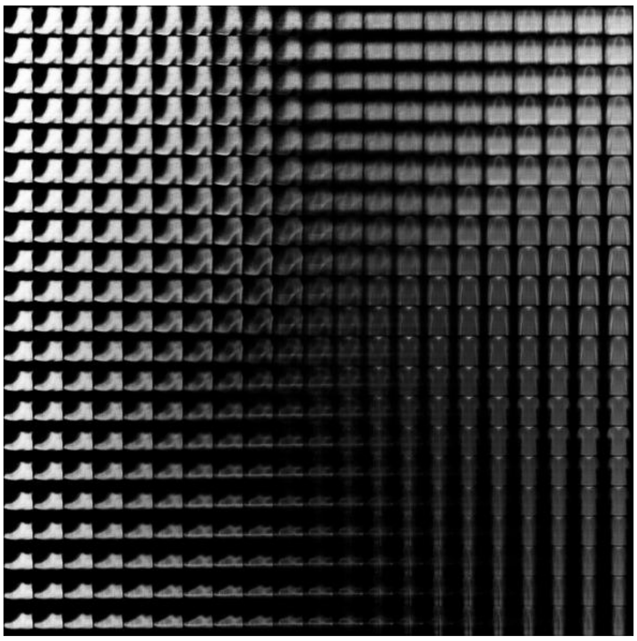
Figure 1: Latent image of VAE.

2 Exploring the code space of a standard auto-encoder

2.1 Systematically sample an Autoencoder

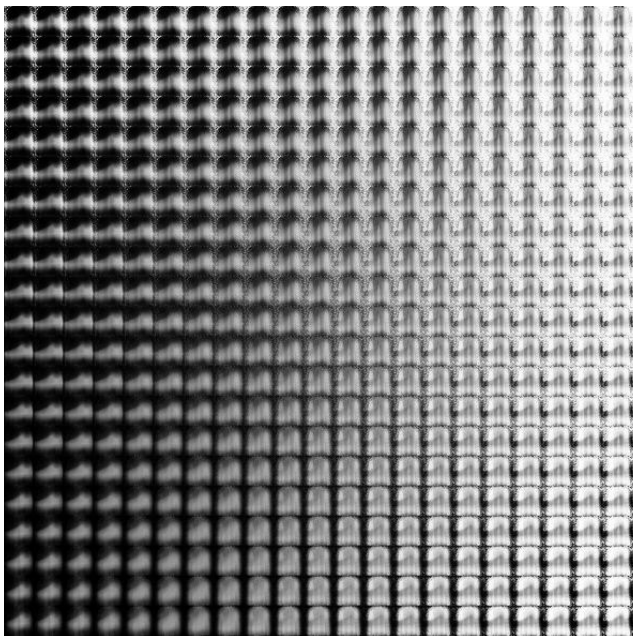
Figure 2: Latent image of autoencoder.

2.2 Compare the latent spaces of the VAE and autoencoder

VAE has a regularized latent space which enables it to generate meaningful decoder output with random values of latent variables. This can be observed from figure 1. The representation of data structures like bag, shoes is visible in the figure.

Autoencoder has non regularized latent space. Sampling a point in this latent space will generate a vaiation of data that the cluster belongs to. AE produces latent space which strongly resembles eigenspace of PCA. From Figure 2 we can see that AE learns most important latent features and compresses the data into smaller subspaces.